“It’s Freedom to Put Things Where My Mind Wants”: Understanding and Improving the User Experience of Structuring Data in Spreadsheets

George Chalhoub
Microsoft Research
& University of Oxford
United Kingdom
george.chalhoub@cs.ox.ac.uk

Advait Sarkar
Microsoft Research
& University of Cambridge
United Kingdom
advait@microsoft.com

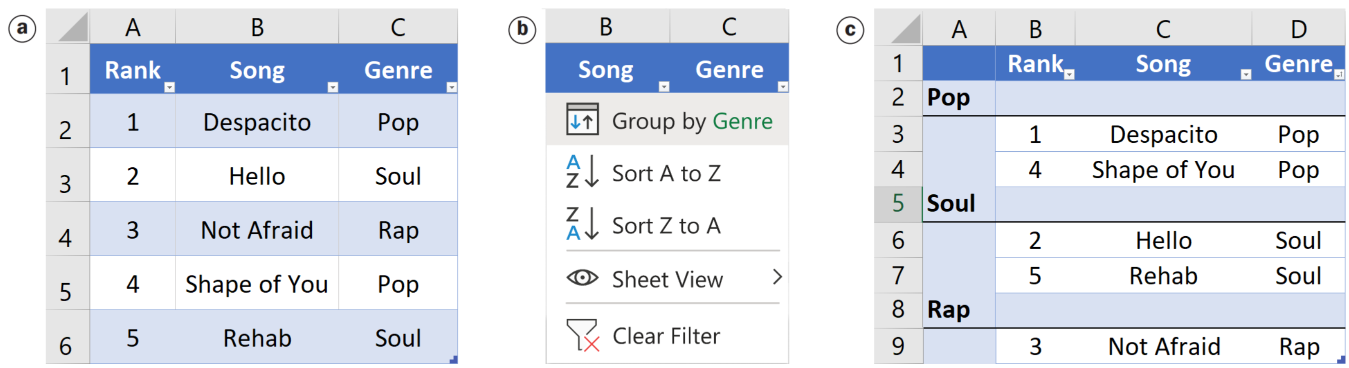

Figure 1: A proposed spreadsheet feature *Group by Column Value* allows users to visually group rows which share the same values in a particular column. (a) Structured table containing song data (e.g., name and genre). (b) User selects the *Group by* option from the *Genre* column drop-down menu. (c) Table layout is re-organized by *Genre*.

ABSTRACT

Despite efforts to augment or replace the 2-dimensional spreadsheet grid with formal data structures such as arrays and tables to ease formula authoring and reduce errors, the flexible grid remains overwhelmingly successful. Why? We interviewed a diverse sample of 21 spreadsheet users about their use of structure in spreadsheets. It emerges that data structuring is subject to a complex network of incentives and constraints, including factors extrinsic to spreadsheets such as the user’s expertise, auxiliary tools, and collaborator needs. Moreover, we find that table columns are an important abstraction, and that operations such as conditional formatting, data validation, and formula authoring can be implemented on table columns, rather than cell ranges. To probe this, we designed 4 click-through prototypes for a follow-up study with 20 participants. We found that although column operations improved the value proposition of structured tables, they are unlikely to supplant the advantages of the flexible grid.

CHI ’22, April 29-May 5, 2022, New Orleans, LA, USA

ACM ISBN 978-1-4503-9157-3/22/04…$15.00
https://doi.org/10.1145/3491102.3501833

CCS CONCEPTS

• **Human-centered computing** → **Empirical studies in HCI**; Interface design prototyping; • **Applied computing** → *Spreadsheets.*

KEYWORDS

spreadsheets, user experience, structure, data

ACM Reference Format:
George Chalhoub and Advait Sarkar. 2022. “It’s Freedom to Put Things Where My Mind Wants”: Understanding and Improving the User Experience of Structuring Data in Spreadsheets. In *CHI Conference on Human Factors in Computing Systems (CHI ’22), April 29-May 5, 2022, New Orleans, LA, USA.* ACM, New York, NY, USA, 24 pages. https://doi.org/10.1145/3491102.3501833

1 INTRODUCTION

The spreadsheet is among the most widely-used tools for data storage, manipulation, and analysis [94]. A key reason for its success as an end-user programming platform is its unrestricted 2-dimensional grid. The grid allows direct inspection and manipulation [98] of data values, effectively starting its *abstraction gradient* [42, 90] at zero. It facilitates exploratory programming [63], since it requires very little *premature commitment* [42] in comparison to traditional textual programming languages, which typically require upfront commitment to a data structure e.g., an array, dictionary, variable, data frame, etc.) and a type, in the case of statically typed languages, before data can be stored and manipulated. In a spreadsheet, to instantiate a data set requires the user only to choose a suitable location and layout for inputting the data.

On the other hand, it is well-documented that the flexibility of spreadsheets also makes them error prone [79–81, 83]. Without formal types or data structures, spreadsheets suffer from classes of error that in traditional programming languages are easily detected, or completely prevented. Perhaps the most famous and disastrous spreadsheet error of all, the Reinhart-Rogoff error [52], could in part be traced back to an instance of such an error class: the omission of certain cells, notionally part of a set, in a `SUM` function. If implemented as a sum over an array in a traditional programming language, the error would have been impossible. Of course, traditional programming languages are susceptible to their own kinds of errors to which spreadsheets are immune, such as the common novice error of forgetting to return a value from a non-void function [16]. Nonetheless, the point here is not that spreadsheets are more error-prone than traditional programming languages in any absolute sense, but rather that the *particular* kind of error in Reinhart-Rogoff is prevented by the design of primitive data structures in many traditional languages.

Spreadsheet users often operate on higher-level abstractions (such as arrays and matrices) despite having neither a formal understanding of such structures, nor first-class support for such structures in their spreadsheet application. This mismatch between the conceptual level at which the user is operating on the data, and the way in which the spreadsheet tool represents the data, is a source of errors. A version of this problem, literally called the 'match-mismatch conjecture', was observed by visual languages researchers in the early 1990s [41].

There have since been several commercial and research attempts to address this mismatch (described in detail in Section 2). The approach is to introduce data structures that aim to strike a balance between the freedom and flexibility of the traditional grid, and the safety and power of formal types and structures. Each of these solutions imposes some restrictions on the freedom of the grid, as well as increases the complexity of the formula language to introduce operations that apply at the level of the structure, a necessary cost of higher abstraction. None of these proposals has seen commercial success in any way comparable to the success of the free grid. Even modest attempts to introduce structure into the free grid in a limited and non-intrusive manner, such as formal tables and arrays, have very limited uptake (we uncover some reasons for this in Section 5).

The failure of widespread adoption of more structured spreadsheets is a true enigma, given that evidence of the mismatch between spreadsheets' low-level data representation and the high-level user representations goes back decades [48]. What could be the reason for this? Maybe structuring data requires too much expertise, or too much effort. Maybe the structures proposed thus far come from a programmer-centric view of spreadsheets, and it is incorrect to assume that the same structures that are so useful to professional programmers (arrays, dictionaries, objects, and the like) are also useful for end-user programmers. Depending on the reason, the match-mismatch problem of spreadsheet data structuring may require very different solutions. Answering this question is therefore an extremely valuable prize. We cannot claim a definitive answer in this paper, but we have nonetheless made important strides that throw fresh light on the issue. This paper makes the following contributions:

- We conducted contextual interviews of 21 spreadsheet users from a variety of domains and levels of expertise, in which we asked about the structures of data in their spreadsheets (Section 4). Our results map out a network of constraints and influences on the layout and structure of user data, which is significantly more complex than previously thought, and leads to the important conclusion that the match-mismatch problem involves several ecosystem concerns that cannot be controlled through innovative spreadsheet design (Section 5).
- We present designs for *table column operations*, a class of potentially valuable augmentations to formal spreadsheet tables grounded in our empirical evidence for users' needs in structuring their data (Section 6). We conducted scenario-based interviews with 20 participants from our initial study to probe whether they could significantly improve the value proposition of formal tables. We found that while column operations were seen as valuable, they are unlikely to supplant the need for the flexible grid (Section 7).
- We discuss how our data extends phenomena observed in previous work by noting that spreadsheet users experience structure as being on a *continuum*, with respect to certain *operations*, and with a sense of loss of *agency*. We confirm speculations in previous work that spreadsheet authors experience tensions in data layout for comprehension. Our conclusion that the flexible grid is useful and necessary despite its drawbacks reveals opportunities in other parts of the spreadsheet ecosystem, such as better integration with auxiliary tools and collaborator training (Section 8).

2 BACKGROUND: THE PROBLEM OF STRUCTURE IN SPREADSHEETS

2.1 Studies of spreadsheet use

Early interviews found that nearly all spreadsheets in work environments were built collaboratively by co-workers of different levels of expertise [77]. The flexibility of spreadsheets led to concerns of spreadsheet risks and errors [32, 47, 48, 80, 95]. More recent ethnographic studies show that the grid structure supports particular patterns of interaction [34, 69, 69, 106]. Several studies have investigated how data workers capture, analyze, process and maintain data in work environments [4, 60, 61, 66, 67, 76, 87, 100, 113]. Spreadsheets are widely used for data entry, storage, structuring, annotation, analysis, and reporting [15, 20, 29, 33, 37, 45, 49, 55, 61, 85].

Others have investigated facets of data work such as electronic data capturing, data discovery, extraction, classification, munging, warehousing, mining, modeling and reporting [4, 60, 76, 113]. Fewer studies extend their consideration to non-expert spreadsheet users, who work with data more informally [7, 13, 24, 82]. Bigelow et al. [7] reported that users who frequently use data alternate between tools and didn't follow a linear sequence of tasks. Convertino et al. [24] found that spreadsheet users did not take full advantage of the data available due to a gap in their quantitative analytical skills. Others have explored the strategies users apply to cope with uncertainty in their data [11–13].

2.2 Spreadsheet data arrangement practices

Spreadsheet data arrangement practices are largely ad hoc [101], and suffer from inadequate tool support, cultural resistance to tool uptake, and limited user expertise [53]. Users commit 'considerable management effort' to locating different sources of data, reconciling differences, and arranging data for sharing.

'Best practices' for organizing and formatting data are often proposed [54, 84]. Professional and standards bodies codify elaborate guidelines for designing comprehensible spreadsheets, e.g., the FAST standard [103], Microsoft Office guidelines [73]. However, a survey of spreadsheet usage found that 90% of users don't follow any written guidelines [104]. Written guidelines are often under-specified with respect to common user concerns, e.g., data validation, conditional formatting rules, charts [101].

Research proposing new spreadsheet data models frequently begin with tabular structures [2, 3, 18, 21, 27, 36, 38, 72]. Others acknowledge common spreadsheet data patterns [50, 99] for use in pattern-matching algorithms. Teixeira and Amaral [102] present a brief catalog of spreadsheet data arrangement patterns derived from an analysis of the Euses and Enron spreadsheet corpora.

Most previous proposals do not begin with an empirical understanding of *why* users might structure their data in a certain way, and how it might differ in different domains [102]. A recent investigation by Bartram et al. [5] begins to address this gap. They conducted a qualitative study with 12 data workers exploring their data layouts. They found that spreadsheet data layouts are meaningful and have structural affordances, e.g., spatial organization, annotations, metadata, that are lost when exported into other analytical or visualization tools (e.g., Tableau). They note that spreadsheet research has mostly focused on 'domain experts' and 'analysts'. To address this, we interviewed spreadsheet users at varying levels of expertise, and from diverse domains (Section 3.1). We expand on their findings by explicitly exploring how the arrangement of spreadsheet data supports or hinders various workflows.

2.3 Addressing errors through grid structure

Spreadsheet errors can be mitigated through auditing and testing tools [2, 23, 40, 51, 111, 114], enhancements to the formula language [57, 58, 74, 78, 93, 97, 107], and grid structure alterations [17, 31, 36, 44, 74, 75]. Discussion of the auditing, testing and enhanced formula approaches to spreadsheet error reduction is beyond the scope of this paper — they are mentioned here for completeness. The primary concern of this paper is how the grid structure itself has been previously conceived of as a source of error and a potential target for design.

Grid structure alterations suggest alternatives to the 2-dimensional spreadsheet grid to deal with errors. For example, gradual structuring [75], the 'lish' data model [44], Forms/3 [17], and object spreadsheets [70] all modify the spreadsheet so that instead of an unbounded and unstructured 2-dimensional grid, the spreadsheet becomes a repository for typed structures such as arrays and objects, which are in turn visually rendered as smaller, bounded grids.

To maintain compatibility with commercially dominant spreadsheet applications, one influential line of research has explored 'model-driven spreadsheet development' [36, 38, 72]. In this approach, an abstract specification language is used to generate a model to which a spreadsheet must conform. A spreadsheet application can then validate the user's spreadsheet against the model periodically as the user edits.

While superficially similar to spreadsheet grids, altered grids have strict limitations derived from their underlying models and data types. For example, the template rule of the lish model enforces that the first element of a list forms a template for subsequent elements, which must follow the same pattern. Thus, the user may find, in an interface that otherwise resembles a spreadsheet, that certain cells cannot be edited, or cannot be empty, or must contain values of a certain type, in order to match the template. Cells in altered grids are subject to constraints extraneous to the cell itself, in violation of Kay's 'value rule' assumption, which impairs spreadsheet comprehensibility [62]. Proponents of altered grid structures argue that this is the system working as intended, and confers error prevention benefits, while also making many abstract operations less viscous [42], i.e., requiring less manual effort. While true, these tools require substantially greater expertise to use than spreadsheets based on unstructured grids, and put greater restrictions on end users. Their use requires planning and premature commitment [42] to a certain set of structures or a certain regime of editing operations, which is simply incompatible with the exploratory, variable, and contingent nature of daily spreadsheet use.

To avoid grid structure alteration but retain the benefits of typed data structures, three alternative approaches have been taken. The first is to embed the typed structure into individual cells (such as Blackwell et al.'s matrices in cells [10]). These allow types with abstract operations to co-exist with unstructured data on the grid. The second is to dynamically allocate regions of the grid as being renderers for higher-level types. Both Microsoft Excel's 'dynamic arrays' [109] and tables[1] follow this paradigm. In this approach portions of the grid behave differently, with lesser or greater flexibility, depending on whether they are part of an array/table or not. The third approach is 'multiple representations', as adopted by Calculation View [93], where regions of the grid which conceptually correspond to a higher-level structure can be viewed and manipulated as such in a separate, non-grid view.

While these approaches show the potential value of integrating or layering typed data structures with flexible 2-dimensional grids, we lack empirical data that might allow us to reason through in detail how their limitations might impact the daily work of non-expert end users. We attempt to address this shortcoming by explicitly exploring the constraints and pressures that spreadsheet users experience when structuring their data.

In summary, the flexibility and ease-of-use of spreadsheets have made them a global commercial success, but also extremely error-prone. Approaches that introduce formal structures into spreadsheets reduce the flexibility of the grid, but the reasons for their limited uptake have not been well-studied. Previous studies have made relevant observations, but have not yet mapped out the wide set of constraints and influences on spreadsheet data structures that are encountered by users of different levels of expertise and different domains. To address this, we study the data structuring experiences of a diverse sample of 21 spreadsheet users.

[1]https://support.microsoft.com/en-us/office/overview-of-excel-tables-7ab0bb7d-3a9e-4b56-a3c9-6c94334e492c

3 METHODS

Our research questions are:

(1) What factors influence users' choices for structuring data in spreadsheets?
(2) Can the design of spreadsheet structures reduce or eliminate the need for the flexible, 2-dimensional grid?

We conducted two studies. First, a semi-structured contextual interview (n=21) exploring how users experience structuring data in spreadsheets; their needs, constraints and pain points (Sections 4 and 5). Second, a scenario-based interview, with participants from study 1 (n=20), using click-through prototypes of spreadsheet features aimed to address needs identified in study 1. The objective of the second study was not to evaluate the prototypes, but rather to determine whether improving the value proposition of formally structured data can reduce or eliminate the need for a flexible, 2-dimensional grid (Sections 6 and 7).

