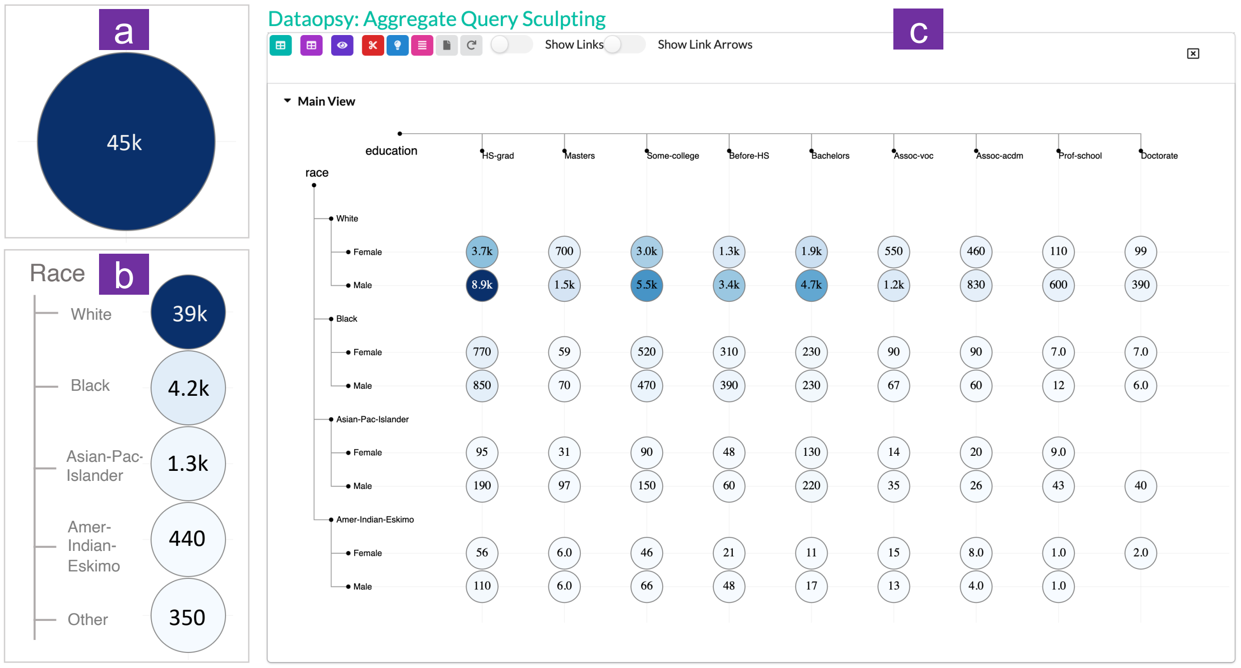
Dataopsy: Scalable and Fluid Visual Exploration using Aggregate Query Sculpting

Md Naimul Hoque and Niklas Elmqvist



race	gender	HS-grad	Masters	Some-college	Before-HS	Bachelors	Assoc-voc	Assoc-acdm	Prof-school	Doctorate
White	Female	3.7k	700	3.0k	1.3k	1.9k	550	460	110	99
	Male	8.9k	1.5k	5.5k	3.4k	4.7k	1.2k	830	600	390
Black	Female	770	59	520	310	230	90	90	7.0	7.0
	Male	850	70	470	390	230	67	60	12	6.0
Asian-Pac-Islander	Female	95	31	90	48	130	14	20	9.0	
	Male	190	97	150	60	220	35	26	43	40
Amer-Indian-Eskimo	Female	56	6.0	46	21	11	15	8.0	1.0	2.0
	Male	110	6.0	66	48	17	13	4.0	1.0	

Fig. 1: **The Adult Income [17] dataset visualized in Dataopsy.** (a) The initial view in Dataopsy is a single supernode, an aggregate representation of all 45,000 data points in the dataset. Dataopsy provides six interactions ($\mathbb{P}^6$) to explore the dataset iteratively. (b) For example, a user can *partition* the initial supernode by RACE, transforming it into five new supernodes. (c) The user can *prune* the OTHER category in Race and then add GENDER on the vertical axis and EDUCATION on the horizontal axis for *partitioning*. At the top of each view in Dataopsy, a control panel contains stylized buttons for performing and tracking the interactions.

Abstract—We present *aggregate query sculpting* (AQS), a faceted visual query technique for large-scale multidimensional data. As a "born scalable" query technique, AQS starts visualization with a single visual mark representing an aggregation of the entire dataset. The user can then progressively explore the dataset through a sequence of operations abbreviated as $\mathbb{P}^6$: *pivot* (facet an aggregate based on an attribute), *partition* (lay out a facet in space), *peek* (see inside a subset using an aggregate visual representation), *pile* (merge two or more subsets), *project* (extracting a subset into a new substrate), and *prune* (discard an aggregate not currently of interest). We validate AQS with DATAOPSY, a prototype implementation of AQS that has been designed for fluid interaction on desktop and touch-based mobile devices. We demonstrate AQS and Dataopsy using two case studies and three application examples.

Index Terms—Multidimensional data visualization, multivariate graphs, visual queries, visual exploration.

1 INTRODUCTION

E pluribus unum—Latin for "out of many, one"—is one of the official mottos of the United States, and also happens to be a dominant strategy for managing scale in data visualization: through *aggregation* of many data items into one visual mark [15]. Visualizing today's real-world datasets, from the Facebook social network to the billion-parameter large language models (LLMs) of Jurassic-1 and GPT-3, is more or less impractical when using a representation that insists on using one mark per data item—so-called *unit visualizations* [38]. This fact is exacerbated by today's trend towards mobility in data analysis [28] using mobile devices that have pathologically small mobile screens [21]. And finally, even if we had enough pixels, there is a limit to the number of data items that the human perceptual system can interpret effectively [15, 33]. What is truly needed to tackle the next generation of data visualization challenges are techniques that are "born scalable"; i.e., that have been designed to be scalable from inception.

In this paper, we propose such a "born scalable" visualization technique that we call *aggregate query sculpting* (AQS). AQS is primarily designed for multidimensional tabular data, but can also be used for multivariate networks (i.e., links connecting data items or nodes). Unlike typical overviews containing thousands or tens of thousands of data items, the initial AQS view is a substrate containing a single *su-*

• *Md Naimul Hoque is with University of Maryland, College Park in College Park, MD, USA. E-mail: nhoque@umd.edu.*
• *Niklas Elmqvist is with Aarhus University in Aarhus, Denmark; the work was conducted at University of Maryland, College Park, MD, United States. E-mail: elm@cs.au.dk.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

pernode (aggregate node) representing all of the data items or nodes in the dataset (Figure 1a). From this point on, the goal of AQS is to provide the user with a set of fluid interactions to split the supernode into smaller data subsets during exploration, similar to how a sculptor will iteratively cut a large piece of clay into pieces to shape and mold separately. There are six such interactions, and they are summarized by the acronym $\mathbb{P}^6$ for *pivot*, *partition*, *peek*, *pile*, *project*, and *prune*. Pivoting splits a supernode into facet supernodes based on a data attribute; partitioning lays out facet nodes in visual space; peeking shows the data distribution inside a node; piling merges two or more nodes; projection extracts a subset into a new visual substrate where $\mathbb{P}^6$ operations can continue; and pruning eliminates undesired supernodes.

To validate the utility of aggregate query sculpting, we present DATAOPSY, a web-based prototype implementation of AQS capable of visualizing thousands of entities in a standard web browser. Dataopsy and its AQS concepts draw on many existing concepts. Pivoting is based on graph pivoting techniques from PivotGraph [48] as well as Tableau (née Polaris [46]). Microsoft's Sand Dance [12] and Google's Facets [20, 49] use a similar attribute-based 2D layout, but Dataopsy is designed for networks and maintains aggregate supernodes instead of a unit representation. Furthermore, the projection technique, which is inspired by Shneiderman and Aris's semantic substrates [3, 44], enables creating multiple linked substrates in the same visual space to avoid inelegant deep facet nesting, which is a problem for PivotGraph, Polaris, Sand Dance, and Facet browsers. Finally, the Dataopsy interface is designed for fluid interaction on a desktop and tablet. The result is a smooth and scalable data exploration application for multivariate data synthesizing features no comparable tool possesses.

We present two case studies and three application examples involving Dataopsy to demonstrate the utility of aggregate query sculpting. In the first case study, two data scientists and algorithmic fairness researchers used Dataopsy to evaluate machine bias in the Adult Income dataset [6, 17]. In the second case study, a creative writer used Dataopsy to navigate a complex set of scenes, locations, chapters, characters, and events from a fiction novel and sculpted it for adaptation to a screenplay format. As an application example, we showed how an analyst could use Dataopsy to understand the linguistic properties of inter-community conflicts from 300,000 Reddit posts [27]. We then analyzed 1.7 billion taxi rides in New York City to identify hotspots for rides [34]. Finally, we used the VisPub [24] dataset to analyze the IEEE VIS scientific community over the years. Overall, the case studies and examples show the generalizability and scalability of AQS and Dataopsy in exploring diverse multidimensional datasets.

2 BACKGROUND

Building and managing queries is as old as visualization itself; the "zoom and filter" part of Shneiderman's visual information seeking mantra [43] refers to controlling which data items to show on the screen, primarily to manage scale. In this section, we review the related work on query management and visual information seeking, including for multivariate datasets, faceted browsing, and multivariate graphs.

