

Keshif: Rapid and Expressive Tabular Data Exploration for Novices

Mehmet Adil Yalçın, Niklas Elmqvist, *Senior Member, IEEE*, and Benjamin B. Bederson

Abstract—General purpose graphical interfaces for data exploration are typically based on manual visualization and interaction specifications. While designing manual specification can be very expressive, it demands high efforts to make effective decisions, therefore reducing exploratory speed. Instead, principled automated designs can increase exploratory speed, decrease learning efforts, help avoid ineffective decisions, and therefore better support data analytics novices. Towards these goals, we present Keshif, a new systematic design for tabular data exploration. To summarize a given dataset, Keshif aggregates records by value within attribute summaries, and visualizes aggregate characteristics using a consistent design based on data types. To reveal data distribution details, Keshif features three complementary linked selections: highlighting, filtering, and comparison. Keshif further increases expressiveness through aggregate metrics, absolute/part-of scale modes, calculated attributes, and saved selections, all working in synchrony. Its automated design approach also simplifies authoring of dashboards composed of summaries and individual records from raw data using fluid interaction. We show examples selected from 160+ datasets from diverse domains. Our study with novices shows that after exploring raw data for 15 minutes, our participants reached close to 30 data insights on average, comparable to other studies with skilled users using more complex tools.

Index Terms—Interactive data exploration and discovery, data visualization, data exploration, graphical user interfaces, interaction, design, user-centered design

1 INTRODUCTION

Visual data exploration is often performed using visualization design environments (VDEs) (such as Lyra [29], iVisDesigner [25], and Tableau [31]) that enable constructing visualizations and interactions based on rich visual grammars, interactive features, and data pipelines. VDEs are also designed to support explanatory tasks, such as storytelling and interactive infographics. As a result, VDEs typically define highly expressive configuration spaces that support complex queries. However, they require users to make many decisions to create effective data views. This process demands high cognitive effort, requires knowledge and experience, and reduces exploratory speed, affecting both novices [11] and experts [3].

We propose that automated and principled designs can reduce learning and analysis efforts, help the user avoid ineffective decisions, increase exploration speed and in turn make data exploration more welcoming to novices. Specifically, we present Keshif, a tool with a new systematic design for exploration of generic tabular data. To streamline the exploration process from overview to detail, Keshif condenses data attributes using summaries which present aggregations of records by value, and re-

veals distributions of data subsets using three complementary linked selections: (i) highlighting (rapidly previewing record groups), (ii) filtering (focusing on a record group), and (iii) comparison (locking selection of record groups). This minimalist approach significantly reduces visualization or interaction decisions in comparison to manual / highly configurable design environments.

Keshif automatically selects a visual form for aggregations depending on the data type, and aims to offer perceptual clarity, consistency, and scalability in the number of records, while reducing barriers for choosing visual encodings [11]. In addition to summary visualizations, individual records are shown in a separate view, either as a sorted list, as a map to reveal spatial features, as a node-link diagram to reveal references across records, or as a scatterplot to reveal relations across numeric attributes. Keshif also offers an expressive set of integrated rich data exploration options including aggregate measure metrics, visual scale modes, calculated attributes, and saved selections, as well as unconventional visual forms for percentile and set-pair analysis [39]. All of these analytical features are fully synchronized across the interface in order to maintain consistency of exploration and clarity of visualizations. With these strategies, Keshif aims to help novice analysts focus on data-driven insights, rather than visualization and interaction specifications.

Keshif's automated, minimalist model also supports simple drag-and-drop authoring of dashboards from raw data. Dashboards can be shared on the web with a unique URL, embedded into web pages, and customized with basic web programming. As a result, Keshif provides an out-of-the-box tabular data exploration environment that enables rapid data exploration for novices.

* Keşif (keshif) means discovery and exploration in Turkish. The tool is available at keshif.me and github.com/adilyalcin/keshif.

- Mehmet Adil Yalçın is with Keshif, LLC. E-mail: yalcin@keshif.me.
- Niklas Elmqvist is with University of Maryland, College Park, MD 20742. E-mail: elm@umd.edu.
- Benjamin B. Bederson is with University of Maryland, College Park, MD 20742. E-mail: bederson@umd.edu.