3.1 Recruitment

We sent targeted emails and advertisements to institutional mailing lists and social media. Interested participants were directed to a screening questionnaire containing questions about gender, age, and their use of spreadsheets. We asked participants to specify which spreadsheet software they use (e.g., Microsoft Excel, Google Sheets), which spreadsheet features they use (e.g., sorting/filtering, data validation), how often they use spreadsheet tools, the approximate amount of time spent using spreadsheets at work (e.g., 0-25%, 51-75%), a self-assessed skill level (similar to previous studies [89, 91]), and whether they were willing to share and walk us through at least one recent spreadsheet they have created, explaining decisions, practices, and challenges.

We received 91 responses to the screening form. Of these, we selected a sample of 21 users stratified across domains and levels of expertise, who were willing to walk us through their spreadsheets. We arranged interviews with them over email. Prior to each interview, we asked applicants to email anonymous versions of their spreadsheets with any confidential or personally identifiable data removed.

Table 1 summarizes the demographics of our sample. We interviewed 15 men, 6 women, 0 non-binary/other. 18 participants were between 18 and 34 years old, 2 between 35 and 44, and 1 between 45 and 54. 13 participants were employed, 6 were university students, and 2 were self-employed. We used participants' responses to the questions around the frequency and types of spreadsheet use, as well as their self-assessed skill, to classify their spreadsheet expertise on the Dreyfus skill acquisition scale (novice, competent, proficient, expert, and master) [35]. Participants were thus classified as proficient (n=8), expert (n=5), novice (n=4), master (n=3) and competent (n=1). Microsoft Excel was the most used spreadsheet (n=18), followed by Google Sheets (n=10) and Apple Numbers (n=2). 10 participants used multiple spreadsheet software, while other participants exclusively used Microsoft Excel (n=8), Google Sheets (n=2) and Apple Numbers (n=1).

Our organization's ethics committee approved this project. Prior to each interview, participants were briefed and signed an informed consent form explaining our study and data confidentiality practices. Participants were compensated USD $50 in electronic store vouchers for participation in both studies, and $25 for participation in only the first. Participants could withdraw themselves and their data at any point, without loss of compensation, and without providing a reason. No participant withdrew. Our study materials including our recruitment questionnaire and interview protocol are available here: https://osf.io/enwxj/.

4 INTERVIEW 1: WHAT FACTORS INFLUENCE USERS' CHOICES FOR STRUCTURING DATA IN SPREADSHEETS?

4.1 Interview Procedure

We conducted semi-structured interviews as follows. First, we briefed our participants, asked them to introduce themselves and how they used spreadsheets. Using screen-sharing, we asked them to walk us through their own spreadsheet(s), their goals and structuring experiences. Interviews covered the following themes:

How users structure data: We asked participants about data structuring purposes (e.g., testing data, arithmetic/logic operations), moving to questions about the shape (e.g., tabular, non-tabular), the structure (e.g., unstructured, semi-structured), and the arrangement (e.g., multiple, single sheet). We asked how they imported data (e.g., data entry and input), manipulated it (e.g., data cleaning, formatting and processing), and exported it (e.g., auxiliary tools).

Factors affecting structure: We asked participants why they chose a particular data structure (e.g., improving comprehension, doing computations), and what influenced their decisions (e.g., audience, collaborators). We asked participants about external pressures (e.g., requirements from senior management). We asked if the complexity and length of the data or purpose of a spreadsheet affected the structure.

Pain points: We asked participants to describe frustrations (e.g., performance), challenges (e.g., inability to use functions), constraints (e.g., time and knowledge limitations), restrictions (e.g., inability to export data) or pain points experienced when structuring data. We asked how they discover and resolve spreadsheet errors, seek help when facing issues, and deal with repetitive or laborious work.

Use of formal structures: We asked participants whether and how they used any formal or structured tables (e.g., Excel tables) and arrays (e.g., dynamic arrays). We asked about their perceptions or experiences of these structures, probing about their relative advantages (e.g., more automation, less errors) and disadvantages (e.g., difficulty of use, less flexibility) , probing users how they could be improved to suit their workflows.

Participants were encouraged to elaborate and ask for clarification at will. As is common practice in semi-structured interviews, the interviewer followed a predetermined list of questions, but with the discretion to ask follow-up probing questions or skip questions that had already been covered. A single researcher conducted all interviews between June and August 2021 in the United Kingdom in English. The interviews were conducted remotely via Microsoft Teams and Zoom. We recorded audio, video, and notes. Our interview protocol can be found at https://osf.io/uyr3t/.

Table 1: Participant demographics

P# (Expertise)	Age (Gender)	Job Title (Field)	Software	Spreadsheet Use	Collaboration	Input Source	Output Source
P1 (Proficient)	25 - 34 (M)	Postdoctoral Researcher (Linguistics & Languages)	Microsoft Excel	Creating, manipulating and exporting dictionaries.	No Collaboration	.TSV from Python	.TSV for Github repository
P2 (Expert)	25-34 (M)	Mid- Consultant (Monitoring & Evaluation)	Microsoft Excel	Analyzing the performance of NGO programs.	Work Colleagues	.CSV from IBM SPSS Statistics	.CSV for IBM SPSS Statistics
P3 (Expert)	25-34 (M)	Research Associate (Computer Science)	Microsoft Excel Google Sheets	Tracking survey data and participant demographics.	Research Team	.CSV from Prolific Academic	.CSV for Python & R
P4 (Expert)	25-34 (M)	Risk Analyst (Commercial Insurance)	Microsoft Excel	Analyzing the risk level of overseas investments.	Senior Management	.CSV from Salesforce's CRM	Graphs and Tables for Microsoft Word
P5 (Proficient)	25-34 (M)	Self-employed Developer (Full Stack Development)	Microsoft Excel Google Sheets	Tracking hours worked and client payments.	Freelance Clients	Manually Added Data	None
P6 (Proficient)	25-34 (M)	Biomedical Scientist (Immunology)	Microsoft Excel Google Sheets	Analyzing and storing vaccine immunology data.	Work Colleagues	.CSV from GraphPad Prism	.CSV for R
P7 (Proficient)	25-34 (M)	Logistics Officer (Military)	Microsoft Excel	Tracking the maintenance for military vehicles.	Work Colleagues	Manual Added Data	Tables for Microsoft PowerPoint
P8 (Proficient)	25-34 (F)	Compliance Manager (Compliance Industry)	Microsoft Excel	Tracking risks and contracts with third parties.	Work Colleagues	Manual Added Data	None
P9 (Master)	35-44 (F)	Program Manager (Clinical Informatics)	Microsoft Excel	Warehousing, cataloging and analyzing hospital data.	Work Colleagues	.CSVs from work colleagues	Pivot Tables for Microsoft Word
P10 (Expert)	25 - 34 (M)	Doctoral Researcher (Cyber Security)	Numbers (Apple)	Analyzing cyber security lab experimental data.	Research Team	.CSV from Python	.CSV for Python & R
P11 (Expert)	25-34 (F)	Clinical Doctor (Human Aid & Relief)	Microsoft Excel	Tracking daily patient health and symptoms.	No Collaboration	Manually Added Data	Graphs and Tables for Microsoft Word
P12 (Proficient)	25 - 34 (M)	Education Administrator (Education Leadership)	Google Sheets Microsoft Excel	Analyzing student grades and teacher evaluation.	No Collaboration	.CSV from AP Central	Graphs and Tables for Microsoft Word
P13 (Novice)	25-34 (F)	Response Coordinator (Youth Services)	Microsoft Excel	Tracking emergency response activities.	Work Colleagues	Manually Added Data	None
P14 (Novice)	18 - 24 (M)	PhD Student (History)	Google Sheets	Tracking the books read for comprehensive exams.	Academic Supervisor	Manually Added Data	None
P15 (Master)	25-34 (M)	Data Scientist (Investment Management)	Microsoft Excel Google Sheets	Analyzing the performance of investment portfolios.	Work Colleagues	.CSV and .XLSX from co-workers	.CSV for Python & R
P16 (Novice)	25-34 (M)	Doctoral Researcher (Theology & Religion)	Google Sheets	Tracking and analyzing library-based research data.	Research Team	Manually Added Data	None
P17 (Master)	45-54 (M)	CISO (Professional Services)	Microsoft Excel Numbers (Apple)	Analyzing security risks, threats and vulnerabilities.	Senior Management	.CSV from Microsoft Access	Graphs and Tables for Google Docs
P18 (Novice)	25-34 (F)	M.D. Candidate (Medicine)	Microsoft Excel Google Sheets	Analyzing data from orthopedic conferences.	Academic Advisor	Manually Added Data	None
P19 (Proficient)	25-34 (F)	Research Assistant (Political Science)	Microsoft Excel Google Sheets	Analyzing US state policies on dual enrollment.	Principal Investigator	.CSV from US Dept. of Education	.CSV file for Stata
P20 (Competent)	25-34 (M)	Masters Student (Computer Science)	Microsoft Excel Google Sheets	Tracking purchased for video game item purchases.	No Collaboration	Manually Added Data	None
P21 (Proficient)	35-44 (M)	Principal Researcher (Computer Science)	Microsoft Excel Google Sheets	Analyzing and cleaning data from a large diary study.	Research Team	.XLSX from work colleagues	.CSV for ATLAS.ti

4.2 Data Analysis

We transcribed and analyzed all 21 interviews using iterative open coding [105] in accordance with Braun and Clarke's thematic analysis [14]. We observed data saturation [25, 43, 96] (i.e., no new codes emerged) between the 19^{th} and the 21^{st} interview. The analyzed material of the first study consisted of 21 hours and 46 minutes of recorded interviews (~151,055 words), each on average 62 minutes long (~7,193 words).

The researcher who conducted the interviews independently completed an initial coding of all transcripts, identifying 608 relevant participant utterances and assigning them to 272 codes. A second researcher then cross-checked the codes against the interview transcripts, asking for clarifications and additional context from the first researcher, who annotated the study data to note ambiguities and disagreements. The initial agreement was 88%. The two researchers negotiated each disagreement, resulting in the re-coding of 73 participant utterances (12% of the total set of utterances), the addition of 30 codes, the deletion of 45 codes, and the merging of 19 codes, resulting in a final set of 236 unique codes.

Through discussion and a prioritization process based on the strength of evidence for each code (i.e., in how many participants the code occurred and how important it was to their experience of data structuring), we organized these codes into themes which are reported in detail in Section 5. The full list of codes is available in Appendix A.

4.3 Limitations

Our study has limitations common to much qualitative research. First, research quality depends on the interviewer's skill [59] and the quality of the questions asked [8]. Inexperienced interviewers may not be able to formulate good prompt or probe questions, thus missing relevant data [64], or introducing their own personal biases. To address this, one researcher, who was trained to conduct interviews consistently and neutrally, conducted all 21 interviews.

Second, self-reporting biases are common in interviews [1]. Participants might not respond accurately because they do not remember specific details. Others could be concerned about the interviewer's perception of them and therefore answer according to how they wish to be perceived. Factors such as ethnicity influence the answers that different social groups are willing to give [19]. To minimize these biases, we avoided leading questions and relied on open-ended questions, inviting participants to answer in their own words. When participant answers were less detailed, we prompted participants to give concrete examples.

Third, spreadsheets often contain proprietary or confidential information, business logic and trade secrets. Ten participants anonymised or removed information from their spreadsheets before sharing them and could have stripped valuable research data. To avoid this, we asked participants to preserve the structure, layout and style of their spreadsheets while anonymising them.

Fourth, interview studies are limited by the size and diversity of the sample. We followed recommendations from prior work to interview between 12 and 25 participants [22], until saturation. We aimed to recruit a demographically diverse sample. We ensured that participants were from different fields (e.g., history, political science, computer science, medicine, insurance) and various expertise levels in order to increase the likelihood of relevant findings being mentioned by at least one participant. To improve the generalisability of this study, our protocol is publicly documented (https://osf.io/enwxj/), to aid replication with different samples.

5 INTERVIEW 1 RESULTS: THE COMPLEXITY OF STRUCTURING DATA IN SPREADSHEETS

We discovered that the user's choice of a certain level and type of structure is subject to numerous tensions that the user must resolve, to varying degrees of success, to arrive at a satisfactory solution for their spreadsheet. This can be conceived of as a 'tug of war' between pushes and pulls, visualized in Figure 2. There are factors that push the user's spreadsheet towards structured data, but which are met with countervailing forces from barriers to structuring data. At the same time, there are factors pulling the user's spreadsheet towards unstructured data. In the following sections, we explain these factors in greater detail.

Before we proceed it is important to define what we mean by 'structure'. We found that participants can refer to three distinct concepts when talking about structure. The most common connotation of 'structured' is highly regular, tabular data that is in a relational schema and could therefore be (or already is) stored in a formal table. Compared to this, data that is irregular or regular but non-relational can be considered unstructured. The second connotation of 'structured' is laid out with respect to some operation of interest to the user. Thus in this view even irregular data, if it facilitates some operation (e.g., comprehension) can be considered structured, and regular data can be considered unstructured if it hinders this operation. Finally, 'structured' can connote the effort invested in the deliberate layout of the data. More structure implies more conscious decisions and effort, and conversely when less effort has been invested the data is viewed as less structured, even if the actual layout is regular and tabular. In the interviews, participants used the word 'structure' to mean all these things freely and interchangeably. In the following section, we provide context and examples to clarify which connotation of structure is being used.

5.1 Factors pushing towards structured data

We found four major factors that caused users to engage in intentional data structuring activities: spreadsheet features which benefited from or required a particular structure, influences from auxiliary spreadsheet tools, needs of the audience and the collaborators, and the desire for error prevention.

5.1.1 Using spreadsheet features that benefit from structure. Spreadsheet tools such as formulas, sorting, filtering, pivot tables, and add-ins (e.g., Power Query) pushed participants (n=16) towards deliberate structuring. P9 arranged data to accommodate the assumptions of the `VLOOKUP` function: "*I'm a big fan of* `VLOOKUP`*, I always structure my spreadsheets so that I can do what I need to.*" P13 added structure to "*sort by zip code*". Similarly, P4 structured data to use pivot tables; P8 used structured tabular data to create graphs; P1 structured their data to use the `LEFT` function; P2 used add-ins in Microsoft Excel such as Power BI, which required structured tabular data.

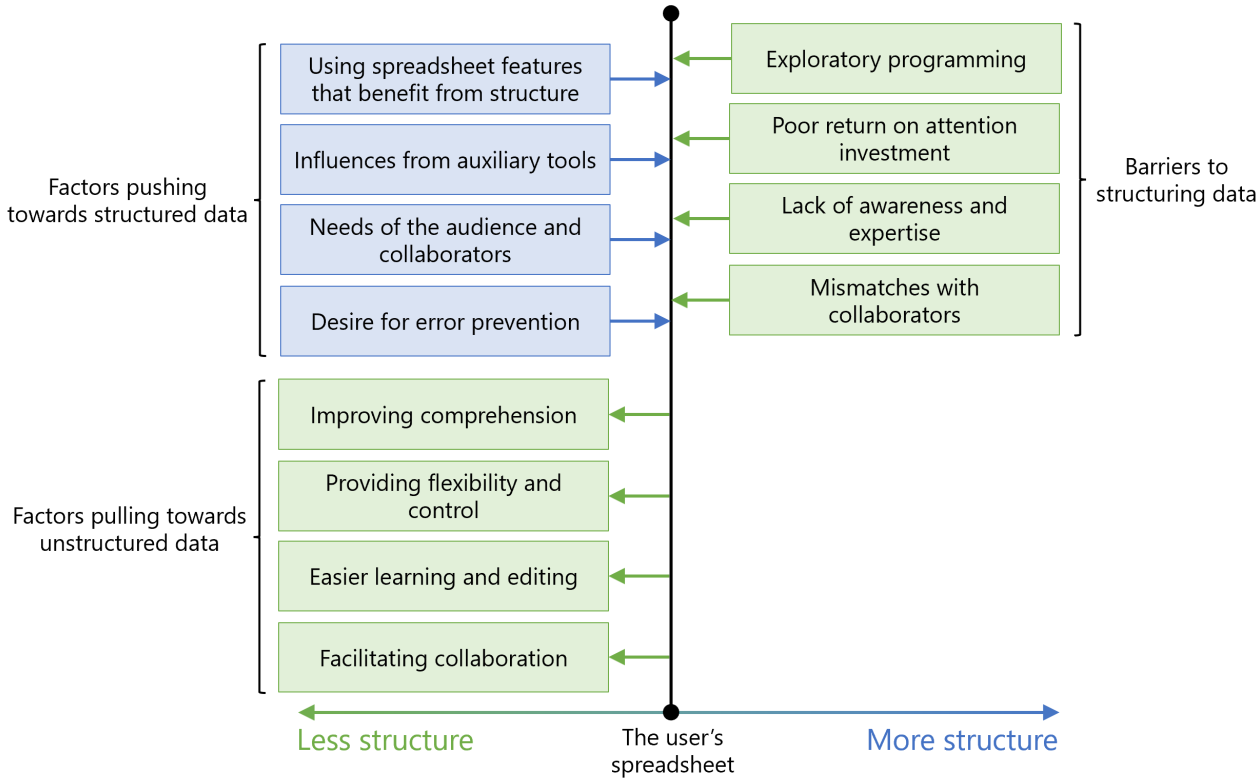

Figure 2: The 'tug of war' of spreadsheet structure. The user's spreadsheet is subject to a complex set of pushes and pulls that the user must resolve to arrive at a type and level of structure that is suitable for the task at hand.