2.1 Multivariate or Multidimensional Visual Exploration

Multidimensional datasets consist of many attributes per observation, and are routinely found in both tabular as well network applications. For example, the U.S. Census dataset of citizen demographics includes hundreds of attributes capturing individuals living in the United States, including properties such as age, gender, education, annual income, and marital status. Searching and filtering such multivariate datasets was a challenging prospect often involving writing SQL queries until Williamson and Shneiderman proposed *dynamic queries* using double-ended range sliders [50]. These sliders enabled selecting an interval in both quantitative and—later—categorical axes [2] in the dataset.

Significant work has since been conducted on searching and querying multidimensional datasets. Many visual query techniques are intimately tied to a visual representation. For example, axis filtering [41] is designed for filtering on the axes in a parallel coordinate plot. ExPlates [25] *spatializes* multidimensional interaction into 2D space. As the name implies, the ScatterDice system [14] is based on scatterplot matrices and introduces the concept of *query sculpting* where a dataset is filtered from different angles until the final desired result is reached. The idea was later generalized in the VisDock [7] cross-cutting interaction library and became a fundamental feature of the Keshif visual data browser [51, 52]. In this paper, we build on query sculpting but generalize it to visual aggregates, where massive datasets have been hierarchically grouped into aggregation trees for scalability [15].

Polaris [46] and FromDaDy [23] were early examples of highly interactive multivariate query and visualization systems. Most multivariate visualizations are *unit visualizations* [38] in that they represent each data item with exactly one visual mark. More recently, Microsoft's SandDance [12] and Google Facets [20, 49] enable using similar interaction, layout, and query techniques to visualize multivariate datasets, such as machine learning training data. ATOM [38] was designed as a declarative grammar for building unit visualizations. We differ from prior works in several dimensions. *First,* instead of unit marks, we use aggregated marks to scale unit visualization to large datasets. *Second,* AQS introduces six interactions for iterative explorations of the aggregated marks. While some of these interactions are motivated by prior works, the combination of them provides analytical capabilities that no prior works possess. For example, Polaris's Cross and Nest operations motivated our pivot and partitioning operations. However, Cross and Nest could create inelegant nesting and visual clutter. We solve this limitation by integrating the Projection operation, motivated from semantic subtrates [3, 44]. *Finally,* prior works primarily focus on desktop applications whereas AQS is suitable for data analysis in touch-based mobile devices and extends to network analysis.

2.2 Faceted Browsing

Faceted browsing [53], where a corpus is explored along one or more conceptual dimensions, was introduced as an alternative to keyword search and image similarity for browsing large-scale image repositories. The idea was quickly generalized to any multidimensional dataset and then adopted by many internet search providers, particularly for e-commerce and real estate websites.

Of course, faceted browsing is a powerful idea with applications to many information retrieval and query research problems. One of the early applications was FacetMap [45], a highly visual and dynamic visualization that summarizes the current state of the filters and search results based on a space-filling rectilinear layout approach. FacetMap shares many similarities with Dataopsy and AQS, but our approach uses a single set of vertical and horizontal axis mappings for displaying dimensions. For this reason, Dataopsy yields visual representations that are more stable and easier to understand. Nevertheless, we draw inspiration from FacetMap's aggregated and scalable visual encoding.

Several other research tools are based on faceted browsing. FacetLens [29] build on FacetMap and uses a similar representation, but support visual comparison view as well as more advanced pivoting operations. FacetZoom [9] enable smooth exploration of hierarchical metadata using a continuous zooming interaction. Finally, PivotPaths [11] provides a fluid and highly interactive browsing experience that externalizes the links (or paths) between different facet values. These existing tools all served as inspiration for our work in this paper.

2.3 Multivariate Graphs

From a data visualization perspective, the leap from table to network is small: all you need are relations connecting entities [33]. Practically, this means adding a second "edge table" linking keys in the original node table. While layout techniques are mostly radically different for tables vs. networks, there is one approach that is shared: *attribute-based layout* [35, 48], where the position of a node on a geometric axis is dependent on a specific attribute associated with the node. Not surprisingly, this kind of visual mapping is commonly applied to data points when mapping a data table to visual space, such as in a scatterplot.

A canonical example of attribute-based layout is PivotGraphs [48], which uses vertical and horizontal space to unpack a single aggregated node in a node-link into 2D space. Aggregated edges show relations in the resulting graph. Our work in this paper draws heavily on PivotGraphs, but generalizes the idea to both tabular as well as network data, and also introduces several new operations to improve on the idea.

GraphDice [4] is another example of attribute-based layout: it is essentially a network version of the original ScatterDice [14] system discussed above. However, unlike PivotGraphs, GraphDice is a unit visualization system with one visual mark per data item. While our Dataopsy system is based on visual aggregates, we borrow the faceted navigation supported by the GraphDice tool for our work.

Scale is a perpetual problem for graph visualization. ASK-GraphView [1] supports interactive visual exploration of node-link diagrams consisting of millions of nodes through the use of clustering and animation. ZAME [13] instead uses adjacency matrices and a level-of-detail pyramid to support massive scale. DOI graphs [47] handle the problem by showing only subsets of graphs. Finally, Refinery [26] uses a similar form of "associative browsing" to support browsing on limited neighborhoods of a massive heterogeneous graph.

Heterogeneous—or multimodal—graphs are a specialized subset of multivariate graphs because one attribute governs the *type* of the node; e.g., students and courses in a registrar's database, books, magazines, and digital media in a library database, or gokarts, trainers, and drivers in a racing club roster. Aris and Shneiderman studied how to best visualize such multimodal data by separating them into individual *semantic substrates* [3, 44], one for each node type. Ghani et al. [19] studied the use of visualization techniques for heterogeneous data for social network analysis, presenting an approach akin to parallel coordinate displays for network data in response. Finally, Ploceus [31] generalizes the idea of linked tables yielding heterogeneous networks, presenting an algebra and an interactive visualization system to support it. All of these existing tools and techniques were influential in our design of AQS. However, none of them provide the same kind of fluid, highly interactive, and scalable approach to visual exploration and querying that AQS and Dataopsy do.

3 Aggregate Query Sculpting

Aggregate query sculpting (AQS) is an iterative filtering technique for multivariate data. It draws inspiration from the ScatterDice technique [14] where the *query sculpting* concept was introduced based on the metaphor of a sculptor repeatedly chiseling away at a block of stone until the sculpture is complete. However, while the original implementation of the techniques was intended for unit visualizations [38] such as scatterplots, where each individual data point is represented by a unique visual mark, aggregate query sculpting, as the name suggests, is designed for aggregated visual representations where individual marks can represent many—potentially thousands or even millions—of data items. We use the term "born scalable" to refer to visual representations and interaction techniques that were designed for massive scale.

Here we describe the data model for aggregate query sculpting and the fundamental $\mathbb{P}^6$ operations in abstract terms. In the following section, we discuss how to implement the $\mathbb{P}^6$ operations in the web-based Dataopsy prototype tool for multivariate data.

3.1 Design Rationale

We designed aggregate query sculpting by harnessing prior art from the literature with three specific design goals (DG1–DG3) in mind:

DG1 **Born scalable:** Realistic datasets cannot be visualized as unit visualizations [38] because of both technical and perceptual limitations. Instead, robust visual representations must be designed from the ground up using visual and data aggregation [15].

DG2 **Fluid interaction:** We envision an iterative and progressive filtering method based on fluid interaction [16], where user actions promote flow [8], are based on direct manipulation [42], and minimize the gulfs of execution and evaluation [36]

DG3 **Faceted browsing:** Multidimensional datasets are easily navigated, filtered, and queried using *faceted search* [53] where filters can be expressed across multiple hierarchical dimensions (*facets*).

3.2 Data Model

All AQS operations are applied to a multidimensional dataset $\mathbb{D}' \subseteq \mathbb{D}$, where $\mathbb{D}$ is the full dataset currently being visualized. The $\mathbb{D}'$ is called a *supernode* even if the data is not relational. Supernodes that are part of networks also have *superlinks* $\mathbb{E}'$; edge subsets drawn from the full edge set $\mathbb{E}$. Some AQS operations operate purely on a data level, whereas others operate on a visual level, and others still operate on both. Furthermore, the operations tend to apply either to entire rows or entire columns—or the entire dataset—in a substrate based on the visual layout. We discuss these details for each operation below.

3.3 Visual Representation

A multidimensional visual representation managed using aggregate query sculpting includes one or more *semantic substrates* [3, 44]: a 2D visual space of any geometric dimension. Additional substrates can be laid out depending on the nature of the data and the user's wishes; for example, two substrates can share a horizontal axis, making it useful to stack the substrates vertically (one above the other).

Each substrate contains one or more supernodes $\mathbb{D}'$ (which could potentially be the entire dataset $\mathbb{D}$ if only one substrate is in use). Inside the substrate, visual aggregates representing datasets are organized in a regular 2D grid with row and column headers. Each supernode $\mathbb{D}'$ is represented by a single visual mark (DG1); this is typically a simple 2D geometric shape such as a circle. The size of the underlying data can be conveyed using multiple different methods (sometimes redundantly), such as using a label, a color scale, or the size of shape. When the dataset is a multivariate network, the relationships (edges) between entities (vertices) in the underlying network are also aggregated into superlinks that are represented using single link marks (DG1). Again, the number of aggregated edges can be conveyed using color or thickness.