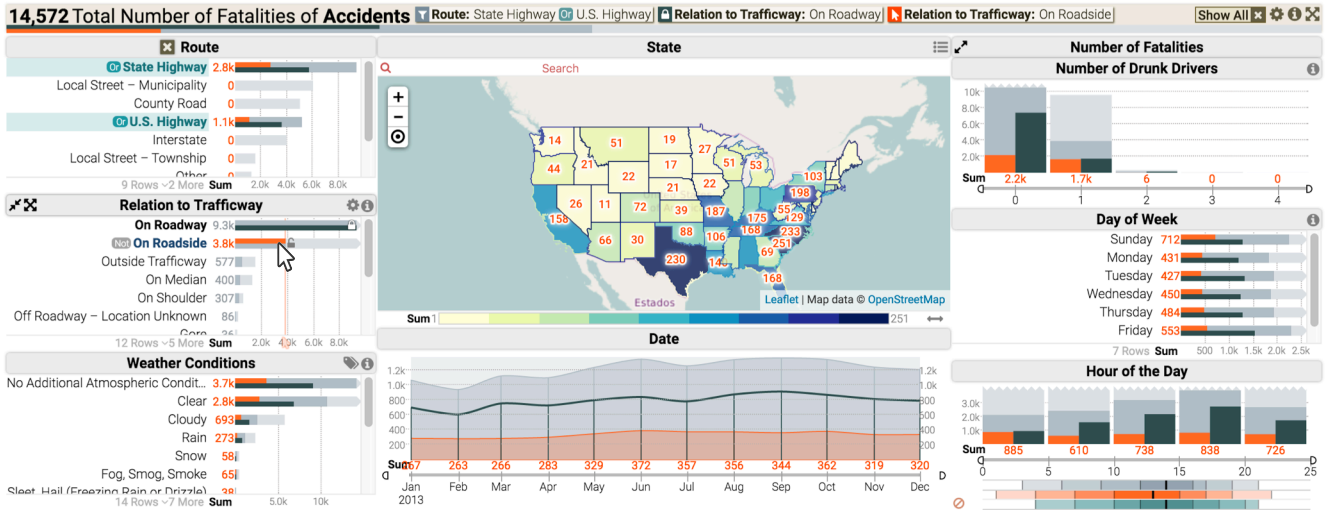


Fig. 1. This Keshif dashboard enables exploration of fatal traffic accidents in 2013 in the United States. Selected attributes are summarized using data aggregations, measuring the total number of fatalities. Visualizations show data distributions of three linked selections (☑, ☑, ☑). This view shows bar, line, map, and percentile charts. Accidents on State Highway or U.S. Highway (route categories) are selected by filtering ☑. Roadway accidents are selected by locking ☑, and roadside accidents are selected by highlighting ☑ on mouse-over. This dashboard and exploratory view can be easily and rapidly authored from raw data using the graphical interface; saved, shared, and embedded into existing web pages.

We evaluated Keshif for exploratory data analysis using the insight-based methodology with visual analytics novices. After a short training, participants explored raw datasets in a casual, unguided setting. The results indicate that Keshif enables rapid learning, authoring, and data discovery, with our participants averaging close to two insights shared per minute. We also validated the design of Keshif through the construction of 160+ dashboards of public datasets across many disciplines.

2 RELATED WORK

Visualization design environments (VDEs) enable specifying visualizations using graphical interfaces that commonly support drag-and-drop placement along with traditional menu options. Tableau [31] uses a shelf-based authoring approach for high-level specifications, and an internal formal language to generate queries and calculate data features. Lyra [29] and iVisDesigner [25] include lower-level abstractions of marks, drop-zones, connectors, handles, and data pipelines. However, encoding data-to-visual relationships is a bottleneck for infovis novices [11]; they commonly prefer familiar, simple visualizations, and “failure to choose appropriate views” becomes a roadblock [17].