5.1.2 Influences from auxiliary tools.

Spreadsheets served specific roles in a pipeline: Participants (n=13) used spreadsheets along other auxiliary tools. Some used spreadsheets for data cleaning or quick prototyping before using analysis tools (e.g., Python, R, Stata). P10 and P15 used spreadsheets as a prototyping tool to create proof of concept graphs or analyses, getting feedback from their collaborators before modeling in Python or R. P10 explained: "*[In spreadsheets I can] look at some trends, see how some certain parameters influence my result. I could do this by writing a Python script, for example, and using Matplotlib, but it's just easier and quicker to just open a .CSV file, import it into [a spreadsheet], filter the data, and then just plot a graph.*" Similarly, P19 and P11 used spreadsheets to clean .CSV files before using more advanced analysis tools. Some participants (n=5) used spreadsheets to export graphs and tables. Others (n=4) used spreadsheets as a database or data visualization tool (Figure 3) after using auxiliary tools (e.g., SPSS statistics).

Imported data was structured: Participants (n=8) imported data that was pre-structured by other tools. For instance, P4 explained: "*What I'm exporting is already the mandatory fields on that Salesforce system [...] by the time I have exported this, it's already in an almost ready-to-go format for analysis.*" P2's exported data out of survey software was always structured. P2 explained: "*My team created the survey [in] SPSS and it was collected here with mobile phones and tablets [...] and the raw data is then downloaded into Excel.*" Similarly, P15 and P21's exported data out of Microsoft Forms, an online survey creator, was in a structured Microsoft Excel table. Other participants used datasets from database management systems (e.g., Microsoft Access), graphing software (e.g., GraphPad Prism), recruitment services (e.g., Prolific Academic) and educational services (e.g., AP Central).

Twelve participants imported data from collaborators, open data and scraped data. P15 received pre-formatted investment data from co-workers to analyze. P19 imported datasets (see Figure 5) from government open data (e.g., US Department of Education) while P1 used web scraping to build datasets from online dictionaries. Some participants (n=3) used pre-existing spreadsheet templates which had a default structure.

Structuring data to export it to auxiliary tools: Participants (n=13) structured their data for export into auxiliary tools such as analysis tools (e.g., IBM SPSS, Stata, Python, R) (n=7), word documents (n=3), visualization tools (n=2) and databases (n=1). P15 explained: "*My immediate focus is on trying to make [...] the file very simple and clean in order to export it as a .CSV so that some of our data pipelines can pick it up easily.*" Similarly, both P2 and P17 structured data for export into a visualization tool. P17 said: "*I'd like to copy and paste the data from Excel to Power BI without doing any additional work.*"

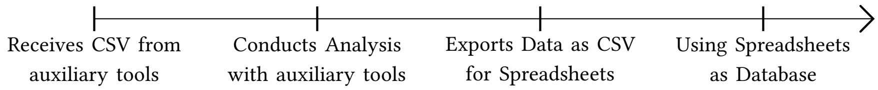

Figure 3: Journey map illustrating P2 and P3's use of spreadsheets to store and organize data analyzed by auxiliary tools.

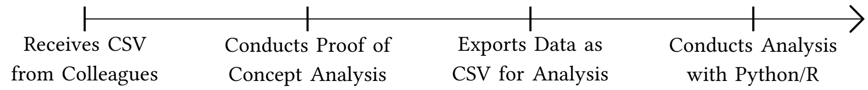

Figure 4: Journey map illustrating P10 and P15's use of spreadsheets to conduct proof of concept analysis.

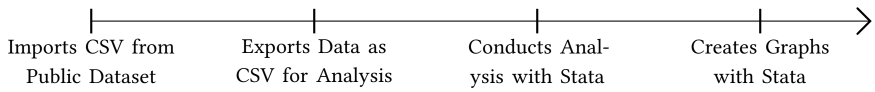

Figure 5: Journey map illustrating P11 and P19's use of spreadsheets to clean data before exporting it for auxiliary tools.

Participants exported summary graphs and tables for company reports and presentations. For instance, P4 exported summaries into Word documents for their managers: "*People are not going to flip through the 600 rows here. They would just want to see it in a Word document [...] Most of the senior management or executives [...] they don't like so much if you just send them a spreadsheet for them to read through. They would like to have someone summarize it and pick out the highlights.*"

Participants (n=4) used dataset formats (e.g., .TSV and .CSV) to export data for use in Python or R programs. P3 exported their spreadsheet data into R to compute statistics on psychological survey data. Other participants (n=3) exported datasets for statistical (n=2) and qualitative analysis software (n=1). For instance, P19 exported their data into Stata to compute summary statistics for dual or concurrent enrollment at US public schools (Figure 5).

5.1.3 Needs of the audience and collaborators.

Usability and reputation: Participants (n=15) structured their spreadsheet in consideration of the audience. Participants were concerned both about the usability of the spreadsheet by others as well as how it would reflect on their reputation. Spreadsheets shared publicly (n=5) and with co-workers and supervisors (n=10) had more structure. For example, P18 structured a scientific spreadsheet for publication: "*This is something that I'm going to end up publishing and people are going to read. You need it to be as structured as possible.*" Similarly, P2 added formatting because "*it might be published online [...] for anyone to download*". P1 cleaned and structured their spreadsheet for publication on GitHub.

Moreover, spreadsheets shared in a work context with work colleagues, managers, clients, compliance, etc., had more structure. P4 and P8 both added structure because of company audits. P4 explained: "*We could get audited on these processes by one of the external auditors or internal audit team*". P5 structured their spreadsheet invoice before sharing it with their client. Other participants such as P17 structured sheets before they were shared to senior members of their company: "*I had to be ready to share my spreadsheet with any member of the leadership in my company, and they should be able to read it without me explaining what it was saying*".

Conversely, participants (n=5) who kept private spreadsheets added less structure and formatting than usual. P18 used less structure on a financial spreadsheet because it was private: "*This is not for anyone else to understand but me. If I'm trying to make someone else understand this, yes, maybe I would have been a little bit more organized.*" Similarly, P10 and P13 said their private spreadsheets usually lack structure and formatting compared to their shared spreadsheets.

Using a 'master' workbook to conduct core analysis: Several participants (n=12) used a so-called 'master' workbook. '*Master*' workbooks contained sophisticated analysis that was often not annotated. Moreover, they contained unstructured data involving tabular and raw data that could not easily be understood by collaborators. Most spreadsheet (n=10) users kept their '*master*' workbook private and instead shared tailored copies when prompted. For instance, P2 derived a redacted version of their workbook when they were required to share it. P2 explained: "*I have a previous file which is the percentages and numbers but without all of my workings, so I would send the data file, the percentages, and the numbers to the client. Then, once I finish, I delete the sheets or I'll save it somewhere separately away from my master sheet.*"

Other participants used separate sheets to display summaries of their data or analysis, keeping their 'master' sheets intact. We use the word 'sheet' here to mean an individual 'worksheet' commonly represented as a tab in commercial spreadsheet applications; a spreadsheet file is a 'workbook' consisting of multiple sheets. For instance, P6 builds graphs and tables out of their data in separate sheets when prompted to share their analysis.

Separating data to manage collaboration: Participants (n=4) split data into separate sheets to allow multiple collaborators to work separately. P2 instructed their co-workers to conduct analysis on their own sheets: "*They can have their own copy of it and do their* *own analysis separately.*" Similarly, P15 uses six different sheets with other co-workers on a collaborative spreadsheet, creating "*cross-references between the sheets*" which allows them to reference data from other sheets without disrupting each other's workflows.

5.1.4 Desire for error prevention. Participants (n=12) added structure to reduce or prevent errors when adding data. For instance, P6 used Microsoft Excel's Text Import Wizard to prevent the conversion of gene names into dates. P6 explained: "*Somehow some gene would have the same format as a date, if you input it here this will change the gene into a date format.*" Similarly, P15 added structure to prevent character encoding errors when importing data: "*We've got some people in France, some people in Germany, some people in the US. [...] If there are any special characters, those display differently for them. That is a major issue.*"

Other participants introduced structure to prevent errors through data validation rules. For instance, P12 added structure to use "*error checking, and the formula auditing tools.*" P8 and P20 added data validation rules to restrict the number of allowed entries on a cell. P6 created auxiliary tables to support data entry: "*If I want to restrict the column in entry to some specified values, I would have to create another sheet where I have the available entries that are permitted to enter here.*"

Some participants (n=5) stored an unedited version of their data to prevent loss of information. For instance, P12 said: "*It is easier to have one sheet for raw data, particularly if I try to do different things with it [...] When I get raw data, I copy it right away in a separate sheet that I don't touch [...] to keep that data clean.*" P19 and P21 also create backup copies of their raw data before making any changes.

5.2 Barriers to structuring data

In many cases, despite desiring and understanding the benefits of structured data, users experienced barriers to structuring their data. We found four major barriers to structuring data: exploratory programming, poor return on attention investment, a lack of awareness and expertise, and mismatches with collaborators.

5.2.1 Exploratory programming. Working with unanticipated or exploratory data was a barrier to structuring data. Participants (n=5) who added unanticipated data over time could not easily structure their spreadsheets. P5's data was repeatedly re-structured due to unanticipated requests from clients: "*One of the reasons it would be this way is, let's say I start working, I do this table and then I do this later on. Then the client asks me to include the hours so I just add the hours. Then the client asks me to add how much he has paid and I add them here.*" Similarly, P16 was unable to easily structure a spreadsheet because "*new information was being put together for it*". Participants (n=4) conducting open-ended analyses couldn't anticipate what structure would be useful. P18 said: "*I probably would have started off in the more table-structured format. I didn't know what I was looking for, necessarily. I just wanted to put all the information I had in my head.*"

5.2.2 Poor return on attention investment. Participants (n=5) said they did not have enough time to structure their data. P9 explained: "*I didn't have enough time to make it as nice and neatly organized as I would have done if this was the main thing that I had to focus on. I quite often do this as much as it pays me.*" Similarly, P20 said they were "*stressed at the time*" they created an unstructured grade calculation spreadsheet. P4 lacked time to go back and re-structure historical company spreadsheets that were still in use. Participants (n=3) said they didn't have enough motivation to introduce structure. P7 said: "*We tried to make a family budget for a year [...] I made it 10 minutes, and I was like, 'This is more time than it's worth. Let's just go back.*"' P20 was not motivated to structure their video game investment spreadsheet because it was not vital to their task.

Participants (n=15) did not see benefit in structuring spreadsheets that were not used often or had little data. P8's data was unstructured because it was edited "*only two times per month*". Similarly, P20's grade calculation spreadsheet was unstructured due to editing it only "*once at the end of each semester*". Moreover, P12 didn't structure a personal spreadsheet because it only had 20-30 entries: "*If I had a list of every apartment rental in Boston, structuring would be really useful, but I'm looking at 20 some. I don't have enough data that I'm that overwhelmed about.*"

Participants (n=7) found formal tables in spreadsheets to have a poor value proposition in comparison with simply using a tabular layout on the flexible unstructured grid. This was due to formal tables having unclear advantages, requiring learning, limiting flexibility, and lacking necessary features such as column-based operations.

Need for column-based operations: Participants (n=8) expressed the need to apply operations such as conditional formatting, data validation, formula authoring, and grouping on columns in structured tables. P15 wished to define formulas in column headers: "*Debugging Excel is difficult, so if I could just define the formula in the header and then have it pop-up down here, like change the formatting to show, "Hey, this is a computed column. Don't change this" or maybe even it won't let me change it. That would be great.*" Similarly, P3, P10 and P14 expressed the need to group by header values. P10 said: "*Aggregating would be nice. If you could just create groups, say you only want to see or you want to group the data by customer.*" P8 desired column-based conditional formatting: "*That's something that could be set up at the column level as well, it'd be really nice to define that in here rather than having to do that by selecting the columns and then going through conditional formatting.*"

5.2.3 A lack of awareness and expertise. Not being aware of or knowing how to use spreadsheet features was an obstacle to structuring data (n=7). P11 said they often want to use structuring features but they are unsure if they exist: "*Sometimes you don't even know that you don't know, so you don't know that there is a function to solve what you're trying to do. The formula function is really useful, but you have to know what to search for.*" P11 was frustrated when trying to discover new features in Microsoft Excel because it was difficult to know where they were located in the "ribbon" (a tabbed graphical menu in Microsoft Office). Other participants (n=3) such as P9 described the process of discovering features time consuming: "*What could be improved is the amount of time that I get to spend on really getting to know all of the features because I know there's lots of features in there that I would find very helpful.*" Similarly, P14 often had to "*click through a million things*" or "*watch a video*" to discover new features.

Participants (n=6) cited a lack of expertise in using spreadsheet features (e.g., pivot tables and reporting, dynamic arrays, advanced conditional formatting, data simulations) which prevented them from structuring data. Some participants (n=3) found pivot tables difficult to use and confusing. P17 was frustrated after being unable to understand how pivot tables work: "*Pivot tables are the most complicated [...] I am a software engineer, and I don't understand pivot tables.*" Similarly, P2 said they were taught how to use dynamic arrays in spreadsheets but they found them to be "*too complicated*".

5.2.4 Mismatches with collaborators. Collaborators with different priorities (n=5), unstructured templates (n=5), and different technical skills (n=2) were barriers to structuring. Spreadsheets used across teams with different priorities and goals were subject to compromises in structuring. P4 explained: "*In a large organization quite often you've got junior self-inputting data. Then you've got other teams doing analysis with the data [...] that analysis or report goes to senior management. There are different processes in the chain. It's not often one person seeing it from start to finish. Structuring quirks or annoyances can build up.*" Moreover, P7's collaborator lacked the technical skills necessary to maintain the structure: "*When I was training my replacement, they didn't know how that conditional formatting worked. Within a month or two it was just gone. They could still put the data in, but it wasn't really as helpful as it was before.*" Some participants (n=5) used unstructured collaborator templates. P18's template was provided by their advisor: "*Someone else created it and I've just been working on it. Someone else said this is how the format should be.*"

5.3 Factors pulling towards unstructured data

Most participants (n=20) reported advantages to unstructured data: improving comprehension, providing flexibility and control, enabling easier learning and editing, and facilitating collaboration. In this section we use 'structure' primarily to mean the first of the three connotations discussed at the start of Section 5, i.e., highly regular, tabular data in a relational format. Thus, this section discusses factors that pull data away from such regular structures.

5.3.1 Improving comprehension. Irregular formats and layouts improved comprehension and readability (n=19) through coloring, annotations and spacing (e.g., empty cells, rows, and columns). Participants frequently used colors to improve comprehension and understanding. P7 used colors to make a military equipment tracking spreadsheet more readable: "*In the Army, colors are life. Everybody understands red, green, yellow, black. If you can find a way to present your data with a color scheme that shows very quickly the status of whatever you're responsible for, that's helpful.*" P14 said that colors allow them to visualize their library spreadsheet "*more clearly*". Unstructured sheets facilitated comprehension by allowing participants to combine many sources of information, with varying types and levels of structure. P16 explained: "*I have everything laid out [...] to ensure that I can see everything in one place.*" Similarly, P13 and P19 said that spreadsheets allowed them to visualize multiple sources of information in one sheet.