3.4 Interaction

We envision aggregate query sculpting as a highly interactive and fluid [16] query technique where the user rapidly performs multiple operations to identify the data they are interested in (DG2). Furthermore, operations can be performed on individual rows or columns, or entire groups of rows or columns by using the partitioning hierarchy (Figure 2). To facilitate such rapid, direct, and reversible interaction, we suggest including an interaction stack where users can easily overview, undo, and redo individual operations.

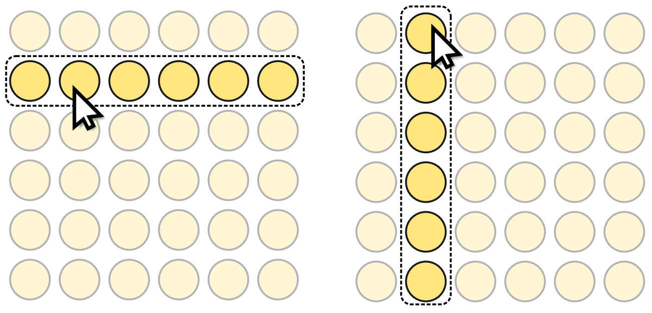

(a) Row-level operations. (b) Column-level operations.

Fig. 2: **Operation scope.** Aggregate query sculpting operations are performed on entire facets, which means that the scope will be entire rows or columns (or dimensions that are currently hidden).

3.5 Query Operations

We define aggregate query sculpting based on six fundamental sculpting operations that we call $\mathbb{P}^6$ for *pivot*, *partition*, *peek*, *pile*, *project*, and *prune*. Each operation is applied on a row or column basis to ensure a consistent grid layout. For the treatment below, assume that $\mathbb{D}$ is a multidimensional dataset consisting of cars with standard dimensions such as gas mileage, acceleration, weight, cylinders, origin, etc.

Pivot. The *pivot* operation splits a supernode $\mathbb{D}'$ into N disjoint supernodes $\{\mathbb{D}'_1, \ldots, \mathbb{D}'_N\}$ based on a group criterion. This is a generalized form of faceted browsing (DG3). Some group criteria use nominal data types; for example, we could imagine pivoting $\mathbb{D}'$ based on the number of cylinders, resulting in three supernodes $\mathbb{D}'_i$ for 4, 6, and 8 cylinders (Figure 3). Alternatively, we could pivot by binning a quantitative value, such as five intervals of gas mileages for $[0,10), [10,20), [20,30), [30,40)$, and $[40,\infty)$ miles per gallons.

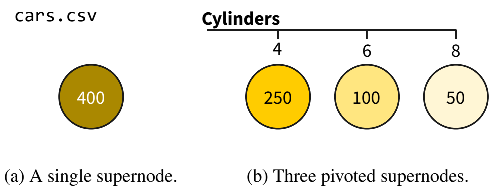


Fig. 3: **Pivoting and partitioning supernodes.** Laying out pivoted supernodes along the horizontal axis.

Partition. The *partition* operation is a pure visual operation for laying out supernodes $\mathbb{D}'_i$ in 2D space along a vertical or horizontal geometric axis. Partitioning will use the entire available space along the chosen geometric axis in the current substrate. This is typically done by allocating an equal amount of visual space to each supernode, although it is also possible to allocate visual space proportional to the size of each supernode (i.e., the number of items in each supernode). The approach is similar to PivotGraphs [48] and Polaris [46], but supports nesting in multiple levels. If the chosen geometric axis has already been used for partitioning, the next level of partitioning will be nested. For example, if we first partition the horizontal axis based on the three sets of cylinders (4, 6, and 8), we can then partition each of these three categories based on the four gas mileage groups; see Figure 4.

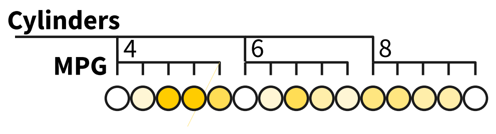


Fig. 4: **Nested axis partitioning.** Example showing how to nest multiple pivoted axes inside a single geometric axis.

Peek. Sometimes the user wants to see inside a supernode without pivoting and partitioning. The *peek* operation transforms the visual representation of one or all aggregate marks into a *glyph representation* [15] showing the contents of each mark based on some axis. For example, peeking can change the color-coded circles into pie charts showing the origin (U.S., Europe, or Asia) of each group of cars pivoted and partitioned based on number of cylinders and then gas mileage.

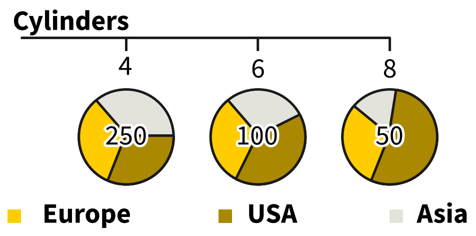


Fig. 5: **Peeking into supernodes.** Representing each visual aggregate as a pie chart showing the origin of each subset of cars.

Pile. *Piling* merges two or more selected supernodes into a single supernode (i.e., as the union of the selected supernodes), potentially enabling the user to name the resulting supernode. This can be useful when an automatic binning operation yields too many individual supernodes, some of which are meaningless on their own. For example, the user could choose to pile the [0, 10) and [10, 20) gas mileage supernode into a single supernode that they name "poor fuel economy" (Figure 6).

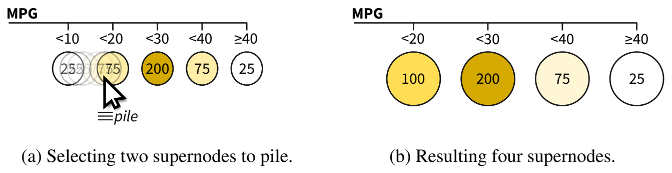

MPG
<10
<20
<30
<40
≥40
25
75
200
75
25
pile
(a) Selecting two supernodes to pile.
MPG
<20
<30
<40
≥40
100
200
75
25
(b) Resulting four supernodes.

Fig. 6: **Piling supernodes.** Grouping the two lowest mileage supernode of cars into a single supernode.

Project. Partitioning multiple pivots into the same geometric axis will eventually yield deep nesting and an explosion of supernode combinations. To reduce clutter, the *project* operation enables selecting a subset of the data and projecting it onto a new semantic substrate that is laid out independent of the originating substrate. The selected data is subtracted from the original substrate, ensuring that the substrates remain disjoint. For example, the user could select all of the low fuel economy cars and project them onto a new substrate to enable further exploration while avoiding to add to the existing nested hierarchy of partitions (Figure 7).

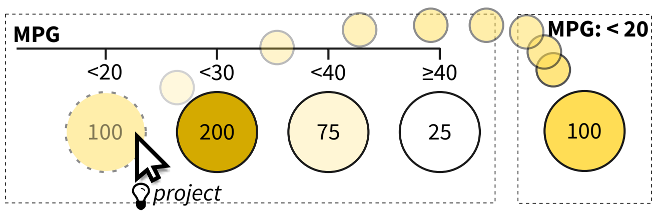


Fig. 7: **Projecting to a new substrate.** Extracting low fuel economy cars to a new semantic substrate for continued exploration.

Prune. Finally, *prune* allows for eliminating (e.g., hiding; all actions are reversible) selected supernodes from view. The operation is similar to the FromDaDy multidimensional visualization tool [23]. It can be applied to entire nested hierarchies, or to specific data values. For example, the user could easily eliminate all U.S. cars from the low fuel economy substrate by pruning on that data value in the origin dimension (Figure 8).

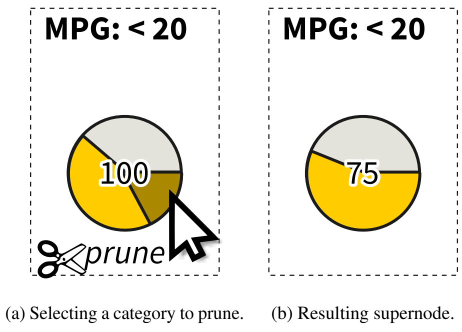


Fig. 8: **Pruning supernodes.** Eliminating entire subsets or values from consideration using pruning.

4 Dataopsy: AQS for Multidimensional Data

We developed Dataopsy, a web-based visual analytics tool, to demonstrate AQS in practice. Dataopsy can be used in a standard web browser using any medium to large screen device For example, Figure 9 shows Dataopsy on a Samsung Galaxy S8 tablet device. In this section, we describe the visual interface of Dataopsy as well as query actions and interactions supported. We also include a video demonstration in the supplemental materials.

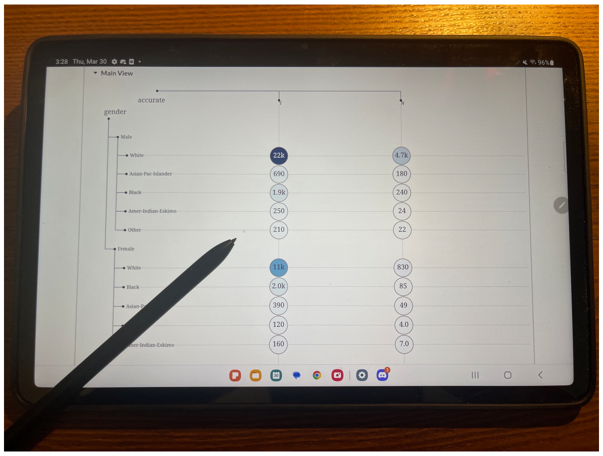


Fig. 9: **Dataopsy on a Samsung Galaxy S8 tablet.** The tool is particularly powerful on a mobile tablet device with touch or pen interaction.