To simplify the data exploration process, **visualization recommendation** approaches can be used. Tableau’s *Show Me* [21] uses a rule-based approach on selected attribute types to recommend a short list of charts based on its visual query language. This provides a higher-level charting layer, but does not fully hide the underlying customization complexities. Voyager [37] uses faceted browsing of visualizations which are dynamically generated from attribute variations using a formal grammar. VizDeck [15] uses a card game metaphor based on simple statistical data properties. SAGE [27] uses a knowledge-based semi-automated approach. SeeDB [34] focuses on computational challenges using a formal complex database query

model and an optimized query execution engine. Golemati et al. consider user context [10], and Elzen and van Wijk [9] use small multiples to show parameter and value option variations of a single large chart, explicitly constraining the exploration flow from one chart to another. In contrast, we (i) use a set of fixed visual representations and interactions designed to support accurate perception in statistical graphics [7] and to facilitate a rapid exploration flow from overview to details, (ii) give the user full control in selecting attributes, queries, and few well-defined visualization modes for expressiveness, and (iii) offer few semantic visual alternatives where relevant.

Single-chart visualizations: Chart templates offer a generalized solution for data visualization. They require explicit selection of the chart template (with available options), followed by the specification of data-to-visual encodings within the template slots. ManyEyes [35] was among the first platforms to offer visualization templates as a web service for many chart types, also supporting data upload, hosting, and commenting. Spreadsheets (e.g. Microsoft Excel, Google Sheets) also offer charting with templates and data specifications. However, templated charts present a bottleneck for novices by requiring visual decisions upfront, and some decisions may lead to ineffective data views.

Coordinated Multiple Views (CMVs). Powerful data exploration approaches typically use multiple interactive views which are coordinated during interaction (such as by brushing and linking) to reveal relationships across multiple views. Roberts [26] provides a survey on CMVs. Snap-together [23] treats coordination as database join queries. Improvise [36] provides a rich, customizable coordination model on shared objects and dependencies. These systems target expert users and developers, offering high flexibility with a wide range of use cases and patterns. Their design approach relies on many menus and configuration options. As Roberts [26] notes, “Concurrently they (developers) need to decide how the in-

formation will be aggregated or abstracted and finally work out how the user interacts with the system.” Novices are particularly disadvantaged from these shortcomings. The notion that “theoretically any operation can be coordinated between multiple views” [5], [26], [36] does not consider the increased costs on usability, discoverability, learnability, and decision making to explore relations.

Domain-specific systems present design solutions, guidelines, and case studies based on a detailed analysis of domain requirements. Examples include energy portfolios [6], online communities [16], funding portfolios [22], temporal transactions [20], and literature surveys [2]. Domain specific systems can assume or emphasize specific properties or relations within their domain, yet potentially limit generalizations, i.e. transfer of solutions across datasets and domains. For example, SurVis [2] focuses on literature datasets including keywords and citations. In contrast, we generalize exploration of self-referencing attributes as node-link charts, exploration of categories as sorted histograms, and offer a unified interaction model independent of underlying data domain.

Foundations. Faceted browsing [41] (based on query previews [12]) has become a ubiquitous model of organizing and browsing tabular datasets. Dynamic queries [1] enable querying of data using interface elements such as links, buttons, sliders, and maps. Similarly, Keshif tightly integrates visual representations and interaction, and extends the minimalist design basis of [1], [33], [41] for rich exploration by including rich visualizations of multiple selections and aggregate measures. Compared to AggreSet [39], which focuses on visualizing set-typed data, we present a full data exploration workflow from raw data (authoring and sharing dashboards), and an extended model on data types (percentiles, maps, time), aggregate measures, and multiple compare selections.

Web-based systems. Exhibit [13] allows construction of faceted data interfaces using XML specifications. Likewise, Keshif is easy to deploy, while also providing richer exploratory features and a graphical authoring. VisGets [8] provides an exploratory interface for time (histogram), location (bubble map) and tags (word clouds) in document collections. Compared to our system, it does not define a generalized visualization and interaction model, does not support selections to enable side-by-side comparisons, and does not support graphical authoring. Its user evaluation is limited to self-reported usability, in contrast to our in-depth insight-based methodology.

3 DESIGN GOALS

Our main goal is to lower barriers to exploring tabular data, and to enable many insights from data in a short time. We summarize our specific goals below.

Support novice audiences. The design should allow data analysis novices to quickly learn and perform data exploration to find data-driven insights rapidly. The data analysis barriers are higher for novices with less existing knowledge. We prioritize established techniques over experimental designs. For Keshif, these techniques include faceted browsing and coordinated multiple views.