5.3.2 Providing flexibility and control. Unstructured data provided a sense of flexibility, control, autonomy and ownership for many participants (n=16). For instance, P2 said: "*One of the reasons I use Excel in partnership with the other programs is that I have the flexibility to just cut and paste stuff all over the place with formulas where I like. It's freedom to put things where my mind wants to put them.*" P18 used unstructured spreadsheets for budgeting instead of using dedicated tools (e.g., Mint budget tracker app). P18 explained: "*I have tried [Mint] for a couple of months and I just went back to Excel. I feel I have more control, I can add and change things where I want. Mint is more rigid [...] if something doesn't really fit into a category that Mint had, it is an issue.*" Moreover, P7 said they appreciate the "*freedom*" and "*flexibility*" of being able to lay out data and coloring on the grid.

5.3.3 Easier learning and editing. Participants (n=18) described working with unstructured data as quick and learnable.

Some participants (n=3) such as P6 described unstructured spreadsheets as quick to use: "*That's why we just have some form of formula and equations. It's just to facilitate the process as we do experiments. So it's a little bit faster.*" P10 used an unstructured sheet to "*calculate quickly how much in total a trip would cost*". Participants (n=3) found unstructured data more reusable. P16 easily calculated house bills through reuse: "*You can make a sum of [amounts] then when you copy and paste it, that template is already there, so you just have to make revisions to the amount.*"

Participants (n=4) also found spreadsheets easy to learn without having to wade through documentation. For instance, P11 found unstructured spreadsheets easier to learn compared to project management tools: "*It always seemed quite a barrier to entry with those project management software, like Asana, or whatever. I'd have to learn a lot to get the functionality.*" Similarly, P18 described unstructured spreadsheets as "*easy to use*" for calculations, and P9 found that "*looking down a list is very easy*".

5.3.4 Facilitating collaboration. Participants found unstructured spreadsheets (n=6) to be effective at collaboration due to being ubiquitous and familiar to collaborators. P9 described unstructured spreadsheets as a usable data entry tool for hospital patients: "*Giving people a spreadsheet to fill in is a lot easier than getting them to complete a web form for example [...] it's what people are used to and it's what is most likely to get them to do it.*" P15 heavily uses spreadsheets for workplace collaboration due to their ubiquity: "*The value of Excel for me is that everyone has it. It exists on all the computers in the workplace. If I need to get an SME to review something, they'll probably send me an Excel file.*" Similarly, P17 described spreadsheets as the "*universal language of business*" due to being able to share them with any employees. P8 found that collaborators can interact with "*spreadsheets fairly quickly*" due to their portability and familiarity. P16 found that spreadsheets are "*helpful for making a system that's easy to follow*".

5.4 Coping with unstructured data

Despite the benefits, participants experienced several limitations when using unstructured data. Unstructured data was difficult to use with tools such as charts, sorting, filtering, pivot tables and formulas. Some participants were unable to perform basic arithmetic operations or data re-arrangement.

P18 couldn't sort or filter entries due to the lack of tabular schema: "*I could [import my bank statement] but it's not organized by column.*" Moreover, P4 was unable to use the `VLOOKUP` function due to unstructured data created by collaborators: "*The* `VLOOKUP` *doesn't* *work because of the structuring and the inconsistency in some of these. You can't just do it.*" Unstructured data could obstruct certain analyses. P16 was not able to efficiently search a library spreadsheet due to unclear structure and attributes that were not predefined.

To work around these limitations, users developed several coping mechanisms, including manual work, using auxiliary tools, getting help, and continuous experimentation with structure.

5.4.1 Manual work. Participants (n=8) performed manual and repetitive tasks in order to address the limitations of unstructured data, such as manually computing formulas, manually creating pivot tables, manually sorting and filtering rows, manually applying data validation and conditional formatting.

For example, P1 spent hours manually creating pivot tables for unstructured data provided by their employer: "*I couldn't use Excel to make those, but [...] I know how to make a pivot table manually. It would take two hours to make multiple pivot tables because my job was asking me to do it.*" P12 and P17 also computed pivot tables manually. Similarly, P3 "*manually searched through*" 500 rows to remove duplicate entries from unstructured survey data.

P9 manually applied data validation rules to 480 unstructured columns: "*We structured this poorly, in that we've not assigned a unique identifier to each entry so I can't just send a new spreadsheet with a unique identifier. I have to add columns of validation to the existing data. I have to, 60 times 8 times, add validation to a set of columns.*" Similarly, P6 and P20 manually applied conditional formatting to unstructured data.

5.4.2 Using auxiliary tools. Participants (n=15) used auxiliary tools to be able to compute formulas, perform data analysis and generate pivot tables for unstructured data. P1 combined entries from two unstructured tables using Python because they couldn't easily use existing functions (e.g., `VLOOKUP`). P1 explained: "*I will Google things that say that I have to use something like a* `VLOOKUP` *search or whatever. I find all those functions confusing. I just make my own Python scripts, do that, then put it back in Excel.*" Similarly, P3 used R, a programming language to conduct computations for unstructured data. P3 explained: "*I use R for sums, averages, standard deviation, basic statistical functions. The more in-depth stuff ends up being much easier to do in R.*" P2 used dedicated statistical software to derive pivot tables: "*The tables that we created are kind of like pivot tables. I'm much quicker to do it through SPSS, than to do it through a Power Pivot table.*"

Moreover, P1 was frustrated with manually creating pivot tables (Section 5.4.1) and eventually used auxiliary code to generate them: "*It was a very horrible experience with Excel that I just ended up Googling. I eventually found multiple chunks of tiny codes to use from Stack Exchange to create my own automatic real CSV file, create an Excel sheet from that, made a pivot from that thing.*"

The formal programming languages in these auxiliary tools almost certainly require structuring and shaping before unstructured data can be used. The key observation here is that the user perceives the benefit of performing this structuring *outside* the spreadsheet to exceed the benefit (or have lower cost, or both) of performing the structuring *within* the spreadsheet application.

5.4.3 Getting help. Participants sought help online or from friends and colleagues when faced with limitations.

Participants (n=12) used online search and video tutorials to get support. For example, P21: "*I almost assuredly went to some website that said how to create a pivot table, looked at that very quickly.*" Similarly, P11 said: "*I Googled, 'How do you make a drop down menu,' and it took me to the data validation.*" P11 also searched for unstructured data "*workarounds*" on spreadsheet forums, and P5 and P7 looked for spreadsheet video tutorials on YouTube.

Participants (n=4) asked friends and colleagues to help with structuring their data. For example, P14 asked friends to help in running computations on an unstructured list of books: "*I have two of my really good friends who are software engineers that I ask for assistance.*" P16 asked work colleagues for help when faced with limitations: "*At one point, I didn't know how to sort using Excel. Someone was like, 'Hey, why are you doing that manually?' and taught me the process. Even basic things like using some functions, I don't know any advanced functions. Usually, it's only when someone's watching or someone's seeing me put all this extra effort into something that they're able to point out other ways to it.*"

5.4.4 Continuous experimentation with structure. Participants (n=3) attempted to introduce structure to their data through trial and error. For example, P12 added structure to their data through trial and error to use `VLOOKUP`: "*Sometimes I'm successful at using* `VLOOKUP` *and sometimes I'm not. You just mess up enough times, you're just kind of go like, 'Okay, what if I do this instead?' I can work my way through it.*" Similarly, P5 explained their experimental approach to structuring: "*I don't exactly follow the regular conventions or the guidelines. I haven't really learned how to structure. I've learned it through just trial and error, just testing stuff.*"

Participants (n=7) worked on finding the right balance between structure and flexibility when working with spreadsheet data. On one hand, structured tabular data allowed participants to perform calculations, derive summaries and use arrangement features (e.g., sort and filter). On the other hand, unstructured data was more flexible.

For example, P6 added more structure when there is a need for calculations: "*We just want to write down the experimental steps that we're going to perform in sequential order. There's not a need for a table format. Then, at some point, we want to perform very basic calculations. That's where we have a little bit more structure in terms of table.*"

Similarly, P11 added unstructured data near charts and graphs to improve comprehension. However, they rearranged their data in a tabular format when they needed to make pivot tables. P9 resisted adding structure until it became a major barrier: "*I'll use it a few times, and if it gets annoying, then I will take the plunge and fix it up. I guess the disadvantages really are if it's labor-intensive to use when having it better formatted would be less labor-intensive to use.*"

P14 manually tweaked an arithmetic formula to account for unstructured descriptive text found in a table column: "*I realized that the [extra descriptive rows] ruined it for me, so I had to figure out a way. While I could have really coded it and figured it out differently, it made a lot more sense just do a manual minus eight, and that way the problem was fixed and it was good enough.*"

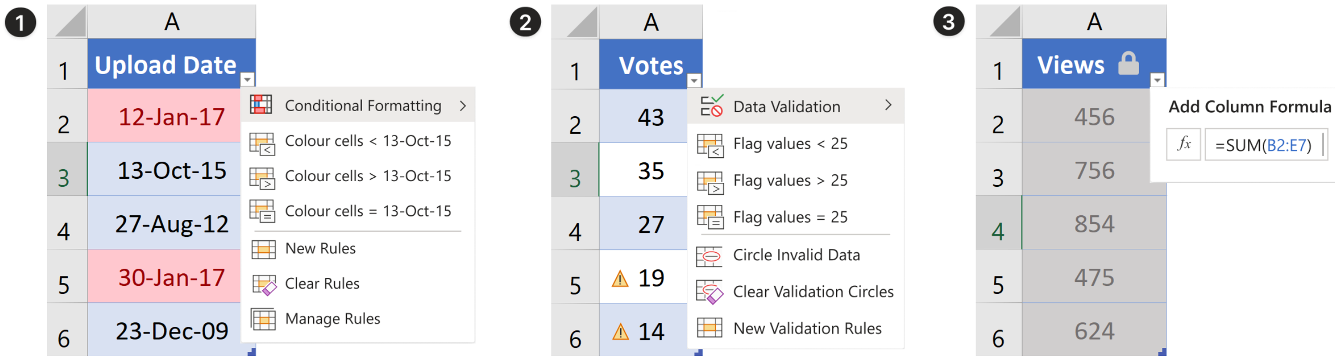

Figure 6: (1) Column conditional formatting prototype: dates before 13-Oct-15 are coloured red. (2) Column data validation prototype: values less than 25 are flagged. (3) Column header formula prototype: summation formula added to cells in column.

6 INTERVIEW 2: CAN WELL-DESIGNED FORMAL STRUCTURES REPLACE THE FLEXIBLE GRID?

In this study, we explored whether augmenting formal spreadsheet tables with affordances that were sensitive to the barriers and constraints to structured data experienced by end-users could meaningfully change the value proposition of formal structures in spreadsheets, and motivate more people to use them.

This was not an evaluation of specific features for tables and their utility. Rather, we asked whether improving the capabilities of database-style tables in line with empirically observed user needs would significantly lessen the *comparative utility* of the flexible grid. Thus we conducted *scenario-based interviews*, wherein participants are shown materials that help them understand a future design scenario, and invited to reflect on them, a method previously applied for understanding proposed spreadsheet features [10]. It bears some similarity to the method of design-led inquiry [108], albeit in a more condensed time frame.

6.1 Click-through prototypes

We observed an interesting opportunity for designing around the *table column*, which seemed to be a highly common unit of operation for participants (Section 5.2.2). Rather than talk about tables as entire structures, we noted many instances where participants referred to desired properties of and actions on individual columns, which were not well-supported by formal tables, and participants were therefore dissuaded from using formal tables. We identified four common operations to be built into table columns, and created click-through prototypes for each. Concretely, the experiences prototyped were:

(1) *Column value grouping:* this depicts the ability to group rows by column entries from the column header, creating a result similar to a pivot table or the SQL `GROUPBY` operation. Category sorting, category calculations and nested grouping are also visible. A simplified vignette is presented in Figure 1.
(2) *Column header formula:* this depicts an interaction where a formula can be written in a column header and automatically propagated to all the cells in the column. By disabling ('graying out') editing from within the column cells itself, this interaction mode enforces a single editing location for all formulas in the column (Figure 6-(1)).
(3) *Column data validation:* this depicts an interaction where data validation can be applied to an entire column. Data validation checks whether values entered into the cells fulfill certain constraints, e.g., are numeric (Figure 6-(2)).
(4) *Column conditional formatting:* this depicts an interaction where conditional formatting can be applied to an entire column. Conditional formatting controls cell appearance based on a rule, e.g., coloring all cells with values less than zero red (Figure 6-(3)).

Furthermore, we presented participants with four scenarios corresponding to degrees of 'opinionation' that the spreadsheet application might have about encouraging the use of formal tables. These provided a concrete basis for participants to reason through the implications of working with formal tables, table column operations, and their likely trade-offs. The scenarios helped participants extrapolate their shallow interactions with the click-through prototypes to the potential deep impact on their day-to-day work, regardless of the degree to which they already used formal tables. These were the four scenarios (simplified vignettes in Figure 7):

(1) *Explicit tables without suggestion*: This scenario corresponds to the current state of major spreadsheet packages, which require an explicit conversion step from unstructured data in the grid, to a formal table (e.g., in Microsoft Excel, this is done by selecting the grid range to be converted and pressing 'Ctrl-T'), and there is no suggestion from the application about when to do this.
(2) *Explicit tables with suggestion:* This, like the previous scenario, maintains a clear user-triggered conversion step, except that the spreadsheet application uses heuristics or a machine learning model to suggest that the user trigger the conversion.
(3) *Automatic tables:* This scenario uses heuristics or a machine learning model to detect whether the user has a table-like structure in the grid, and automatically converts it without user intervention. Thus the user gains the use of beneficial table features such as sorting, filtering, and column operations, without having to perform manual conversion.

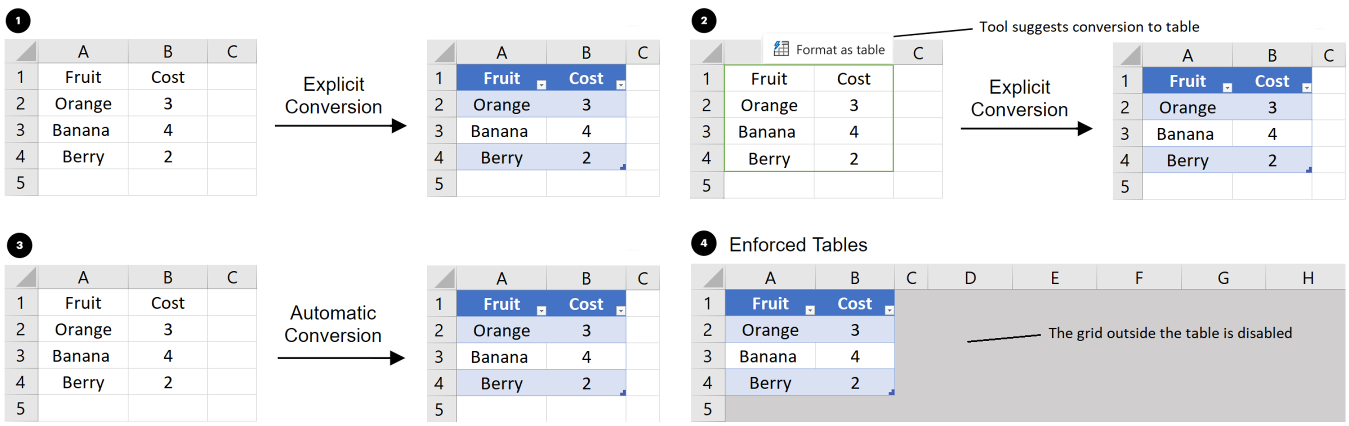

Figure 7: Four scenarios of 'opinionation' that a spreadsheet application might have regarding the use of formal tables. (1) Explicit conversion without suggestion. (2) Explicit conversion with suggestion. (3) Automatic tables. (4) Enforced tables.

(4) *Enforced tables:* In this scenario, there is no unstructured two-dimensional grid. The user can only instantiate structured tables and data must reside in these tables. This is the approach taken by some contemporary applications, which are promoted as having a spreadsheet-like interface (e.g., Airtable[2]).

While these were click-through prototypes (i.e., with very limited interaction), they were at a high visual fidelity to increase the realism of the probe experience [39].

6.2 Scenario-based interview protocol

We first briefed our participants about the experiment, and summarized their responses from previous interviews, recapitulating in particular their attitudes towards formal, database-style tables. We walked participants through the prototypes demonstrating the column operations, as well as prototypes demonstrating each of the table 'opinionation' scenarios.