4.1 Card Design

The central user interface (UI) component of Dataopsy is a card (Figure 1c), following the design of the popular UI component with the same name.[1] Cards are typically used to couple relevant information into a modular container. We chose cards as our core component as our system should support semantic substrates, views with identical functionalities albeit different underlying data subsets.

The default card, called Main, contains all data points in a single supernode (Figure 1a). From this initial card, the user can use the Projection operation to create a new substrate, which is then projected in a new card with identical design. The cards in our system are flexible. By default, they align horizontally and take 2/3 of the horizontal and vertical dimensions of the screen; but users can change their order, collapse them, or delete them at any time. Each card has two subcomponents: a header and a body.

4.1.1 Card Header

The header contains styled icons to support $\mathbb{P}^6$ (Figure 10). We do not include a separate icon for pivoting as partitioning or laying out the visual marks on the 2D space directly depends on pivoting. Instead, we provide two icons and dropdowns within them to partition horizontal and vertical axis. Clicking the prune icon deletes selected data points from the card. Similarly, clicking the project icon copies selected data points from the current card and opens a new card with the copied data. Pile option combines selected data points together. Users can optionally provide a name to the merged categories using a popup. A user selects data points by directly interacting with the visualization (described in Section 4.3).

All AQS operations are saved in an interaction log or stack (Figure 10d). Using this stack, a user can go back and forth between any stage of the exploration process. Further, we provide options to save and download the current state of the data as a CSV file. This is helpful for exporting data after transforming the original data using pruning and piling. Finally, a user can optionally configure data attributes such as defining alphabetical or numerical sorting options for the attributes.

4.1.2 Card Body

The card body contains the SVG container for the visualization. We describe the visualization design within the card body next.

4.2 Visual Representation

Dataopsy uses a 2D grid view to lay out the supernodes along the horizontal and vertical axis. Figure 11 shows an example representation of 300,000 posts among different communities on Reddit [27]. We describe the details about the representation below.

[1]https://www.nngroup.com/articles/cards-component/

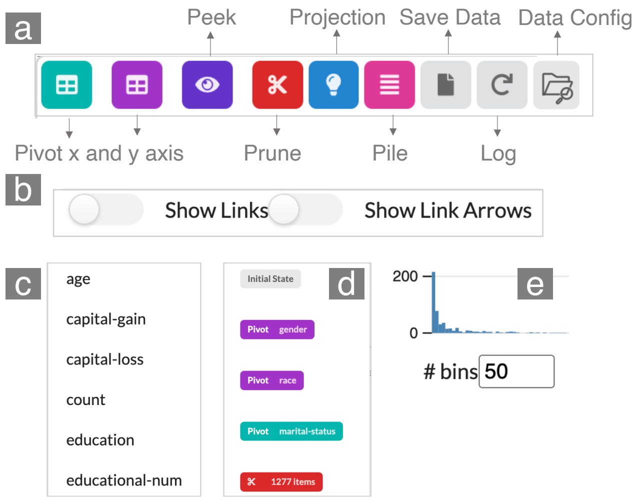


Fig. 10: **Different components of the card header.** (a) Buttons and dropdowns as styled icons for supporting $\mathbb{P}^6$. (b) Two checkboxes to toggle seeing links and link arrows. (c) A sample dropdown containing the data attributes that opens when a user clicks on either partition or peek icons. (d) Interaction log recorded by Dataopsy. This dropdown opens up when a user clicks on the Log icon. The user can go back and forth between different stages using this dropdown. (e) A histogram of the target variable. A user can select and prune data using this histogram (e.g., pruning data points with less than 5 occurrences).

4.2.1 Axis Labels

We used hierarchical grouping to place the labels on the geometric axes. The order of the variables on the hierarchy depends on the order they were added for partitioning by the user. For example, in Figure 11, we first add the readability index and then sentiment on the vertical axis. We also place variable names on the axis whenever space permits. On the vertical axis, we only show the first variable name as nested variable names may look indecent.

4.2.2 Supernodes

The visual marks in Dataopsy are typically aggregations of multiple data points, in contrast to the typical one mark per one data. We call these visual marks *supernodes*, although they may or may not have links depending on whether the domain is a network or not. In the current implementation of Dataopsy, we represent the supernodes with circles; however, they can be any 2D geometric shapes such as rectangles.

Dataopsy automatically determines the radius of the circles based on Equation 1.

$$S = \mathtt{min}(width/N_x, height/N_y)$$
$$r = \begin{cases} S, & \text{if } S > \alpha \\ \alpha, & \text{otherwise} \end{cases} \tag{1}$$

Here N_x and N_y are the numbers of categories on the horizontal and vertical axes. $(width, height)$ is the dimension of the SVG, inherited from the card body. We set α, the minimum possible radius, to 5. When $r = \alpha$, we update the size of the SVG by using Equation 2. However, we do not change the size of the card body; instead, we wrap the extended SVG within the card body and provide scrollbars to see the extended contents (see supplement for an example). This allows us to scale the representation for a large number of categories and avoid visual clutter.

$$\begin{aligned} width &= N_x \cdot r \\ height &= N_y \cdot r \end{aligned} \tag{2}$$

By default, each circle encodes the number of data points in the supernode using a linear color scale. However, the user can transform

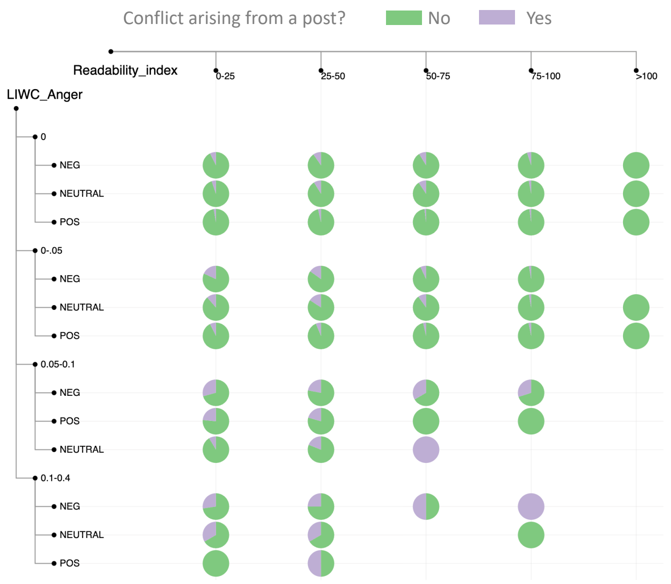


Fig. 11: **Two-dimensional grid view with hierarchical nesting.** Analyzing linguistic properties of 300,000 inter-community posts on Reddit [27]. Each row in this dataset contains information about a Reddit post from a source to a target community. We ⊞ partition the vertical axis by the use of anger-related words, according to Linguistic Inquiry and Word Count (LIWC) and sentiment of the posts, and the horizontal axis by the readability index. We 👁 peek at the variable of interest, whether or not a post starts a conflict between the source and target community. Posts with higher uses of anger-related words give rise to more conflicts among communities. Also, posts that are difficult to read (readability index > 100) have less chance of starting a conflict.

the circles into pie charts for seeing distribution along a new dimension using the 👁 peek operation. Figure 11 shows one example of 👁 peeking, where we see the number of conflicts arising from the Reddit posts in pie charts. We also place the number of data points at the center of the circles whenever space permits. For example, Figure 9 and 12 show the numbers in the circles whereas Figure 11 does not show the numbers due to the small size of the circles. The user can always see the numbers in a popup when hovering over the circles or pie charts.

4.2.3 Superlinks

Similar to the concept of supernodes, we call aggregated links connecting the supernodes *superlinks*. We encode edge weights with the thickness of the links. We used D3's `arc` function to draw the links. To reduce visual clutter, we bundle the links and set their default color to light gray with opacity set to 0.3. Despite these design decisions, too many links can create clutter. To avoid that, Superlinks are hidden by default. A user can choose to see all links by toggling the "Show Links" option (Figure 10b). Similarly, a user can choose to see direction arrows for the links by toggling "Show Link Arrows." Finally, on hovering over a node, we also highlight links originating and ending at the node with light purple and green colors, respectively (see Figure 13 and 15).

4.3 Interactions

We designed fluid interactions to help users perform AQS operations. There are multiple ways to interact with the visualization in Dataopsy.

The first is to interact with the axis labels. On hovering over an axis label, Dataopsy highlights all supernodes belonging to that row or column. Clicking labels will toggle selecting the whole row or column. After selection, a user can use the icon buttons in the card header (Figure 10) to ✂ prune, 💡 project, or ☰ pile the selected values.

Similar to the axis labels, a user can interact with the supernodes directly. On hovering over a node, we highlight the axis labels relevant to the node and show the number of data points belonging to the node in a popup. If the data contains links, we also show the links originating and ending at the node. On clicking a node, Dataopsy will toggle selecting the data points belonging to the node and allow a user to ✂ prune, 💡 project, or ☰ pile.