Focus on exploratory tasks, instead of presentation, customizations, annotations, and storytelling. The design should create an explicit overview-to-detail flow [30] for exploration from basic to detailed features and insights.

Aim for a minimalist, automated design. A core set of features that are seamlessly integrated with consistency can reduce user efforts and potential errors with fewer representations and interactions to master. Designs can be automated to help avoiding ineffective user decisions, minimizing mistakes, and increasing productive time on exploration, essentially supporting different cognitive activities [38]. The interface should have high data-ink ratio [32], and few component types. Synchronizing selection and visualization states across all components by default increases consistency. Additional configurations should not be necessary to create effective visualizations or to perform data queries. Multivariate exploration can be enabled with synchronized, dynamic, basic charts rather than complex charts which visualize multiple variables concurrently using multiple encodings.

Increase expressiveness. Data analysis and insights can be enriched with complementary analytical modes. The challenge is in creating designs that increase expressiveness without sacrificing ease of use and sensible defaults. Data transformations can also enrich data analysis.

Enable simple authoring of the exploratory environment. Exploration requires not only querying and observing relations, but adding and adjusting data perspectives.

Target raw tabular data. Tabular data is a very common and descriptive data format. Each record (row) presents a single observation (event, object, etc.) with multiple attributes (columns), which are commonly categorical or interval (numeric or timestamp). This structure supports flexible querying and many perspectives for data exploration. We extend this data model beyond existing practices to include multi-valued categorical values (sets) [33], spatial regions per category or record, and attributes which link records (i.e. to create networks).

We also aimed for an implementation that would be easy to deploy, share, and maintain on the web natively. We preferred a lean technology stack, and avoided complex frameworks and server/backend setups.

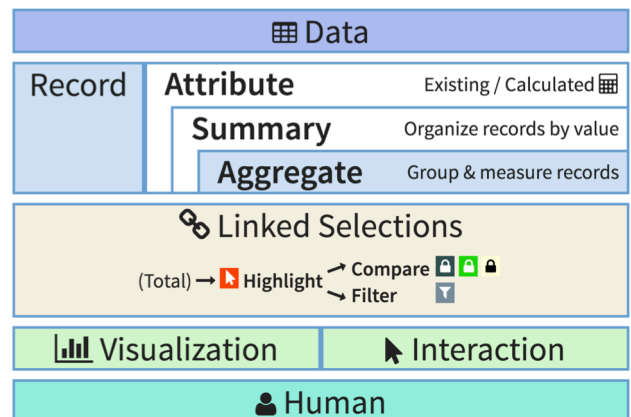


Fig. 2. The aggregate summaries and linked selection model creates a data↔human interface. Data consists of records with attributes. Attributes are summarized to aggregates, which measure record group metrics. Three linked selections enable the dialogue with data.

4 KESHIF: DESIGN, MODEL, AND FEATURES

Keshif is a unified visual and interactive data exploration design that enables rapid and expressive tabular data exploration for analytics novices. The **aggregate summaries and linked selection model** provides a simple yet flexible foundation (Figure 2). Attributes are summarized by aggregating records and measuring group metrics. The visualizations and summaries are designed per data type and semantics (Table 1, 3 and 4). Linked selections enable querying and visualizing multiple distributions and record features. To increase **exploratory expressiveness** compared to linked faceted browsing, Keshif uses a fully-integrated combination of *aggregate metrics* (count/

sum/average), *measure scale mode* (absolute/part-of), *measure label mode* (absolute/percentage), *calculated attributes* (data transformations), *saved selections* (flexible queries) and *highlight & compare selections*, as well as alternative semantic visual forms (list/ map/ network/scatterplot), and percentile and set-pair charts.