Next, we asked questions based on Hassenzahl and Tractinsky [46], covering the following themes: the perceived usability and value of the features, perceptions of the different table opinionation scenarios, and their relative advantages and disadvantages.[3]

We designed these questions to get participants to critically engage with each scenario and reason through how it would apply to their own work. Because we had previously interviewed these participants and were familiar with their work context, we ensured that follow-up questions were tailored to participants' individual contexts to elicit more grounded and factual responses.

We conducted interviews in August 2021 with 20 of the participants from our initial interviews. The interviews were conducted remotely using Microsoft Teams and Zoom, with an average duration of 58 minutes. Participants interacted with the prototypes by remotely controlling the interviewer's computer. The interviewer did not interfere during the session except to provide technical support if necessary. We recorded audio, video, and took notes.

[2]https://www.airtable.com/
[3]Our detailed protocol can be found at https://osf.io/hj24b/

6.3 Limitations

First, a *present–future gap* is implicit in the evaluation of any prototypes [88]. Ours is a study of '*what might be*' in an attempt to anticipate in the best possible way a future spreadsheet product and its future context. Our prototypes exist firmly in the present world, in specific study circumstances, but their actual use context is partially unknown and in an inherently uncertain future. To mitigate the present–future gap, we adopted Salovaara et al. [88]'s suggested practices: mindset, reflection, replication, and transparency.

Second, participants can be biased towards a technological artifact if they believe it is favored by the interviewer (the 'yours is better' bias [30]). To mitigate this, the interviewer dissociated themselves from the prototypes of this study and focused on obtaining factual, rather than subjective, information [6]. In addition, we used implicit metrics [28] and triangulation [68] to validate the data collected.

7 INTERVIEW 2 RESULTS: THE FLEXIBLE GRID REMAINS IMPORTANT

7.1 Perceived usability of column operations

All participants (n=20) found the *column value grouping* features useful and usable. P17 said: "*I love that because if I want this organization from data, I either go with a pivot table or go to [structured database software]*" P21 explained how this feature would be an improvement on pivot tables: "*When I [group] with pivot tables, I create a whole lot of copies of the sheet and do one pivot table per sheet because I have no confidence that if I change things in the first pivot table that everything's going to work. This would give me much more confidence that I can just play around with different views, especially clicking on it again to send it back to where it was.*"

Most participants found *column conditional formatting* (n=16) and *Data Validation* (n=15) features to be useful and usable. Two participants did not find either feature useful. P20 found column conditional formatting features easier to use: "*Conditional formatting seemed hard or that I had to research it [to] learn it. It's not as intuitive as it looks. Now, it looks pretty easy.*" Other participants were neutral towards column conditional formatting (n=2) and column data validation (n=3).

Nearly half of the participants (n=13) found the *column header formula* prototype to be useful and usable, citing saving time and reducing errors as the primary benefits. However, some (n=5) found them unhelpful, potentially confusing, and lacking transparency. E.g., P14: "*I think I wouldn't understand it, which sounds very silly, but I think [it] would confuse me.*". Some participants (n=2) were neutral. P17 worried about errors: "*There's also the potential that I entered the wrong formula, right? I'm not really a fan of this one.*"

Unlike column value grouping, data validation, and conditional formatting, which affect only the layout or formatting of the data, the column header formula feature determines the actual data contents of the column. We hypothesize that this distinction may partly have caused participants' relative hesitancy towards this feature, and would make for an interesting future investigation.

7.2 Attitudes towards opinionation scenarios

While the perceived utility of column operations is interesting, their evaluation was not the intent of this study. Recall that our focus was on participants' attitudes towards the *table opinionation scenarios*, which directly inform us about the comparative utility of structured tables versus the unstructured grid.

Two researchers analyzed the transcripts based on the valence (i.e., sentiment) of participants' responses according to a simple closed-coding scheme: positive, neutral, negative, or undeterminable. Author 1 (who conducted the interviews) and author 2 (the principal investigator) independently completed an initial coding of all transcripts. The initial coding had an agreement of 0.61 (average Cohen's kappa coefficient (κ) for all codes in our data). After cross-reviewing coding decisions, clarifying coding rules, and independently re-coding the utterances, inter-rater reliability increased to an acceptable level (average Cohen's κ was 0.82) [71]. The remaining disagreements were individually negotiated and resolved.

The proportion of codes in each scenario is summarized in Figure 8. We present this figure only as a summary of the findings elaborated in this section. This figure should not be read as implying that the relative frequency of these qualitative codes is representative of the distribution in the general population.

7.2.1 Explicit tables without suggestions. In the explicit tables without suggestions scenario, the majority of participants (n=14) were positive. Participants' opinions in this scenario were based primarily on the perceived utility of the table features (previously detailed). There were no objections to the level of automation in this scenario, as with complete manual control and no intelligent intervention, user agency was maximized and there was no potential for disruption to the user's workflow.

Some participants remarked on the predictability, control and agency of this scenario. P20 said: "*I know it's reading [my data] correctly because it's an explicit table. [...] I prefer to convert the table first to make sure everything's working correctly.*" P21 similarly said: "*What I like about [this scenario] is I know that whatever is inside the [table] is going to be acted upon, and whatever is outside of the [table] is not going to be acted upon.*"

7.2.2 Explicit tables with suggestions. In the explicit tables with suggestions scenario, the majority of participants (n=17) were positive, but began expressing concerns, largely about the potential for interruption. For some participants, such as P1, this was due to outright apathy for prompts from the application: "*I would just ignore whatever Excel is suggesting and do what I want.*". Others conditioned their acceptance of suggestions based on various features, such as the ability to undo the conversion (P14: "*I would just press Ctrl-Z if I didn't like what it ended up doing.*"), and the ability to preview the result of the table conversion (P16: "*I like being able to see what it would look like [as a formal table] before [I accepted the suggestion]*"). Still others anticipated the burden to check the conversion had completed correctly, e.g., P10: "*I think it completely depends on how big the data is [...] I wouldn't be able to just scroll through 50,000 entries to check if the conversion is correct [...] For this small table, I assume, yes, it would be helpful.*"

7.2.3 Automatic tables. In the automatic tables scenario, the concerns that began to show in the explicit tables with suggestions scenario became amplified. Far fewer participants (n=6) were positive, their positivity mainly due to the lower anticipated cost of invoking table functionality (P4: "*it will save me quite a few steps*"). Some were neutral (n=4), wary about the fallibility of the conversion heuristics (P20: "*I wouldn't be sure that it has all the data or that it's reading the data correctly.*"; P21: "*Knowing technology, there's going to be times when it gets it wrong. How easy is it to change that and be sure that I've not altered my data?*").

Most were negative (n=9). Often, participants could not conceive of how the system could adapt to their specialized data layouts (P21: "*I put scratchings and all sorts of stuff into other cells [...] they wouldn't make sense as tables, or worse, if they were suddenly included into [my explicit tables]. Then, it somehow screwed up my thing and I've got to undo [...]*"; P2: "*I just want to go into [an adjacent column] here and just type that in. I don't want it to then make [the column] into a part of the table as well. Sometimes I don't want stuff in a table because I want to just move things around, and my brain's just being scatterbrained.*"; P11: "*If you've got missing data it's not that useful.*"; P12: "*Sometimes the things I'm doing are not really meant to be a data table. They sort of are, but sort of not. I might just be putting some numbers and seeing how they interact with each other.*").

P21 re-emphasized the ability to recover from automation errors: "*If it's going to do it automatically, then it would need to ask me, and there needs to be an easy route to don't do it that way, or undo that.*". P15 became concerned about the ability to export data from automatic tables into programming languages (e.g., CSVs): "*If formatting things into a table like this makes ingesting into those programming languages easier, I will force everyone to start using these. If they make them in any way harder, I will explicitly tell everyone to not use these.*"

Many of these participant reactions can be viewed through the lens of trust. Trust is a well-documented challenge of interaction with intelligent systems [56, 86]. Participants viewed potential heuristics for table detection as fallible and therefore potentially untrustworthy, and the strategies they suggested, namely the ability to undo or change the inferred table boundaries, ostensibly improve the perceived trustworthiness of the system. Since 6 participants were positive, there are clearly cases where trust is less of a concern,

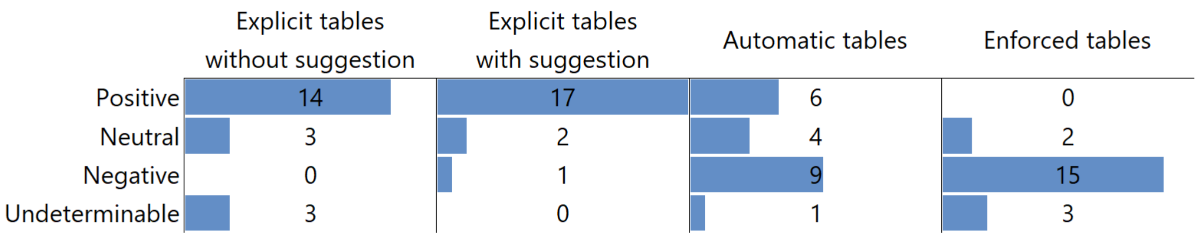

Figure 8: Counts of participants with a certain attitude (positive, negative, neutral, undeterminable) to each opinionation scenario. Participants become more negative as the level of automation increases.

and the nature of the participants' data and their work context may give clues. For example, if a user works with highly tabular data and must repeatedly invoke explicit conversion, the effort saved by automatic conversion may be a worthwhile trade-off for accuracy of the table detection heuristic. We could not probe the issue of trust further within the scope of this study since we were working with non-functional mock-ups, but in future work the role of trust could be better explored in the context of a concrete implementation with measurable performance and known failure modes.

7.2.4 Enforced tables. In the enforced tables scenario, the majority of participants (n=15) were negative. For some, the departure from the flexible grid was too extreme (P14: "*I'm so used to [the grid]. I'd be very overwhelmed if the default was no longer what I'm used to.* "). P16 noted that the flexible grid was a core value proposition of spreadsheets: "*The benefit of Excel is the fact that you can do so much with it. If you try and control the user, then that is stripped from you.*"; P3: "*It's a database now [...] the reason that tools like Excel are useful is that they're flexible. You can have like mostly a table, but also with some other random stuff on the side that you need or whatever makes sense.*"

The reasons participants cited for remaining negative were similar to the automatic tables scenario: losing total flexibility and control, having incompatible source data, difficulty of use by collaborators, not being familiar, difficulty editing data, backward compatibility concerns, performance concerns and data export concerns. Importantly, these persistent concerns demonstrate why the need for the unstructured grid cannot be alleviated simply by improving the value proposition of formal table structures, such as through column operations.

However, some participants (n=2) remained neutral even regarding enforced tables. The reasons these participants were willing to accept enforced tables were either that they consistently used structured tabular data, or that they were (resignedly) willing to adapt their own workflows to fit the limitations of their tools, such as with P1: "*My philosophy is, I just want to do my stuff. I don't care about what your suggestion is, when I do my stuff. If I'm forced to do it, I'm like, 'Okay, well, I just want to get my stuff done.'*"

7.3 The need for both formal tables and the flexible grid

Presenting the scenarios prompted many participants to discuss their preference for working with tabular data within the formalized structures provided by the application, as compared to outside such structures. While some participants favored exclusively using the grid, the majority insisted that both forms of data management are necessary and have their own value. P6: "*Why not have both because if you make it either-or, then the people who used to make the raw data [...] will now not be able to.*" Some expressed it as freedom of choice. P5: "*I would want the freedom to choose between having a Ctrl-T table or not.*"; P13: "*It gives you that choice, and I think that's what people are looking for.*"

P7 referenced their implicit attention investment trade-off: "*if it was a product that I was going to come back and reference regularly, [...] I think I probably would take the extra step and make a table, but if it's like a one or two-time use, then I probably wouldn't. Sometimes I don't need it to look clean and flashy [...] I just need quick, raw, boring, dirty.*"

P8 identified that the flexible grid is essential during an exploratory programming phase, but formalized structures become valuable once the shape of the data is established: "*In the early drafting of [my spreadsheet], I liked just to add the column headers, add a bit of data in, and then later in the process, make it into a table. Whereas I think if it started out as a table, I think it may be harder to make some edits like adding in new headers [...].*"

Recall from the introduction the motivating question for much research into novel spreadsheet structures, namely: is it possible to design a data structure that fits the needs of spreadsheet users to such an extent, that it could replace the unstructured grid and thereby eliminate all the errors that come with it? Our findings suggest that in fact no such structure can exist, due to the dynamic and contextual nature of user needs, that until now have not been documented in detail.

8 DISCUSSION

8.1 Comparison with previous work

The closest precedent to our work is that of Bartram et al. in characterizing "Untidy data" [5]. Several of our findings resonate with theirs. We also observed the pattern of separating a "master" table from further analysis and refinements (Section 5.1.3), albeit not in all participants. We also observed an emphasis on readability and comprehension (Section 5.3.1), constraints posed by moving between tools (Section 5.1.2), exploratory re-coding of data (Section 5.2), and expertise barriers to using formalized structures (Section 5.2.3).

We also build on the findings of Ragavan et al., who observed that spreadsheet authors make efforts to improve the comprehensibility of their spreadsheets [101]. Our findings reaffirm a deep consideration for the readability of the spreadsheet both by the author and by their collaborators, and that private spreadsheets often contained less structure than shared ones (Section 5.1.3). In that paper, the authors speculate that there may be various trade-offs in the spreadsheet authoring process, such as high manual effort of preparing spreadsheets for comprehension and that the structure that suits one task may not suit others. Our findings confirm these speculations and elaborate how such concerns are in tension with others during the authoring process, when it comes to the structure and layout of data.

Similarly, our findings support those of earlier studies. Sarkar and Gordon found that learning in spreadsheets is "informal, opportunistic and social" [92]. We found participants relying on social interactions with colleagues or friends to learn about structuring techniques or to cope with unstructured data (Section 5.4.3).

8.2 New perspectives on the user experience of data structuring

8.2.1 Structure is task-dependent, dynamic, and continuous. Our study shows that "structure" cannot exist in isolation. The user's experience of structure, as we observed, is always with respect to an operation. There is no such thing as "structured" or "unstructured" data in any absolute sense. Thus, for example, the addition of annotative marginalia at around a table can be viewed either as a subversion of the formal table structure that may interfere with the formal operations of sorting and filtering, or as the addition of structure to improve comprehension. Many of these "operations" – formula authoring, exploratory analysis, preparation for auxiliary tools, comprehension – are not database operations in the traditional sense, but rather denote classes of activities of importance to the user.

At any given moment the same spreadsheet can perform as either structured or unstructured, depending on the user and the task at hand. As observed in Sections 5.2.2 and 5.2.3, the decision to apply or not apply certain kinds of structure is subject to a network of influences that includes an often implicit consideration of which operations the user is likely to perform over the lifetime of the spreadsheet, and with what frequency.

Terms such as "structured", "unstructured", "tidy" and "untidy" not only incorrectly imply that structure is a static property of the data layout in a spreadsheet, whereas participants in our study experienced structure as being dynamic and contingent, but they additionally imply a false dichotomy, whereas instead participants in our study experienced their data as being on a continuum of structure.

8.2.2 Spreadsheets are containers for structural tension. Consider the 'tug of war' of pushes and pulls that the structure of the spreadsheet encounters (Figure 2). Viewed in isolation, this can paint quite a negative picture of the experience of structuring data in spreadsheets, with the user faced with the challenging task of resolving these tensions.

However, this situation can also be viewed in the context of the broader journeys that many users undertake between spreadsheets and various auxiliary tools (e.g., Figures 3 and 5). In these workflows, there are tensions and requirements in the workflow that can only be contained, absorbed, and resolved within the versatile environment of the spreadsheet. An important perspective our data shows is that spreadsheets are an intermediary substrate for ad-hoc data abstractions that enable these ways of working with tools. If a tug of war is an inevitable part of contemporary data work, the spreadsheet is the indispensable rope.