4.4 Implementation Notes

Dataopsy is currently a web-based prototype. We used Python running in a `flask` server as our backend. We used D3 for rendering the visualization in the frontend. All user interactions are supported by JavaScript. The source code and a demo of Dataopsy is available here: https://github.com/tonmoycsedu/Dataopsy

5 Case Studies and Application Example

Section 4 demonstrated how AQS can be used to analyze large-scale social media data (Figure 11). In this section, we demonstrate AQS using Dataopsy in four more scenarios: two case studies involving participants and two application examples.

5.1 Case Study: Data Exploration and Fairness Evaluation

ML and fairness researchers and practitioners often need to ensure how their models perform with respect to sensitive attributes (e.g., gender and race) in the datasets. This is important since ML models can inherit biases from datasets and propagate the biases in sensitive domains (e.g., loan approval, hiring, and healthcare allocation) [6, 18].

We worked with two data scientists and fairness researchers to explore how AQS and Dataopsy can be used to evaluate fairness of an ML model. The first participant (P1) is a male research scientist at a large technology company with a Ph.D. in Computer Science and more than 7 years of research experience in data science, visual analytics, and algorithmic fairness. The second participant (P2) is a female Ph.D. student of Computer Science with more than 4 years of research experience in data science and algorithmic fairness.

After a discussion with the participants, we decided to use the Adult Income dataset in this study. We chose this dataset as it is widely used in the algorithmic fairness and visual analytics literature [6, 17, 18]. The dataset contains 45,222 data points where each data point represents a person described by 14 attributes recorded from the U.S. 1994 census. Here, the prediction task is to classify if a person's income will be greater or less than $50,000 based on attributes such as age, gender, education, marital status, etc.

The first author of this paper met with the participants separately over Zoom. Before the meetings, we asked participants to train a classification model using the dataset. Both participants used Logistic Regression to train the model. Their models achieved 86% accuracy across the training and test sets (70%-30% split). Participants saved the predicted labels and original attributes in a CSV file.

Each study session started with a training phase where participants explored different features of Dataopsy using a training dataset. We encouraged participants to ask questions at this stage. After training, participants uploaded their saved CSV files to Dataopsy and analyzed the model performance. Participants followed a think-aloud protocol during the study. The sessions ended with semi-structured interviews focusing on the utility, limitations, and future directions of Dataopsy.

5.1.1 Results and Feedback

Evaluating Intersectional Fairness. To evaluate the fairness of the trained model, P1 started by ⊞ partitioning the horizontal axis to *train* and *test* sets and the vertical axis to *male* and *female* individuals (Figure 12a). P1 immediately noticed that the dataset is highly skewed towards men. Suspecting the skewed dataset might impact accuracy across the subsets, P1 visualized the ratio of the accurate predictions using the 👁 peek action (Figure 12b). However, the model performed better for females in terms of accuracy. To investigate further, P1 added race on the vertical axis to ⊞ partition male and female individuals (Figure 12c). P1 noticed that accuracies are consistent across training and test sets for all subsets. Among the subsets, *Female White* and *Male White* have the highest number of data points. P1 selected these two subsets and use the 💡 projection action to create a new substrate.

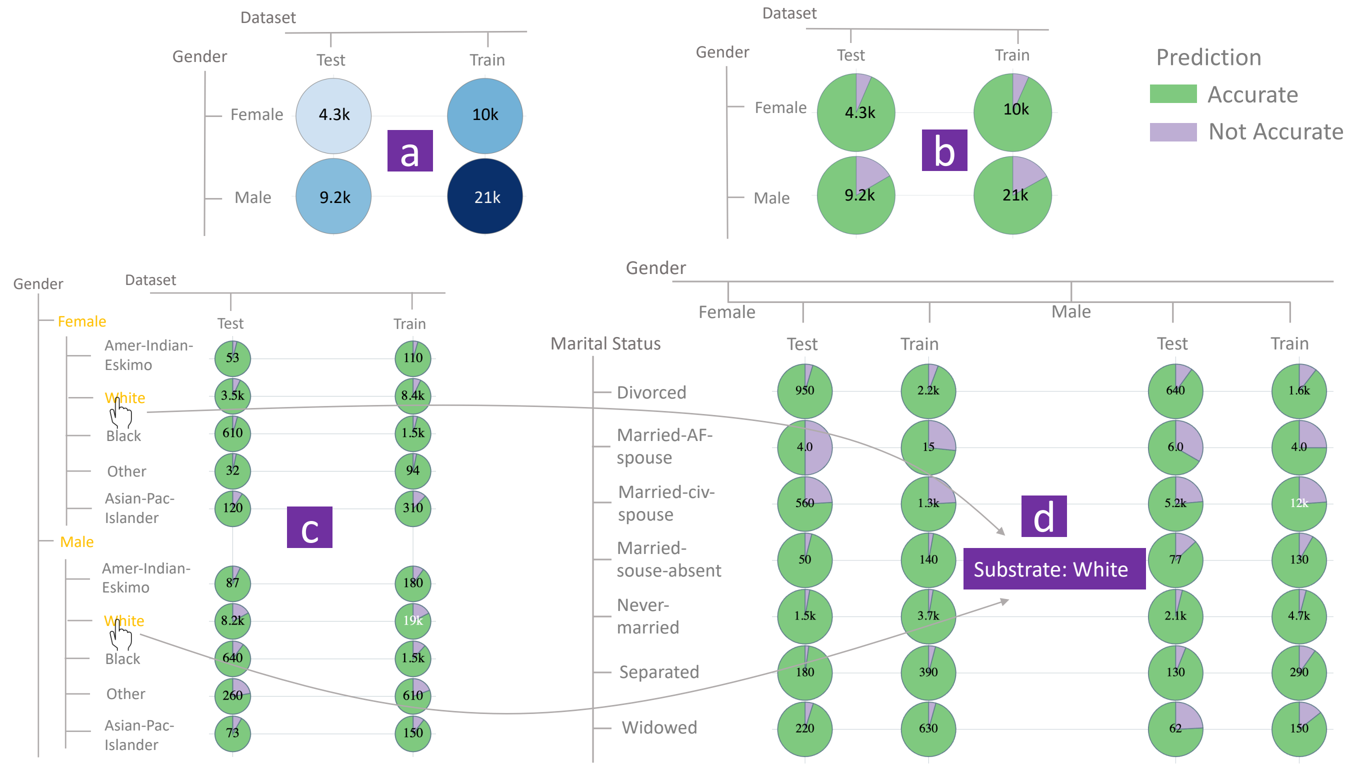


Fig. 12: **Evaluating Fairness in Income Prediction Dataset.** (a) Participant (P1) starts by ⊞ partitioning 45,000 data rows into MALE-FEMALE and TEST-TRAINING subsets. (b) Using the 👁 peek action, P1 evaluates the accuracy of the model on the subsets. (c) P1 further ⊞ partitions male and female subsets by race. P1 💡 projects FEMALE-WHITE and MALE-WHITE (i.e., all WHITE individuals) into a new substrate. (d) P1 now ⊞ partitions the new substrate (all WHITE) horizontally by gender and dataset and vertically by marital status.

P1 then restarted ⊞ partitioning the new substrate, this time by *marital status* on the vertical axis and *gender* and *dataset* on the horizontal axis. P1 immediately noticed that two subsets, *White Married-AF-spouse* and *White Married-civ-spouse*, were suffering from low accuracies (rows 2 and 3 in Figure 12d). Further, data points pertaining to *White Widowed Male* in the test set had a much lower accuracy than the ones in the training set. P1 continued this faceted browsing to find under-performing subsets in the dataset.

P2 followed a similar method to evaluate fairness across subsets. P2 mentioned that Dataopsy allows exploration of the full intersectional space, whereas comparable ML tools often allow only one or two dimensions (e.g., Google Facets).

EDA using Dataopsy. Both P1 and P2 thought Dataopsy is a useful tool for exploratory data analysis (EDA). P2's work requires extensive data analysis, where finding interesting insights often requires hours of coding in computational notebooks. P2 thought Dataopsy's faceted browsing provides a faster and more structured way to explore a dataset. P1's work often requires obtaining a balanced dataset across intersectional groups to reduce the chances of biases in model training. P1 thought Dataopsy is a handy tool to find empty or imbalanced groups or subsets in a dataset.

Recommendation for Dataopsy as a Fairness Tool. P1 and P2 provided several recommendations for adapting AQS in a dedicated fairness tool. One common suggestion was to add functionalities to answer "why" a subset of interest suffers from low accuracy after finding the subset. Such functionalities may include examining individual data rows (P2), observing the distribution of all features inside a subset (P1), and counterfactual analysis (P2). Another suggestion was to include more metrics to evaluate fairness (e.g., F1 score, and Theil Index).