8.2.3 Attention investment trade-offs moderate structuring. An important advance of our study is to provide empirical evidence that users view the costs of structuring as an attention investment trade-off. Blackwell's attention investment model of end-user programming posits that the user weighs any effort invested in authoring their program (in our case, the spreadsheet) against the future payoff of that investment [9]. While this has been shown to be the case in the context of error testing [110, 112], it has only been speculated to be a factor in the problem of structuring data. We found participants explicitly accounting for the longevity of the spreadsheet and the frequency of its use as part of the decision to invest attention in building structure (Section 5.2.2).

8.2.4 Agency and expertise moderate structuring. Another important advance of our study over previous work is to identify the importance of *agency* during spreadsheet use. Agency is a term from cognitive neuroscience referring to "*the experience of controlling one's own actions and, through this control, affecting the external world*" [26]. In this case, the user's sense of agency is with respect to their task, data, and operation of the spreadsheet software. Previous work has observed the need for *expertise*, that is, in order to use a formalized structure such as Excel tables (i.e., "Ctrl-T tables"), the user must acquire expertise in that feature.

While agency and expertise are closely related – expertise is a key factor in creating agency for end-user programmers – they are subtly different. Even when the formal mechanism was well-understood, some participants still felt a greater sense of 'control' over their data when using the flexible grid (Section 5.3.2). For the participants in our study, this primarily stemmed from the fact that formal mechanisms are often opinionated and modify normal modes of operation, such as Excel tables automatically adding rows and columns when data is entered adjacent to a table, which interferes with the user practice of applying marginalia, or automatically using structured reference notation[4] when writing formulas.

Previous work has observed that such opinionation can result in a loss of agency even when operating in consistent and intelligible ways [26]. However, such work has also shown that there may be a threshold degree of automated assistance below which the user does not experience a significant loss of agency, which highlights an opportunity for spreadsheet data structures. Moreover, an opinionated structure may not always entail automated assistance and a consequent loss of agency. For example, the multiple representations approach, as adopted by Calculation View [93], layers a structural abstraction over the grid without affecting any normal modes of operation on the grid.

[4]https://support.microsoft.com/en-us/office/using-structured-references-with-excel-tables-f5ed2452-2337-4f71-bed3-c8ae6d2b276e

8.3 Implications for design and practice

8.3.1 Composable and transient structures. In Section 8.2.1 we discussed that the user's perception of structure is task-dependent, dynamic, and continuous. This has two interesting implications. One implication is that data structuring features in spreadsheets might better align with users' workflows if they had incremental constraints and benefits, and could be composed with each other. Thus a user might want their data to reside simultaneously in an array (which facilitates certain formula operations) as well as in a table column (which facilitates sorting and filtering), choosing only to add structure when the perceived benefits exceed the costs. This is easier said than done. Developing a coherent set of interactions for composable data structures with due consideration of the myriad edge cases would make for valuable and interesting future work; there are likely to be several challenges just in ensuring arrays and tables work together, for instance. Another implication is that no single structure is appropriate in all situations for all users – to the contrary, every structure in a spreadsheet is likely to be inappropriate for many user operations over the lifetime of the spreadsheet. Thus the major design problem is to help users select, deselect, and modify their structure for different scenarios, rather than to identify "better" structures.

8.3.2 Table columns as a user-centric structure. Our findings reveal the table column to be a powerful user-centric data structure. While table columns have always been the locus of sorting and filtering operations, our findings additionally show that they can be appropriated for many other operations such as conditional formatting, data validation, formula authoring, access control, and pivoting.

This may seem superficial, but it is a meaningful departure from programmer-centric data types such as arrays and records, for the following reasons. Table columns display the entirety of the data, i.e., they "*begin the abstraction gradient at zero*" [90], therefore enabling visual inspection and direct manipulation [98] — a finding echoed by Bartram et al. Moreover, the header cell of a table column serves both as a descriptive comprehension aid, as well as a graphical abstraction over the data in the column, and is therefore ideal for situating operations that uniformly affect all cells in the column.

Table column operations are not an entirely new concept. Sorting and filtering through table headers cells is widely implemented in spreadsheet applications. Moreover, data operations such as outlier detection have been implemented at the table column level in applications oriented towards professional data scientists.[5] However, our findings highlight an opportunity to incorporate many more common operations of concern to spreadsheet users into spreadsheet table columns, and our second study suggests that these features would in fact make a meaningful impact on the value proposition of formal tables in spreadsheets.

8.3.3 Opportunities for auxiliary tools. While our study has identified some opportunities for spreadsheet tools that may aid users with the challenges of structuring data, a more important contribution has been to problematize the notion that these challenges can be addressed through the design of spreadsheets alone, by revealing the extent of the network of influences that reach well beyond the spreadsheet tool itself.

[5]https://www.trifacta.com/products/why-trifacta/

There are both technical as well as non-technical opportunities here. For example, scripting tools can be integrated within spreadsheets [65]. Tools that are used as data sources for spreadsheets or data sinks from spreadsheets can be designed to better accommodate different kinds of structure. A tool that expects data to be shaped in a strict relational format can greatly improve the comprehensibility of the spreadsheet being fed into it, if it can account for sub-headers, marginalia and subgroup header rows. While it can be argued that this can make such tools more complex and potentially introduces some kinds of errors, our study and those before it show that the alternative – an unproductive and incomprehensible spreadsheet layout – can be far worse.

8.3.4 Addressing collaborator mismatches. Moreover, several aspects of spreadsheet structuring were influenced by the needs of collaborators and considerations for the audience. When the collaborators' needs and priorities were significantly different from those of the author, this created strong tensions in the kind of data structure needed in the spreadsheet (Section 5.2.4). While Nardi observed that spreadsheet co-authors at different levels of expertise naturally fall into different roles [77], our study shows that when users are called upon to contribute to a spreadsheet whose data structures presuppose more expertise than they have, this can lead to conflict and errors. Our study suggests that while not all collaborator mismatches can be avoided, a relatively straightforward one is a mismatch in user expertise, which can be avoided with careful employee training and collaboration planning. A mismatch in collaborator priorities pulling spreadsheet structure in different directions will almost certainly require negotiation and compromise between the individuals involved, but the tool may be able to help prompt such a negotiation, for example by detecting repeated changes to structure by various collaborators. The tool could then suggest one of the strategies used by our participants to manage spreadsheet collaboration, such as requiring each collaborator to work on an independent sheet .

9 CONCLUSION

We interviewed 21 spreadsheet users from various domains and levels of expertise about their use of structure in spreadsheets. We found several factors that influence the user's choice of structure in any given spreadsheet, including the user's spreadsheet expertise (or lack thereof), the audience of the spreadsheet, needs of collaborators, the use of auxiliary data tools, the potential costs of structuring, concerns around error prevention, considerations of spreadsheet comprehensibility, and a sense of control and agency. Building on previous work that has studied individual aspects of data structuring in spreadsheets, our study is the first to document in detail how these factors may interact.

We found that the table column is a common unit of operation and accordingly created four click-through prototypes showing various spreadsheet operations such as formula authoring and conditional validation applied at the table column level, which we used as design probes in a second study with 20 participants. Through this study we discovered that table column operations are a powerful primitive that improve the utility of tabular structures, but nonetheless cannot be expected to offset the countervailing forces that make the flexible, 2-dimensional grid so useful and popular.

Future work may study in greater detail how these effects vary across different segments of the user population as well as techniques that may infer the user context and suggest an appropriate level of structure. Future studies may also explore in greater detail the nature of collaborative tasks around spreadsheets and the interplay between novice and expert users.

ACKNOWLEDGMENTS

The authors would like to thank Christian Canton, Sara Al Mughairy, and Alfredo Arnaiz for their valuable feedback and discussion at various stages of the project, and Chris Oslund for his assistance with design ideation and mockup development. Additionally, we thank the reviewers for their detailed and helpful feedback, and our study participants for their time.

REFERENCES

[1] 2006. Self-Report Study. In *The SAGE Dictionary of Social Research Methods*. SAGE Publications, Ltd, 1 Oliver's Yard, 55 City Road, London England EC1Y 1SP United Kingdom. https://doi.org/10.4135/9780857020116.n186

[2] R. Abraham and M. Erwig. 2004. Header and Unit Inference for Spreadsheets Through Spatial Analyses. In *2004 IEEE Symposium on Visual Languages - Human Centric Computing*. 165–172. https://doi.org/10.1109/VLHCC.2004.29

[3] Yanif Ahmad, Tudor Antoniu, S. Goldwater, and S. Krishnamurthi. 2003. A type system for statically detecting spreadsheet errors. *18th IEEE International Conference on Automated Software Engineering, 2003. Proceedings.* (2003). https://doi.org/10.1109/ASE.2003.1240305

[4] Sara Alspaugh, Nava Zokaei, Andrea Liu, Cindy Jin, and Marti A. Hearst. 2019. Futzing and Moseying: Interviews with Professional Data Analysts on Exploration Practices. *IEEE Transactions on Visualization and Computer Graphics* 25, 1 (Jan. 2019), 22–31. https://doi.org/10.1109/TVCG.2018.2865040 Conference Name: IEEE Transactions on Visualization and Computer Graphics.

[5] Lyn Bartram, Michael Correll, and Melanie Tory. 2021. Untidy Data: The Unreasonable Effectiveness of Tables. *arXiv e-prints* (June 2021), arXiv:2106.15005. https://ui.adsabs.harvard.edu/abs/2021arXiv210615005B

[6] Michael H. Bernhart, I. G. P. Wiadnyana, Haryoko Wihardjo, and Imbalos Pohan. 1999. Patient satisfaction in developing countries. *Social science & medicine* 48, 8 (1999), 989–996.

[7] Alex Bigelow, Steven Drucker, Danyel Fisher, and Miriah Meyer. 2014. Reflections on how designers design with data. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces (AVI '14)*. Association for Computing Machinery, New York, NY, USA, 17–24. https://doi.org/10.1145/2598153.2598175

[8] Peter Birmingham and David Wilkinson. 2003. *Using research instruments: A guide for researchers.* Routledge.

[9] Alan F Blackwell. 2002. First steps in programming: A rationale for attention investment models. In *Proceedings IEEE 2002 Symposia on Human Centric Computing Languages and Environments*. IEEE, 2–10.

[10] Alan F Blackwell, Margaret M Burnett, and Simon Peyton Jones. 2004. Champagne prototyping: A research technique for early evaluation of complex end-user programming systems. In *2004 IEEE Symposium on Visual Languages-Human Centric Computing*. IEEE, 47–54.

[11] Judith Borghouts, Andrew D Gordon, Advait Sarkar, Kenton P O'Hara, and Neil Toronto. 2019. Somewhere around that number: An interview study of how spreadsheet users manage uncertainty. *arXiv preprint arXiv:1905.13072* (2019).

[12] Judith Borghouts, Andrew D Gordon, Advait Sarkar, and Neil Toronto. 2019. End-user probabilistic programming. In *International Conference on Quantitative Evaluation of Systems*. Springer, 3–24.

[13] Nadia Boukhelifa, Marc-Emmanuel Perrin, Samuel Huron, and James Eagan. 2017. How Data Workers Cope with Uncertainty: A Task Characterisation Study. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 3645–3656. https://doi.org/10.1145/3025453.3025738

[14] Virginia Braun and Victoria Clarke. 2012. Thematic analysis. In *APA handbook of research methods in psychology, Vol 2: Research designs: Quantitative, qualitative, neuropsychological, and biological*. American Psychological Association, Washington, DC, US, 57–71. https://doi.org/10.1037/13620-004

[15] Karl W. Broman and Kara H. Woo. 2018. Data Organization in Spreadsheets. *The American Statistician* 72, 1 (Jan. 2018), 2–10. https://doi.org/10.1080/00031305.2017.1375989 Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/00031305.2017.1375989.

[16] Neil CC Brown and Amjad Altadmri. 2017. Novice Java programming mistakes: Large-scale data vs. educator beliefs. *ACM Transactions on Computing Education (TOCE)* 17, 2 (2017), 1–21.

[17] Margaret Burnett, John Atwood, Rebecca Walpole Djang, James Reichwein, Herkimer Gottfried, and Sherry Yang. 2001. Forms/3: A first-order visual language to explore the boundaries of the spreadsheet paradigm. *Journal of Functional Programming* 11, 2 (March 2001), 155–206. https://doi.org/10.1017/S0956796800003828 Publisher: Cambridge University Press.

[18] M. Burnett and M. Erwig. 2002. Visually customizing inference rules about apples and oranges. In *Proceedings IEEE 2002 Symposia on Human Centric Computing Languages and Environments*. 140–148. https://doi.org/10.1109/HCC.2002.1046366

[19] C N Trueman. 2015. Structured Interviews. https://www.historylearningsite.co.uk/sociology/research-methods-in-sociology/structured-interviews/

[20] Chris Chambers and Chris Scaffidi. 2010. Struggling to Excel: A Field Study of Challenges Faced by Spreadsheet Users. In *2010 IEEE Symposium on Visual Languages and Human-Centric Computing*. 187–194. https://doi.org/10.1109/VLHCC.2010.33 ISSN: 1943-6106.

[21] Kerry Shih-Ping Chang and Brad A. Myers. 2016. Using and Exploring Hierarchical Data in Spreadsheets. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. Association for Computing Machinery, New York, NY, USA, 2497–2507. https://doi.org/10.1145/2858036.2858430

[22] Kathy Charmaz. 2006. *Constructing grounded theory: A practical guide through qualitative analysis.* sage.

[23] Tie Cheng and Xavier Rival. 2015. Static Analysis of Spreadsheet Applications for Type-Unsafe Operations Detection. In *Programming Languages and Systems (Lecture Notes in Computer Science)*, Jan Vitek (Ed.). Springer, Berlin, Heidelberg, 26–52. https://doi.org/10.1007/978-3-662-46669-8_2

[24] Gregorio Convertino and Andy Echenique. 2017. Self-Service Data Preparation and Analysis by Business Users: New Needs, Skills, and Tools. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '17)*. Association for Computing Machinery, New York, NY, USA, 1075–1083. https://doi.org/10.1145/3027063.3053359

[25] Juliet Corbin and Anselm Strauss. 2014. *Basics of qualitative research: Techniques and procedures for developing grounded theory.* Sage publications.

[26] David Coyle, James Moore, Per Ola Kristensson, Paul Fletcher, and Alan Blackwell. 2012. I did that! Measuring users' experience of agency in their own actions. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 2025–2034.

[27] Jácome Cunha, Jorge Mendes, João Saraiva, and João Paulo Fernandes. 2011. Embedding and evolution of spreadsheet models in spreadsheet systems. In *2011 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. 179–186. https://doi.org/10.1109/VLHCC.2011.6070396 ISSN: 1943-6106.

[28] Mary Czerwinski, Eric Horvitz, and Ed Cutrell. 2001. Subjective Duration Assessment: An Implicit Probe for Software Usability. 167–170. https://www.microsoft.com/en-us/research/publication/subjective-duration-assessment-an-implicit-probe-for-software-usability/

[29] Tim Davies and Mark Frank. 2013. 'There's no such thing as raw data': exploring the socio-technical life of a government dataset. In *Proceedings of the 5th Annual ACM Web Science Conference (WebSci '13)*. Association for Computing Machinery, New York, NY, USA, 75–78. https://doi.org/10.1145/2464464.2464472

[30] Nicola Dell, Vidya Vaidyanathan, Indrani Medhi, Edward Cutrell, and William Thies. 2012. "Yours is better!": participant response bias in HCI. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1321–1330. https://doi.org/10.1145/2207676.2208589

[31] R.W. Djang and M.M. Burnett. 1998. Similarity inheritance: a new model of inheritance for spreadsheet VPLs. In *Proceedings. 1998 IEEE Symposium on Visual Languages (Cat. No.98TB100254)*. 134–141. https://doi.org/10.1109/VL.1998.706156 ISSN: 1049-2615.

[32] Elaine Dobell, Sebastian Herold, and Jim Buckley. 2018. Spreadsheet Error Types and Their Prevalence in a Healthcare Context. *Journal of Organizational and End User Computing* 30 (April 2018), 20–42. https://doi.org/10.4018/JOEUC.2018040102

[33] Paul Dourish. 2017. Spreadsheets and Spreadsheet Events in Organizational Life. In *The Stuff of Bits: An Essay on the Materialities of Information*. MIT Press, 81–104. https://ieeexplore.ieee.org/document/7932792 Conference Name: The Stuff of Bits: An Essay on the Materialities of Information.

[34] Paul Dourish. 2017. *The Stuff of Bits: An Essay on the Materialities of Information.* MIT Press. Google-Books-ID: fnPGDgAAQBAJ.

[35] Stuart E. Dreyfus and Hubert L. Dreyfus. 1980. *A five-stage model of the mental activities involved in directed skill acquisition.* Technical Report. California Univ Berkeley Operations Research Center.

[36] Gregor Engels and Martin Erwig. 2005. ClassSheets: automatic generation of spreadsheet applications from object-oriented specifications. In *Proceedings of the 20th IEEE/ACM international Conference on Automated software engineering*. 124–133.

[37] Sheena Erete, Emily Ryou, Geoff Smith, Khristina Marie Fassett, and Sarah Duda. 2016. Storytelling with Data: Examining the Use of Data by Non-Profit Organizations. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing (CSCW '16)*. Association for Computing Machinery, New York, NY, USA, 1273–1283. https://doi.org/10.1145/2818048.2820068

[38] Martin Erwig, Robin Abraham, Steve Kollmansberger, and Irene Cooperstein. 2006. Gencel: a program generator for correct spreadsheets. *Journal of Functional Programming* 16, 3 (May 2006), 293–325. https://doi.org/10.1017/S0956796805005794 Publisher: Cambridge University Press.

[39] Emily Esposito. 2018. Low-fidelity vs. high-fidelity prototyping. https://www.invisionapp.com/inside-design/low-fi-vs-hi-fi-prototyping/

[40] Martin Fowler. 2018. *Refactoring: Improving the Design of Existing Code*. Addison-Wesley Professional. Google-Books-ID: 2H1_DwAAQBAJ.

[41] Thomas RG Green, Marian Petre, and Rachel KE Bellamy. 1991. Comprehensibility of visual and textual programs: A test of superlativism against the'match-mismatch'conjecture. In *Empirical Studies of Programmers: Fourth Workshop*, Vol. 121146. Ablex Publishing, Norwood, NJ.

[42] Thomas R. G. Green and Marian Petre. 1996. Usability analysis of visual programming environments: a 'cognitive dimensions' framework. *Journal of Visual Languages & Computing* 7, 2 (1996), 131–174.

[43] Greg Guest, Arwen Bunce, and Laura Johnson. 2006. How many interviews are enough? An experiment with data saturation and variability. *Field methods* 18, 1 (2006), 59–82.

[44] Alan Hall, Michel Wermelinger, Tony Hirst, and Santi Phithakkitnukoon. 2018. Structuring Spreadsheets with the" Lish" Data Model. *arXiv preprint arXiv:1801.08603* (2018).

[45] Michael J. Handel. 2016. What do people do at work? *Journal for Labour Market Research* 49, 2 (Oct. 2016), 177–197. https://doi.org/10.1007/s12651-016-0213-1 Number: 2 Publisher: SpringerOpen.

[46] Marc Hassenzahl and Noam Tractinsky. 2006. User experience-a research agenda. *Behaviour & information technology* 25, 2 (2006), 91–97.

[47] K. Hassinen. 2021. An Experimental Study of Spreadsheet Errors Made by Novice Spreadsheet Users. (Aug. 2021).

[48] David Hendry and Thomas Green. 1994. Creating, comprehending and explaining spreadsheets: A cognitive interpretation of what discretionary users think of the spreadsheet model. *International Journal of Human-Computer Studies* 40 (June 1994), 1033–1065. https://doi.org/10.1006/ijhc.1994.1047

[49] Felienne Hermans and Emerson Murphy-Hill. 2015. Enron's Spreadsheets and Related Emails: A Dataset and Analysis. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, Vol. 2. 7–16. https://doi.org/10.1109/ICSE.2015.129 ISSN: 1558-1225.

[50] Felienne Hermans, Martin Pinzger, and Arie van Deursen. 2010. Automatically Extracting Class Diagrams from Spreadsheets. In *ECOOP 2010 – Object-Oriented Programming (Lecture Notes in Computer Science)*, Theo D'Hondt (Ed.). Springer, Berlin, Heidelberg, 52–75. https://doi.org/10.1007/978-3-642-14107-2_4

[51] Felienne Hermans, Martin Pinzger, and Arie van Deursen. 2015. Detecting and refactoring code smells in spreadsheet formulas. *Empirical Software Engineering* 20, 2 (April 2015), 549–575. https://doi.org/10.1007/s10664-013-9296-2

[52] Thomas Herndon, Michael Ash, and Robert Pollin. 2014. Does high public debt consistently stifle economic growth? A critique of Reinhart and Rogoff. *Cambridge journal of economics* 38, 2 (2014), 257–279.

[53] Philip Howard. 2005. Managing spreadsheets. *White Paper. Bloor Research, Suite* 4 (2005).

[54] Michael Izza. 2014. Twenty principles for good spreadsheet practice. https://www.icaew.com/technical/technology/excel/twenty-principles

[55] J. Dearing. 2019. Study Finds Spreadsheet Risk is Real; Businesses are Aware of the Risk—Yet Despite Relying on this Data to Make Key Business Decisions—the Risk is Ignored. https://www.businesswire.com/news/home/20190612005248/en/Study-Finds-Spreadsheet-Risk-is-Real-Businesses-are-Aware-of-the-Risk%E2%80%94Yet-Despite-Relying-on-this-Data-to-Make-Key-Business-Decisions%E2%80%94the-Risk-is-Ignored

[56] Alon Jacovi, Ana Marasović, Tim Miller, and Yoav Goldberg. 2021. Formalizing trust in artificial intelligence: Prerequisites, causes and goals of human trust in ai. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. 624–635.

[57] Nima Joharizadeh, Advait Sarkar, Andrew D. Gordon, and Jack Williams. 2020. Gridlets: Reusing Spreadsheet Grids. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems (CHI EA '20)*. Association for Computing Machinery, New York, NY, USA, 1–7. https://doi.org/10.1145/3334480.3382806

[58] Simon Peyton Jones, Alan Blackwell, and Margaret Burnett. 2003. A user-centred approach to functions in Excel. In *Proceedings of the eighth ACM SIGPLAN international conference on Functional programming (ICFP '03)*. Association for Computing Machinery, New York, NY, USA, 165–176. https://doi.org/10.1145/944705.944721

[59] Annabel Bhamani Kajornboon. 2005. Using interviews as research instruments. *E-journal for Research Teachers* 2, 1 (2005), 1–9.

[60] Sean Kandel, Andreas Paepcke, Joseph M. Hellerstein, and Jeffrey Heer. 2012. Enterprise Data Analysis and Visualization: An Interview Study. *IEEE Transactions on Visualization and Computer Graphics* 18, 12 (Dec. 2012), 2917–2926. https://doi.org/10.1109/TVCG.2012.219 Conference Name: IEEE Transactions on Visualization and Computer Graphics.

[61] Eser Kandogan, Aruna Balakrishnan, Eben M. Haber, and Jeffrey S. Pierce. 2014. From data to insight: work practices of analysts in the enterprise. *IEEE computer graphics and applications* 34, 5 (Oct. 2014), 42–50. https://doi.org/10.1109/MCG.2014.62

[62] Alan Kay. 1984. Computer Software. *Scientific American* 251, 3 (1984), 52–59. http://www.jstor.org/stable/24920344

[63] Mary Beth Kery and Brad A Myers. 2017. Exploring exploratory programming. In *2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, 25–29.

[64] Benjamin Koskei and Catherine Simiyu. 2015. Role of interviews, observation, pitfalls and ethical issues in qualitative research methods. *Journal of Educational Policy and Entrepreneurial Research* 2, 3 (2015), 108–117.

[65] Sam Lau, Sruti Srinivasa Srinivasa Ragavan, Ken Milne, Titus Barik, and Advait Sarkar. 2021. TweakIt: Supporting End-User Programmers Who Transmogrify Code. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–12.

[66] Jiali Liu, Nadia Boukhelifa, and James Eagan. 2019. Making sense of data workers' sense making practices. 4. https://hal.inrae.fr/hal-02734014

[67] Juan Liu, Aaron Wilson, and David Gunning. 2014. Workflow-based Human-in-the-Loop Data Analytics. In *Proceedings of the 2014 Workshop on Human Centered Big Data Research (HCBDR '14)*. Association for Computing Machinery, New York, NY, USA, 49–52. https://doi.org/10.1145/2609876.2609888

[68] Wendy E. Mackay. 1998. Triangulation within and across HCI disciplines. *Human-Computer Interaction* 13, 3 (1998), 310–315.

[69] Melissa Mazmanian, Marisa Cohn, and Paul Dourish. 2014. Dynamic reconfiguration in planetary exploration: a sociomaterial ethnography. *MIS Quarterly* 38, 3 (Sept. 2014), 831–848. https://doi.org/10.25300/MISQ/2014/38.3.09

[70] Matt McCutchen, Shachar Itzhaky, and Daniel Jackson. 2016. Object spreadsheets: A new computational model for end-user development of data-centric web applications. In *Proceedings of the 2016 ACM International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software*. 112–127.

[71] Mary L. McHugh. 2012. Interrater reliability: the kappa statistic. *Biochemia medica: Biochemia medica* 22, 3 (2012), 276–282. Publisher: Medicinska naklada.

[72] Jorge Mendes, Jácome Cunha, Francisco Duarte, Gregor Engels, João Saraiva, and Stefan Sauer. 2017. Systematic spreadsheet construction processes. In *2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. 123–127. https://doi.org/10.1109/VLHCC.2017.8103459 ISSN: 1943-6106.

[73] Microsoft. 2021. Guidelines for organizing and formatting data on a worksheet. https://support.microsoft.com/en-us/office/guidelines-for-organizing-and-formatting-data-on-a-worksheet-90895cad-6c85-4e02-90d3-8798660166e3

[74] Darren Miller, Gary Miller, and Luis M. Parrondo. 2010. Sumwise: A smarter spreadsheet. In *EuSpRiG*.

[75] Gary Miller, Felienne Hermans, and Robin Braun. 2016. Gradual structuring: Evolving the spreadsheet paradigm for expressiveness and learnability. In *2016 15th International Conference on Information Technology Based Higher Education and Training (ITHET)*. 1–8. https://doi.org/10.1109/ITHET.2016.7760759

[76] Michael Muller, Ingrid Lange, Dakuo Wang, David Piorkowski, Jason Tsay, Q. Vera Liao, Casey Dugan, and Thomas Erickson. 2019. How Data Science Workers Work with Data: Discovery, Capture, Curation, Design, Creation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–15. https://doi.org/10.1145/3290605.3300356

[77] Bonnie A. Nardi and James R. Miller. 1991. Twinkling lights and nested loops: distributed problem solving and spreadsheet development. *International Journal of Man-Machine Studies* 34, 2 (Feb. 1991), 161–184. https://doi.org/10.1016/0020-7373(91)90040-E

[78] Fabian Nunez. 2000. *An extended spreadsheet paradigm for data visualisation systems, and its implementation*. Master's thesis. University of Cape Town.

[79] Ray Panko. 2015. What we don't know about spreadsheet errors today: The facts, why we don't believe them, and what we need to do. *he European Spreadsheet Risks Interest Group 16th Annual Conference (EuSpRiG 2015)* (2015).

[80] Raymond Panko and Richard Halverson. 1996. Spreadsheets on trial: a survey of research on spreadsheet risks. 326–335 vol.2. https://doi.org/10.1109/HICSS.1996.495416

[81] Raymond R. Panko. 1998. What We Know About Spreadsheet Errors. *Journal of Organizational and End User Computing (JOEUC)* 10, 2 (April 1998), 15–21. https://doi.org/10.4018/joeuc.1998040102 Publisher: IGI Global.

[82] Peter Pirolli and Stuart Card. 2005. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proceedings of international conference on intelligence analysis*, Vol. 5. McLean, VA, USA, 2–4.

[83] Stephen G Powell, Kenneth R Baker, and Barry Lawson. 2008. A critical review of the literature on spreadsheet errors. *Decision Support Systems* 46, 1 (2008), 128–138.

[84] John F. Raffensperger. 2001. New guidelines for spreadsheets. *European Spreadsheet Risks Interest Group (EuSpRIG)* (2001).

[85] Cliff Ragsdale. 2014. *Spreadsheet Modeling and Decision Analysis: A Practical Introduction to Business Analytics*. Cengage Learning. Google-Books-ID: cUHAAgAAQBAJ.

[86] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 1135–1144.

[87] Daniel M. Russell. 2016. Simple is Good: Observations of Visualization Use Amongst the Big Data Digerati. In *Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI '16)*. Association for Computing Machinery, New York, NY, USA, 7–12. https://doi.org/10.1145/2909132.2933287

[88] Antti Salovaara, Antti Oulasvirta, and Giulio Jacucci. 2017. Evaluation of Prototypes and the Problem of Possible Futures. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 2064–2077. https://doi.org/10.1145/3025453.3025658

[89] Advait Sarkar. 2015. The impact of syntax colouring on program comprehension. In *Proceedings of the 26th Annual Conference of the Psychology of Programming Interest Group (PPIG 2015)*. 49–58.

[90] Advait Sarkar. 2016. *Interactive analytical modelling*. Ph.D. Dissertation. University of Cambridge, Cambridge, United Kingdom.

[91] Advait Sarkar, Judith W Borghouts, Anusha Iyer, Sneha Khullar, Christian Canton, Felienne Hermans, Andrew D Gordon, and Jack Williams. 2020. Spreadsheet use and programming experience: An exploratory survey. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–9.

[92] Advait Sarkar and Andrew D. Gordon. 2018. How do people learn to use spreadsheets? (Work in progress). In *Proceedings of the 29th Annual Conference of the Psychology of Programming Interest Group (PPIG 2018)*. 28–35.

[93] Advait Sarkar, Andrew D. Gordon, Simon Peyton Jones, and Neil Toronto. 2018. Calculation View: multiple-representation editing in spreadsheets. In *2018 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. 85–93. https://doi.org/10.1109/VLHCC.2018.8506584 ISSN: 1943-6106.

[94] Christopher Scaffidi, Mary Shaw, and Brad Myers. 2005. Estimating the numbers of end users and end user programmers. In *2005 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'05)*. IEEE, 207–214.

[95] R. Schultheis and M. Sumner. 1994. The relationship of application risks to application controls: A study of microcomputer-based spreadsheet applications. *Journal of End User Computing* 6 (Jan. 1994), 11–18. https://doi.org/10.4018/joeuc.1994040102

[96] Clive Seale. 1999. Quality in qualitative research. *Qualitative inquiry* 5, 4 (1999), 465–478.

[97] Peter Sestoft and Jens Zeilund Sørensen. 2013. Sheet-Defined Functions: Implementation and Initial Evaluation. In *End-User Development (Lecture Notes in Computer Science)*, Yvonne Dittrich, Margaret Burnett, Anders Mørch, and David Redmiles (Eds.). Springer, Berlin, Heidelberg, 88–103. https://doi.org/10.1007/978-3-642-38706-7_8

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

[99] Rishabh Singh, Benjamin Livshits, and Benjamin Zorn. 2017. Melford: Using neural networks to find spreadsheet errors. In *Tech. Rep.* Citeseer.

[100] Justin Smith, Justin A. Middleton, and Nicholas A. Kraft. 2017. Spreadsheet practices and challenges in a large multinational conglomerate. In *2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. 155–163. https://doi.org/10.1109/VLHCC.2017.8103463 ISSN: 1943-6106.

[101] Sruti Srinivasa Ragavan, Advait Sarkar, and Andrew D Gordon. 2021. Spreadsheet Comprehension: Guesswork, Giving Up and Going Back to the Author. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21)*. Association for Computing Machinery, New York, NY, USA, 1–21. https://doi.org/10.1145/3411764.3445634

[102] Ricardo Teixeira and Vasco Amaral. 2016. On the emergence of patterns for spreadsheets data arrangements. In *Federation of International Conferences on Software Technologies: Applications and Foundations*. Springer, 333–345.