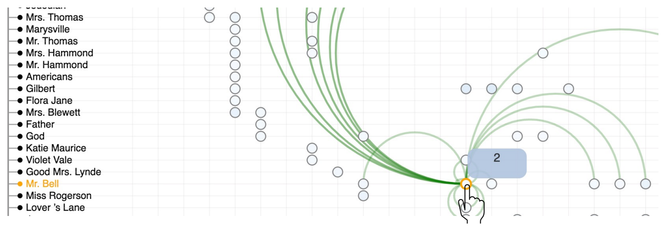


Fig. 13: **Visualizing a story in Dataopsy for writing an adapted screenplay.** Here we are seeing a partial view of the story *Anne of Green Gables* by L. M. Montgomery. The horizontal axis represents chapter and the vertical axis represents entities grouped by their type (person, place, etc). The color of circles represent the number of mentions, extracted using BookNLP. On hover, Dataopsy highlights how an entity (Mr. Bell) is connected to other entities in the story.

5.2 Case Study: Planning for an Adapted Screenplay

In this case study, we demonstrate AQS for a novel application domain: planning for an Adapted Screenplay. Screenplays dictate the making of films, TV shows, and stage performances. Screenplays are often adapted, where writers design their stories based on existing texts. An adapted screenplay is around 120 pages in length whereas a typical novel is around 400-800 pages. The critical challenge of writing an adapted screenplay is to decide what plotlines, characters, or places from the original story to include in the adapted story. With consultation from a creative writer (W1), we explore how AQS can be helpful in this scenario. The writer is a 27 years old female with published articles in their portfolio and a BA degree in English.

For this study, we met with the writer online via Zoom. Before the meeting, we asked the writer if they have a novel of choice. The

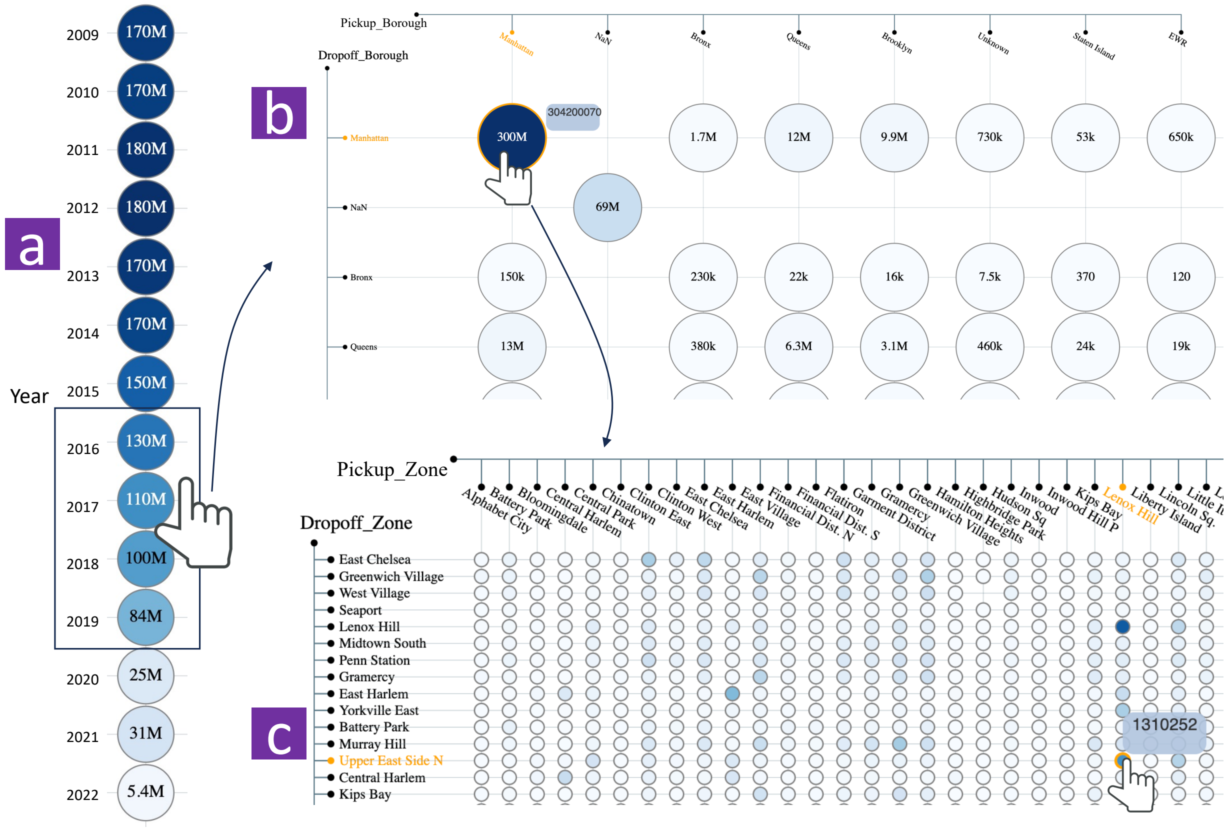


Fig. 14: **Exploring taxi rides in New York City (NYC).** (a) We ⊞ partition the whole dataset (around 1.7 billion trips) by YEAR. We select the trips (around 430M) from pre-pandemic years (2016-2019) and 💡 project them in a new view. (b) We ⊞ partition the pre-pandemic trips by the pickup and dropoff boroughs. Most trips (300M) originated and ended in Manhattan. (c) We can further drill down on the Manhattan trips by 💡 projecting them on a new view and ⊞ partitioning them by pickup and dropoff zones within Manhattan. On hover, we see the number of trips between a hotspot, trips between LENOX HILL and UPPER EAST SIDE NORTH.

writer chose *Anne of Green Gables* by L. M. Montgomery, a famous children's book with around 400 pages. We used BookNLP[2] to extract different types of entities (Persons, Facilities, and Locations) from the story. Similar to our previous case studies, the study included a training phase at the start of the session. We then asked the writer to visualize the collection of entities using Dataopsy and devise a skeleton of the proposed adapted screenplay with the help of the AQS operations.

5.2.1 Results and Feedback

Obtaining an Overview from Different Perspectives. W1 started exploring the story by ⊞ partitioning the horizontal axis into the 35 chapters of the book. They then ⊞ partitioned the vertical axis by the type of entities (persons, places, etc.), followed by the actual entities (Figure 13). W1 then quickly hovered over several entities to see how they are connected to each other.

W1 stated that ⊞ partitioning helped them to see the story from different perspectives. During the session, we noticed W1 continuously changing partitioning order. For example, sometimes W1 used entities and chapters to partition the vertical axis linearly. Other times, W1 used chapters on the horizontal axis and entities on the vertical.

Iterative Pruning and Piling. W1 found ✂ pruning and ≡ piling to be the most useful operations for developing a skeleton for the adapted screenplay. W1 started by pruning entities with low frequency and dependency in the story. Pruning allowed W1 to reduce several plotlines, characters, and places. W1 also used piling to reduce the size of the story. For example, after pruning several entities of a chapter, W1 piled (i.e., merged) the chapter with the previous chapter.

Recommendation for Dataopsy as a Writing Support Tool. W1 provided several suggestions for adopting AQS and Dataopsy in a writing support tool. One expected suggestion was to include a text editor and link the text with entities in the visualization. This would allow writers to see the context in the text and take informed decisions before pruning or piling. Another suggestion was to include social relations (e.g., brother, mother) as a feature so that writers can decide which characters to prune from a social circle.

5.3 Example: Understanding Taxi Trips in New York City

We present an application example on the New York City (NYC) taxi ride dataset [34] to demonstrate how AQS and Dataopsy scale to large datasets. The dataset contains every reported trip from 2009 to 2022 in NYC (approximately 1.7 billion trips). We chose this application because it is a large dataset (69 GB) with many facets for exploration.

The size of the dataset yields a range of analyses to perform. For this example, we show how past taxi rides can be analyzed to identify hotspots and devise a policy for allocating taxi cabs in 2022. Lockdowns and a lack of passengers during the COVID-19 pandemic (2020-2021) heavily disrupted NYC taxi service.[3] Many taxi drivers changed their profession during this time. As the world reopened after the pandemic, analyzing past taxi rides can inform the allocation policy of taxis throughout the city.

We used DASK, a Python library for parallel computing, to conduct the backend analysis. After loading the dataset, we first ⊞ partition the vertical axis by YEAR (Figure 14a). Applying the partition took 30 seconds for Dataopsy. The number of trips has gradually declined over

[2]https://github.com/booknlp/booknlp

[3]https://www.cnn.com/2021/01/09/us/yellow-taxi-drivers-new-york-covid/

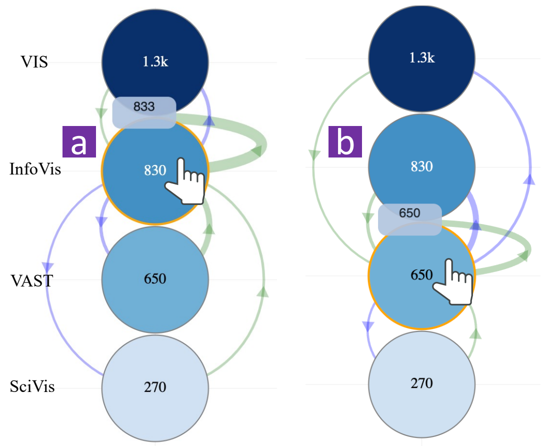


Fig. 15: **Internal citations among the four tracks of IEEE VIS.** (a) On hover, light purple links highlight InfoVis papers citing previous papers and light green links show papers citing InfoVis papers. (b) Similar analysis for VAST.

the years. As expected, we see a significant drop in 2020-2021. For demonstration purposes, we only select data from the first two months of 2022. As taxi rides should presumably return to pre-pandemic status, we select (i.e., Project) the trips from the most recent pre-pandemic years (2016-2019) for further investigation. We then partition the selected trips (430M) by their pickup and dropoff boroughs (Figure 14b). As expected, most taxi rides originated and ended in Manhattan. We can Project the Manhattan trips (300M) and further partition these trips by their pickup and dropoff zones within Manhattan (Figure 14c). We can now clearly see the hotspots within Manhattan. Note that the default *width* and *height* of the SVG are extended using Equation 2 due to the large number of zones. Please see the full screenshot of Figure 14c in the supplement. We can further explore the hotspots (e.g., projecting and partitioning a hotspot by month).