[103] The FAST Standard Organisation. 2019. The FAST Standard | Full version of the FAST Standard modelling guide. https://www.fast-standard.org/the-fast-standard/

[104] Tuck School of Business at Dartmouth College. 2006. Results of SERP Online Survey of Spreadsheet Usage. http://faculty.tuck.dartmouth.edu/images/uploads/faculty/serp/serp_results.pdf

[105] Mojtaba Vaismoradi, Hannele Turunen, and Terese Bondas. 2013. Content analysis and thematic analysis: Implications for conducting a qualitative descriptive study. *Nursing & Health Sciences* 15, 3 (2013), 398–405. https://doi.org/10.1111/nhs.12048 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/nhs.12048.

[106] Janet Vertesi. 2014. Seamful Spaces: Heterogeneous Infrastructures in Interaction. *Science, Technology, & Human Values* 39, 2 (March 2014), 264–284. https://doi.org/10.1177/0162243913516012

[107] David Wakeling. 2007. Spreadsheet functional programming. *Journal of Functional Programming* 17, 1 (Jan. 2007), 131–143. https://doi.org/10.1017/S0956796806006186

[108] Jayne Wallace, Peter C. Wright, John McCarthy, David Philip Green, James Thomas, and Patrick Olivier. 2013. *A Design-Led Inquiry into Personhood in Dementia*. Association for Computing Machinery, New York, NY, USA, 2617–2626. https://doi.org/10.1145/2470654.2481363

[109] Jack Williams, Nima Joharizadeh, Andrew D. Gordon, and Advait Sarkar. 2020. Higher-Order Spreadsheets with Spilled Arrays.. In *ESOP*. 743–769.

[110] Jack Williams, Carina Negreanu, Andrew D Gordon, and Advait Sarkar. 2020. Understanding and Inferring Units in Spreadsheets. In *2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, 1–9.

[111] Aaron Wilson, Margaret Burnett, Laura Beckwith, Orion Granatir, Ledah Casburn, Curtis Cook, Mike Durham, and Gregg Rothermel. 2003. Harnessing curiosity to increase correctness in end-user programming. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '03)*. Association for Computing Machinery, New York, NY, USA, 305–312. https://doi.org/10.1145/642611.642665

[112] Aaron Wilson, Margaret Burnett, Laura Beckwith, Orion Granatir, Ledah Casburn, Curtis Cook, Mike Durham, and Gregg Rothermel. 2003. Harnessing curiosity to increase correctness in end-user programming. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 305–312.

[113] Amy X. Zhang, Michael Muller, and Dakuo Wang. 2020. How do Data Science Workers Collaborate? Roles, Workflows, and Tools. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW1 (May 2020), 022:1–022:23. https://doi.org/10.1145/3392826

[114] Jie Zhang, Shi Han, Dan Hao, Lu Zhang, and Dongmei Zhang. 2018. Automated refactoring of nested-IF formulae in spreadsheets. In *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2018)*. Association for Computing Machinery, New York, NY, USA, 833–838. https://doi.org/10.1145/3236024.3275532

A CODES USED FOR THEMATIC ANALYSIS

A.1 Observed Data Arrangement Layouts

- observed tables: vertical tables
- observed tables: horizontal tables
- observed tables: relationship tables
- observed layouts and structures: structured tabular data
- observed layouts and structures: unstructured tabular data
- observed layouts and structures: unstructured non-tabular data
- columns: hidden helper columns
- columns: empty columns used
- headers: using multi-row headers
- headers: using one header
- headers: descriptive and long headers
- headers: no headers used
- rows: empty rows used
- comments: in-cell commenting
- comments: sticky note commenting
- comments: separate column for comments
- comments: manual comments in the grid
- row annotations: highlighting entire rows
- row annotations: images inside rows
- row annotations: strike-through on rows
- colours: using colours to flag errors
- colours: using colours as information
- colours: using colours to organize data
- optional annotations: using colours to improve understanding
- unnecessary annotations: not useful for understanding text

- how colours are set: using conditional formatting
- how colours are set: manually added

A.2 Data Arrangement Patterns and Practices

- data import: data imported from third party software
- data import: data added manually
- data migration: data migration among spreadsheet software
- data migration: data migration among third parties
- data practices: conversion from raw data to tabular data
- data practices: conducting all analysis on 'master' spreadsheet
- data practices: keeping original data intact
- data practices: keeping raw data intact
- data practices: balancing structure for computation and flexibility
- data practices: making trade-offs to ensure sheet is not labor intensive
- transforming multiple sheets into one: using single sheet to improve comprehension
- transforming multiple sheets into one: using single sheet to improve readability
- transforming multiple sheets into one: using single sheet due to being unable to link sheets
- dividing data into separate sheets: using seperate sheets to conduct additional analysis
- dividing data into separate sheets: using seperate sheets for collaborator analysis
- dividing data into separate sheets: using separate sheets to differentiate content
- dividing data into separate sheets: using seperate sheets for easier sort
- dividing data into separate sheets: using seperate sheets for easier filter
- dividing data into separate sheets: using seperate sheets to store backups
- dividing data into separate sheets: using seperate sheets to hide obsolete data
- dividing data into separate sheets: using seperate sheets to duplicate sheets
- dividing data into separate sheets: using seperate sheets to reuse sheets

A.3 Factors and Motivators affecting Data Arrangement Patterns

- structure of imported data from third parties: structure by third party software has been tidy
- ability to export data into third-party tools: converting to tidy to be able to export to visualization tools
- ability to export data into third-party tools: converting to tidy to be able to export to databases
- ability to export data into third-party tools: converting to tidy to able to export to word tools
- ability to export data into third-party tools: ability to export data into analysis tools
- knowledge over how data would evolve over time: not knowing what additional data added caused less structure over time
- knowledge over how data would evolve over time: knowing how data will be used adds more organization and structure
- lack of skills or knowledge over how to structure: spreadsheet was unstructured because no knowledge over sheet linking
- structure decided by templates provided: voluntarily selecting or using own templates
- structure decided by templates provided: voluntarily selecting or using other's templates
- structure decided by templates provided: creating tidy templates for collaborators
- structure decided by templates provided: creating tidy templates for co-workers
- structure decided by templates provided: unwillingly having to use managers' templates
- structure decided by templates provided: tendency to use managers' templates
- ability to use data arrangement and computation tools: need structure to perform data arrangement and computation
- ability to use data arrangement and computation tools: need structure to perform computation
- ability to use data arrangement and computation tools: no need for structure because not doing computations or data arrangement
- ability to use data arrangement and computation tools: no need for structure because not arranging data
- ability to accomplish the task only matters: having little to no time
- ability to accomplish the task only matters: having no motivation to introduce structure
- ability to accomplish the task only matters: the job can be done with the lack of structure
- ability to accomplish the task only matters: not wanting to spend effort
- ability to accomplish the task only matters: being lazy or idle
- ability to accomplish the task only matters: spreadsheets not structured well because it won't be used often
- volume of data: less structure because data is little
- complexity of data: less structure because data is not complex
- volume of data: more structure because data is large
- complexity of data: more structure because data is complex
- improving understanding and reducing errors for collaborators: adding structure to prevent errors by collaborators
- improving understanding and reducing errors for collaborators: adding structure to prevent errors by colleagues
- improving understanding and reducing errors for collaborators: adjusting structure to make spreadsheets more readable for collaborators
- improving understanding and reducing errors for collaborators: adjusting structure to make spreadsheets more readable for clients
- improving understanding and reducing errors for collaborators: adjusting structure to make spreadsheets more readable for co-workers

- structured influenced/decided by collaborators: multiple collaborators cause untidy structure over time
- structured influenced/decided by collaborators: structure has been completely decided by collaborators
- structured influenced/decided by collaborators: structure has been partially decided by collaborators
- structured influenced/decided by collaborators: structure has been heavily decided by collaborators
- spreadsheet audience: spreadsheets shared publicly were more structured
- spreadsheet audience: spreadsheets shared publicly were cleaner
- spreadsheet audience: spreadsheets shared with clients had more structure
- spreadsheet audience: spreadsheets shared with employers had more structure
- spreadsheet audience: spreadsheets shared with co-workers had more structure
- comprehension cost: spreadsheet is less structured, because it was private
- reputation cost: spreadsheet is more structured, because it was shared with managers
- reputation cost: spreadsheet is more structured to preserve corporate reputation
- reputation cost: spreadsheet is more structured because it was facing audit
- need for comprehension: using rich data to improve visualization
- need for understanding: using rich tables to improve visualization
- need for comprehension: using rich data to be able to understand the spreadsheet later
- need for understanding: using rich data to be able understand the spreadsheet later
- structuring data the way the mind thinks about it: structuring data as tabular the way the mind thinks about it
- structuring data the way the mind thinks about it: layout out unstructured data the way the mind thinks about it
- structuring data the way the mind thinks about it: aligning data the way the mind thinks about it
- need for flexibility and full control: unstructured because need for control and flexibility
- need for total control over data: structured because user likes to have limited structure
- structuring data to be error-prone: adding tabular structure to prevent errors
- structuring data to be error-prone: adding tabular structure to reduce errors
- structuring data to be error-prone: adding tabular data to ensure auditing tools can be run
- structuring data to be error-prone: using data validation to prevent errors
- structuring data to be error-prone: using data validation to reduce errors
- structuring data to be error-prone: preventing spreadsheet errors when importing dates
- structuring data to be error-prone: preventing spreadsheet errors when importing names
- need for ease of use when structuring data: adding unstructured data is easy
- need for ease of use when structuring data: adding unstructured data requires no learning
- need for ease of use when structuring data: adding unstructured data is quick
- need for ease of use when structuring data: adding unstructured data on spreadsheets is efficient because they are ubiquitous
- need for ease of use when structuring data: adding unstructured data allows easy reuse

A.4 Limitations and Drawbacks of Unstructured Data

- need for ease of use when structuring data: inability to run necessary computations and analysis
- need for ease of use when structuring data: inability to use data arrangement and functionalities
- need for ease of use when structuring data: inability to conduct meaningful analysis
- lacking knowledge to use formulas and data arrangement tools: lacking awareness of what features and formulas exist
- lacking knowledge to use formulas and data arrangement tools: lacking willingness and effort to learn new features and formulas
- lacking knowledge to use formulas and data arrangement tools: lacking time to discover new features and formulas
- lacking knowledge to use formulas and data arrangement tools: lacking time to discover new formulas
- lacking knowledge to use formulas and data arrangement tools: difficulty discovering new features
- lacking knowledge to use formulas and data arrangement tools: difficulty discovering new formulas

A.5 Workarounds for dealing with Unstructured Data

- getting help from colleagues and the internet: memorizing formulas through online searchers
- getting help from colleagues and the internet: asking for help from online forums
- getting help from colleagues and the internet: difficulty using already known advanced formulas
- getting help from colleagues and the internet: getting necessary help from colleagues
- getting help from colleagues and the internet: getting necessary help from collaborators
- getting help from colleagues and the internet: getting necessary help from family members
- using trial and error to learn to use formulas: learning to use formulas through trial and error
- manually computing & formatting: manually computing formulas and pivot tables
- manually computing & formatting: manually sorting out rows

- manually computing & formatting: manually filtering out rows
- manually computing & formatting: manually adding colour formatting
- manually computing & formatting: manually adding data validation
- using third parties to perform analysis: third parties are easier to use than excel formulas
- using third parties to perform analysis: third parties provide better analysis than spreadsheets
- using third parties to perform analysis: third parties create tremendous manual labor and work
- manually checking for errors: manually reviewing collaborators edits for errors
- manually checking for errors: manually reviewing for own errors in spreadsheets

A.6 Downsides of Formal or Structured Tables

- formal tables have very poor value: the advantage and value of formal tables is not clear
- formal tables require learning: formal tables are an issue for novice collaborators
- formal tables require learning: formal tables require learning
- formal tables limit structuring data: inability to merge rows
- formal tables limit structuring data: inability to have empty rows
- formal tables limit structuring data: inability to have empty columns
- formal tables limit structuring data: combining columns is challenging
- formal tables limit the flexibility: inability to have full and clear control of the data
- formal tables limit the flexibility: perception of losing control over data
- formal tables create performance problems: slowing down application
- formal tables create performance problems: slowing down device
- formal tables have poor functionality: formal tables don't recognize extra columns
- formal tables remove essential information: single headers are problematic
- formal tables are incompatible with rich data: difficulty merging both
- formal tables cannot be easily exported: difficulty exporting table into other formats

A.7 Improving the Experience of Structuring Tabular Data in Spreadsheets

- headers: ability to easily set nested headers
- columns: ability to easily apply data validation from columns
- columns: ability to easily apply conditional formatting for columns
- columns: ability to group by category in columns
- columns: ability to expand and collapse grouped categories
- columns: ability to sort and filter grouped categories
- columns: ability to conduct computations on grouped categories
- columns: ability to easily aggregate columns
- columns: ability to set formulas on entire columns
- columns: ability to lock formulas on entire columns
- rows: ability to easily apply conditional formatting for rows
- nudging for transformation: ability to preview transformations before they are made
- nudging for transformation: ability to experiment with transformations before they are live
- nudging for transformation: asking users for permission before transformation
- nudging for transformation: giving users information before transformation
- nudging for transformation: ability to permanently hide nudging notifications
- transformations: ability to exclude columns from being reshaped
- transformations: ability to keep the source data intact
- transformations: ability to conduct transformations on a separate sheet
- transformations: transformations should never be automatic or forced
- templates: propose excel tabular templates for users
- templates: propose limited number of templates
- sort and filter: allow users to easily undo sort and filter
- sort and filter: allow users to easily hide sort and filter buttons
- sort and filter: allow users to easily disable sort buttons
- sort and filter: allow users to easily disable filter buttons
- sort and filter: allow users to easily sort and filter empty rows
- sort and filter: allow users to easily shuffle rows
- sort and filter: allow users to easily sort and filter more than two conditions
- sort and filter: allow users to filter rows based on one column value
- styles and formatting: allow users to preserve their own styling
- styles and formatting: allow users to preserve their own colouring
- styles and formatting: allow users easily change styling
- styles and formatting: allow users easily change colouring
- styles and formatting: allow users to automatically apply styling
- styles and formatting: allow users to automatically apply colouring
- styles and formatting: allow users to change the style of sort buttons
- styles and formatting: allow users to change the style of filter buttons
- navigation: allow users to easily select table values
- reducing errors: automatically flag obvious errors to users
- reducing errors: allow users to review errors and take manual actions
- commenting and annotations: allow users to add 'tool-tip' comments for whole columns

- reducing errors: don't automatically convert annotations into columns
- analytics and information: give summary statistics to users
- analytics and information: give deeper insights and analytics to users
- analytics and information: allow users to set reminders
- analytics and information: allowing users to query the tables
- automation and intelligence: automatically provide data validation to users
- automation and intelligence: automatically suggesting autofilling for users
- automation and intelligence: automatically suggesting row selection for users
- machine learning pain points: machine learning insights should only be a suggestions
- machine learning pain points: machine learning cannot know what users exactly ones
- machine learning pain points: users don't trust machine learning to be helpful
- machine learning pain points: users need full transparency and control
- machine learning pain points: machine learning creates privacy issues

A.8 Improving the Experience of Spreadsheets

- improve the experience of cell editing: need for more flexibility in formatting cells
- improve the experience of cell editing: need to add hyperlinks inside cells
- improve the experience of cell editing: need for better merging experiences
- improve the experience of cell editing: need for make better dependencies between contents of different cells
- improve help experiences: need for better help features
- improve help experiences: need for better excel templates
- improve the experience of language tools: need for foreign language support
- improve the experience of language tools: need for clearer spelling check
- improve the experience of language tools: need for academic referencing support
- improve the experience of programming: need to be able to run programming tools within excel
- improve the experience of commenting: improving commenting experiences
- improve the experience of commenting: being able to build charts
- improve the experience of collaboration: being able to better track and review collaborator edits
- improve the experience of collaboration: addressing access control issues by external collaborators
- improve the experience of version history: need better own tracking history
- improve the experience of connectivity: being able to work everywhere
- improve the experience of connectivity: need to stay within the same ecosystem of office products
- improve the experience of intelligence: need for better autopopulation for rows
- improve the experience of intelligence: improve the experience of formula auto-completion
- improve the experience of linking spreadsheets: address the privacy challenges of linking spreadsheets
- improve the experience of linking spreadsheets: improving the experience of combining data from different spreadsheets
- improve the experience of sheet protection: improving the experience of protecting sheets
- improve the experience of exporting: improving the experience of exporting .CSV files
- improve the experience of exporting: improving the experience of exporting .TSV files