5.4 Example: Scientometric Analysis of IEEE VIS Pubs

Our final example is a scientometric analysis of IEEE VIS publications using the VisPub [24, 40] dataset. This example was designed to show how AQS can be used to analyze multivariate networks. We can explore this dataset at different levels of aggregation. For example, Figure 15a shows a citation network between four conference tracks: Vis, InfoVis, VAST, and SciVis. On hover, Dataopsy highlights references originated (light purple) at InfoVis and papers citing InfoVis papers (light green). We noticed that InfoVis papers cite their own papers the most. VAST papers also cite many papers from VAST, although they cite many InfoVis papers too (Figure 15b).

6 Discussion

Here we discuss design implications, limitations, and future work relevant to AQS and Dataopsy.

Changing Perspective with Partitioning. We noticed that participants often used different combinations to partition data, even within the same session. One key observation is that the order of partitioning can change the perspective even if the underlying data is the same. It can potentially affect what information or insights people see first. This phenomenon gives rise to several interesting questions for the VIS community such as *How exactly does the order of pivoting and partitioning impact the data exploration process?* and *Is there an optimum order to hierarchically nest the partitions?* Prior work on finding optimum ordering for parallel coordinates are inspiring in this case [39, 54].

Designing and Evaluating Fluid Interaction. Our case studies show promises for interaction design for exploring multivariate data. The supported interactions in AQS ($\mathbb{P}^6$) are larger than in a typical visualization system. Although AQS was not formally evaluated in the case studies, participants used praises such as "cool," "nice," and "wow" to describe the usability of Dataopsy. Our future work will focus on evaluating Dataopsy in comparison to similar methods (e.g., Google Facets [20, 49]). We can ask users to find answers to queries (e.g., What percentage of white, married, and female individuals were accurately labeled by the model?) and measure efficiency by counting and comparing the number of steps taken to answer the queries using different methods. We can further use NASA-TLX [22] and SUS [5] to evaluate the perceived workload and usability of Dataopsy.

Trade-offs between Aggregated and Unit Representation. AQS is a top-down technique where the exploration starts with a single mark aggregating all data items. This strong aggregation enables us to scale analysis to large datasets. However, compared to unit visualizations, there are fewer chances of serendipitous findings using our approach. Due to the lack of an overview, users need to have prior knowledge and hypotheses to construct the queries. We can partially address this limitation by introducing a recommender system that can recommend subsets using anomaly detection algorithms and prior user interactions [6, 37].

Scalability. As a theoretical concept, AQS is scalable to any number of data points. However, as an early prototype, Dataopsy currently lacks a few engineering features for handling "really big" datasets. For example, despite using parallel computing, for the NYC taxi ride dataset, on average, it took 30 seconds for Dataopsy to respond to the data operations (e.g., partitioning). There are established methods to handle such large datasets in visualization displays [10, 30, 32], which could further improve response time.

Visualizing Quantitative Values. Dataopsy has limitations for analyzing quantitative values. For example, when visualizing a quantitative attribute using peeking, Dataopsy uses binning and categorical color scales to show the distribution in a pie chart, which can be challenging to decode. One option is to extend the supported visual mark type (e.g., histogram in rectangles) to resolve this issue. We will follow the example of prior work such as Polaris [46] to integrate this feature into our tool. Another relevant problem with peeking is that it becomes difficult to measure the size of the nodes (i.e., cardinality) without the color saturation (Figure 11). A solution could be using circular curves along the circles/pies to indicate the size. Another possible solution is using varying circle sizes to represent cardinality.

Adopting Aggregate Query Sculpting. We recommend that practitioners and researchers adopt AQS if the following conditions are met:

- The data has a sufficient amount of facets (>=2).
- The number of data points is too many for unit visualization.
- The goal is to obtain higher-level insights and patterns rather than finding lower-level similarities between individual data points. (For example, AQS may not be feasible for finding clusters in an embedding space.)
- Domain-specific functionalities for the application are easy to integrate with AQS.

7 Conclusion

We have presented Aggregate Query Sculpting (AQS), a novel interaction technique for visualizing and exploring multivariate data. The goal of our work was to solve challenges for large-scale data containing many attributes. Visualizing such datasets using unit visualizations (e.g., scatter plots) often results in visual clutter and inelegant representation. We propose aggregation to be key for solving this issue. As a born scalable technique, AQS initially aggregates all data points into a single visual mark, a supernode. From there, AQS provides six operations, abbreviated as $\mathbb{P}^6$, to iteratively sculpt the data to a desired form. Based on the concept of AQS, we developed Dataopsy, a prototype tool for exploring multivariate data. Dataopsy is equipped to analyze multivariate data from versatile domains. We hope our work will motivate future research for designing visualization that is equipped to manage large-scale data, yet easy to explore.

8 Acknowledgments

This work was partially supported by the U.S. National Science Foundation grant 2211628. Any opinions, findings, and conclusions, or recommendations expressed here are those of the authors and do not necessarily reflect the views of the funding agencies.

References

[1] J. Abello, F. van Ham, and N. Krishnan. ASK-GraphView: A large scale graph visualization system. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):669–676, 2006. doi: 10.1109/TVCG.2006.120 3

[2] C. Ahlberg and B. Shneiderman. The Alphaslider: a compact and rapid selector. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pp. 365–371. ACM, New York, NY, USA, 1994. doi: 10.1145/191666.191790 2

[3] A. Aris and B. Shneiderman. Designing semantic substrates for visual network exploration. *Information Visualization*, 6(4):281–300, 2007. doi: 10.1057/palgrave.ivs.9500162 2, 3

[4] A. Bezerianos, F. Chevalier, P. Dragicevic, N. Elmqvist, and J. Fekete. GraphDice: A system for exploring multivariate social networks. *Computer Graphics Forum*, 29(3):863–872, 2010. doi: 10.1111/j.1467-8659.2009.01687.x 3

[5] J. Brooke et al. SUS: A quick and dirty usability scale. *Usability Evaluation in Industry*, 189(194):4–7, 1996. doi: 10.1201/9781498710411 9

[6] Á. A. Cabrera, W. Epperson, F. Hohman, M. Kahng, J. Morgenstern, and D. H. Chau. FAIRVIS: visual analytics for discovering intersectional bias in machine learning. In *Proceedings of the IEEE Conference on Visual Analytics Science and Technology*, pp. 46–56. IEEE Computer Society, Los Alamitos, CA, USA, 2019. doi: 10.1109/VAST47406.2019.8986948 2, 6, 9

[7] J. Choi, D. G. Park, Y. L. Wong, E. R. Fisher, and N. Elmqvist. VisDock: A toolkit for cross-cutting interactions in visualization. *IEEE Transactions on Visualization and Computer Graphics*, 21(9):1087–1100, 2015. doi: 10.1109/TVCG.2015.2414454 2

[8] M. Csikszentmihalyi. *Flow: The Psychology of Optimal Experience*. Harper Collins, New York, NY, USA, 1991. 3

[9] R. Dachselt, M. Frisch, and M. Weiland. FacetZoom: a continuous multiscale widget for navigating hierarchical metadata. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pp. 1353–1356. ACM, New York, NY, USA, 2008. doi: 10.1145/1357054.1357265 2

[10] C. A. de Lara Pahins, S. A. Stephens, C. Scheidegger, and J. L. D. Comba. Hashedcubes: Simple, low memory, real-time visual exploration of big data. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):671–680, 2017. doi: 10.1109/TVCG.2016.2598624 9

[11] M. Dörk, N. H. Riche, G. A. Ramos, and S. T. Dumais. PivotPaths: Strolling through faceted information spaces. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2709–2718, 2012. doi: 10.1109/TVCG.2012.252 2

[12] S. Drucker and R. Fernandez. A unifying framework for animated and interactive unit visualizations. Technical Report MSR-TR-2015-65, Microsoft Research, August 2015. 2

[13] N. Elmqvist, T. Do, H. Goodell, N. Henry, and J. Fekete. ZAME: interactive large-scale graph visualization. In *Proceedings of the IEEE VGTC Pacific Symposium on Visualization*, pp. 215–222. IEEE Computer Society, Los Alamitos, CA, USA, 2008. doi: 10.1109/PACIFICVIS.2008.4475479 3

[14] N. Elmqvist, P. Dragicevic, and J. Fekete. Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1539–1148, 2008. doi: 10.1109/TVCG.2008.153 2, 3

[15] N. Elmqvist and J. Fekete. Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines. *IEEE Transactions on Visualization and Computer Graphics*, 16(3):439–454, 2010. doi: 10.1109/TVCG.2009.84 1, 2, 3, 4

[16] N. Elmqvist, A. V. Moere, H. Jetter, D. Cernea, H. Reiterer, and T. J. Jankun-Kelly. Fluid interaction for information visualization. *Information Visualization*, 10(4):327–340, 2011. doi: 10.1177/1473871611413180 3

[17] B. Ghai, M. Mishra, and K. Mueller. Cascaded debiasing: Studying the cumulative effect of multiple fairness-enhancing interventions. In *Proceedings of the International Conference on Information & Knowledge Management*, pp. 3082–3091. ACM, New York, NY, USA, 2022. doi: 10.1145/3511808.3557155 1, 2, 6

[18] B. Ghai and K. Mueller. D-BIAS: A causality-based human-in-the-loop system for tackling algorithmic bias. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):473–482, 2023. doi: 10.1109/TVCG.2022.3209484 6

[19] S. Ghani, B. C. Kwon, S. Lee, J. S. Yi, and N. Elmqvist. Visual analytics for multimodal social network analysis: A design study with social scientists. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2032–2041, 2013. doi: 10.1109/TVCG.2013.223 3

[20] Google People + AI Research. Facets – visualization for ML datasets. `https://pair-code.github.io/facets/`, 2017. 2, 9

[21] C. Harrison. Appropriated interaction surfaces. *Computer*, 43(6):86–89, 2010. doi: 10.1109/MC.2010.158 1

[22] S. G. Hart. NASA-task load index (NASA-TLX); 20 years later. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 50(9):904–908, 2006. doi: 10.1177/154193120605000909 9

[23] C. Hurter, B. Tissoires, and S. Conversy. FromDaDy: Spreading aircraft trajectories across views to support iterative queries. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1017–1024, 2009. doi: 10.1109/TVCG.2009.145 2, 4

[24] P. Isenberg, F. Heimerl, S. Koch, T. Isenberg, P. Xu, C. D. Stolper, M. Sedlmair, J. Chen, T. Möller, and J. T. Stasko. Vispubdata.org: A metadata collection about IEEE visualization (VIS) publications. *IEEE Transactions on Visualization and Computer Graphics*, 23(9):2199–2206, 2017. doi: 10.1109/TVCG.2016.2615308 2, 9

[25] W. Javed and N. Elmqvist. ExPlates: Spatializing interactive analysis to scaffold visual exploration. *Computer Graphics Forum*, 32(3):441–450, 2013. doi: 10.1111/cgf.12131 2

[26] S. Kairam, N. H. Riche, S. M. Drucker, R. Fernandez, and J. Heer. Refinery: Visual exploration of large, heterogeneous networks through associative browsing. *Computer Graphics Forum*, 34(3):301–310, 2015. doi: 10.1111/cgf.12642 3

[27] S. Kumar, W. L. Hamilton, J. Leskovec, and D. Jurafsky. Community interaction and conflict on the web. In *Proceedings of the ACM Conference on the World Wide Web*, pp. 933–943. ACM, New York, NY, USA, 2018. doi: 10.1145/3178876.3186141 2, 5, 6

[28] B. Lee, R. Dachselt, P. Isenberg, and E. K. Choe, eds. *Mobile Data Visualization*. AK Peters Visulization Series. Chapman & Hall, 2021. 1

[29] B. Lee, G. Smith, G. G. Robertson, M. Czerwinski, and D. S. Tan. FacetLens: exposing trends and relationships to support sensemaking within faceted datasets. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pp. 1293–1302. ACM, New York, NY, USA, 2009. doi: 10.1145/1518701.1518896 2

[30] Z. Liu, B. Jiang, and J. Heer. *imMens*: Real-time visual querying of big data. *Computer Graphics Forum*, 32(3):421–430, 2013. doi: 10.1111/cgf.12129 9

[31] Z. Liu, S. B. Navathe, and J. T. Stasko. Ploceus: Modeling, visualizing, and analyzing tabular data as networks. *Information Visualization*, 13(1):59–89, 2014. doi: 10.1177/1473871613488591 3

[32] D. Moritz, B. Howe, and J. Heer. Falcon: Balancing interactive latency and resolution sensitivity for scalable linked visualizations. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pp. 694:1–694:11. ACM, New York, NY, USA, 2019. doi: 10.1145/3290605.3300924 9

[33] T. Munzner. *Visualization Analysis and Design*. AK Peters Visualization Series. CRC Press, Boca Raton, FL, USA, 2014. 1, 2

[34] New York City Taxi & Limousine Commission. TLC trip record data. `https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page`, 2023. 2, 8

[35] C. Nobre, M. D. Meyer, M. Streit, and A. Lex. The state of the art in visualizing multivariate networks. *Computer Graphics Forum*, 38(3):807–832, 2019. doi: 10.1111/cgf.13728 2

[36] D. A. Norman and S. W. Draper, eds. *User Centered System Design: New Perspectives on Human-Computer Interaction*. Lawrence Erlbaum Associates, Hillsdale, NJ, USA, 1986. 3

[37] A. Pandey, A. Srinivasan, and V. Setlur. Medley: Intent-based recommendations to support dashboard composition. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):1135–1145, 2023. doi: 10.1109/TVCG.2022.3209421 9

[38] D. Park, S. M. Drucker, R. Fernandez, and N. Elmqvist. ATOM: A grammar for unit visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 24(12):3032–3043, 2018. doi: 10.1109/TVCG.2017.

2785807 1, 2, 3

[39] G. E. Rosario, E. A. Rundensteiner, D. C. Brown, M. O. Ward, and S. Huang. Mapping nominal values to numbers for effective visualization. *Information Visualization*, 3(2):80–95, 2004. doi: 10.1057/palgrave.ivs.9500072 9

[40] A. Sarvghad, R. Franqui-Nadal, R. Reznik-Zellen, R. Chawla, and N. Mahyar. Scientometric analysis of interdisciplinary collaboration and gender trends in 30 years of IEEE VIS publications. *IEEE Transactions on Visualization and Computer Graphics*, 7(1):3340–3353, 2022. doi: 10.1109/TVCG.2022.3158236 9

[41] J. Seo and B. Shneiderman. Interactively exploring hierarchical clustering results. *Computer*, 35(7):80–86, 2002. doi: 10.1109/MC.2002.1016905 2

[42] B. Shneiderman. Direct manipulation: A step beyond programming languages. *Computer*, 16(8):57–69, 1983. doi: 10.1109/MC.1983.1654471 3

[43] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the IEEE Symposium on Visual Languages*, pp. 336–343. IEEE Computer Society, Los Alamitos, CA, USA, 1996. doi: 10.1109/VL.1996.545307 2

[44] B. Shneiderman and A. Aris. Network visualization by semantic substrates. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):733–740, 2006. doi: 10.1109/TVCG.2006.166 2, 3

[45] G. Smith, M. Czerwinski, B. Meyers, D. C. Robbins, G. G. Robertson, and D. S. Tan. FacetMap: A scalable search and browse visualization. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):797–804, 2006. doi: 10.1109/TVCG.2006.142 2

[46] C. Stolte and P. Hanrahan. Polaris: A system for query, analysis and visualization of multi-dimensional relational databases. In *IEEE Symposium on Information Visualization*, pp. 5–14. IEEE Computer Society, 2000. doi: 10.1109/INFVIS.2000.885086 2, 4, 9

[47] F. van Ham and A. Perer. "Search, Show Context, Expand on Demand": Supporting large graph exploration with degree-of-interest. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):953–960, 2009. doi: 10.1109/TVCG.2009.108 3

[48] M. Wattenberg. Visual exploration of multivariate graphs. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pp. 811–819. ACM, New York, NY, USA, 2006. doi: 10.1145/1124772.1124891 2, 4

[49] J. Wexler. Facets: An open source visualization tool for machine learning training data. `https://ai.googleblog.com/2017/07/facets-open-source-visualization-tool.html`, 2017. 2, 9

[50] C. Williamson and B. Shneiderman. The dynamic homefinder: Evaluating dynamic queries in a real-estate information exploration system. In *Proceedings of the ACM Conference on Research and Development in Information Retrieval*, pp. 338–346. ACM, New York, NY, USA, 1992. doi: 10.1145/133160.133216 2

[51] M. A. Yalçin, N. Elmqvist, and B. B. Bederson. AggreSet: Rich and scalable set exploration using visualizations of element aggregations. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):688–697, 2016. doi: 10.1109/TVCG.2015.2467051 2

[52] M. A. Yalçin, N. Elmqvist, and B. B. Bederson. Keshif: Rapid and expressive tabular data exploration for novices. *IEEE Transactions on Visualization and Computer Graphics*, 24(8):2339–2352, 2018. doi: 10.1109/TVCG.2017.2723393 2

[53] K. Yee, K. Swearingen, K. Li, and M. A. Hearst. Faceted metadata for image search and browsing. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pp. 401–408. ACM, 2003. doi: 10.1145/642611.642681 2, 3

[54] Z. Zhang, K. T. McDonnell, and K. Mueller. A network-based interface for the exploration of high-dimensional data spaces. In *Proceedings of the IEEE Pacific Symposium on Visualization*, pp. 17–24. IEEE Computer Society, Los Alamitos, CA, USA, 2012. doi: 10.1109/PacificVis.2012.6183569 9