A Keshif dashboard (Figure 1, 3, 5, 6, 7, 8) is a synchronized data exploration space based on attribute summaries and a record display. Its layout is designed to reduce positioning decisions at exploration time, and to avoid overlaps across summaries and the record display. The layout includes four panels that can display multiple summaries (Figure 7), the record display in the middle, and the header that displays active exploration state and

Data Type	Glyph	Visualization	Data Type	Glyph	Visualization
Category	Bar (Category)	Absolute Scale A 276 B 256 C 148 D 81	Number	Bar (Interval range bin)	Absolute Scale A 73% B 86% C 66% D 79%
		Part-of Scale			Part-of Scale
	Encoding	→ Length (Width)	Encoding	↑ Length (Height)	
	Position	Category order, next to category label		Position	Interval Range
Time	Line (Interval range bin)	Absolute Scale 49 42 81	Percentile Distribution	Block (Percentile range)	Part-of Scale 50% 59% 60%
		Part-of Scale			Distribution of a numerical attribute. Simple alternative to box-plots without visualization of outliers. Percentiles are independent of scale mode.
	Encoding	↑Length for measure value. ↔Line connects bins. Area-fill for non-compare selections.	Encoding	Color: Four fixed percentile ranges with 10% steps. Darker color towards the median (50%).	
	Position	Interval Range		Position	The percentile ranges of the selected records
Set Pair (Multi-Value Category)	Disc	Absolute Scale	Spatial Area	Region (Map)	Part-of Scale
		Part-of Scale			In part-of scale, color is scaled from 0% to the maximum % value of all (filtered) regions.
	Encoding	Filtered: ⊙ Circular area. Highlighted: Arc area (0°-360°) Compared: Arc border (0°-360°) Total: None. Exists: Cell background color. Strength: Circle color (part-of scale). For details, see AggreSet [33].	Encoding	Color: [0 - max(distribution)]. Visualizes one distribution by color mapping. Default is filtered selection. Highlight-selection takes precedence when enabled.	
	Position	Set-pair location on grid. Small glyph size.		Position	Geographically defined. Fixed shape and size.
No-Value (Missing)	Icon	Aggregates records with no-value in summary.	All Records (Global)	Bar (Full width)	2,294 Victims
					Encoding
	Position	Fixed (Lower-left corner of summary)	Position	Fixed (Top of the dashboard)	

Table 1. Visual aggregate encodings for common data types. The design aims to minimize overlaps, assist accurate graphical perception, achieve scalability and consistency, and support multiple selections. Data transitions on visual encodings are animated.

Summary	Form	Navigation	Tasks and Features
Categorical	List	↓ Scroll (1D)	<ul style="list-style-type: none"> ♦ Sorting: Automated re-sorting with staged animations after filtering to emphasize most relevant first. ♦ Supports multiple sorting options, custom ordering (ordinal data), and inverse sorting.
	Map	⊕ Pan & Zoom (2D)	<ul style="list-style-type: none"> ♦ Select records by spatial (rectangle) query.
Interval	Histogram (Numeric)	<ul style="list-style-type: none"> ⊕ Zoom to filtered range ⊕ Zoom to total range (Finer vs. coarser bins) 	<ul style="list-style-type: none"> ♦ Linear/log scale binning, based on data distribution. ♦ Supports percentile aggregations. ♦ Supports unit names (10 mg, \$100, etc.)
	Line (Time)		<ul style="list-style-type: none"> ♦ Only linear-scale binning.
Set-Pair	Matrix	⊕ Pan & Zoom (2D)	<ul style="list-style-type: none"> ♦ Visualizes set-pair strength & subset relations (design on data-semantic) ♦ Connected (next) to categorical list summary with synchronized scrolling navigation.



Table 2. Tasks on different summary and form types are designed to reflect data types and semantics.

Aggregate Metric	Metric Summary	Relation across selection values (distributions)	Measure Scale	
			Absolute (Shared scale within summary)	Part-of (Scale per aggregate)
Count	NA	Total ≥ Filtered	0→max(filtered value of summary aggregates)	0→filtered value of aggregate. Presented in percentage (0-100%)
Sum (Total)	A numeric attribute	Filtered ≥ Highlighted, Compared	0→max(filtered/compared value of aggregates)	NA (not well defined)
Average		NA		

Table 3. Properties of three aggregate metrics and two measure scale modes.





summarizes the dataset. The dashboards can be authored using drag-and-drop, or using a text configuration format. They can then be shared and embedded on the web.

4.1 The Aggregate Summary Model

A Keshif **summary** extracts an attribute value from all records, and **aggregates** records by that value. Thus, a summary can be considered a *unit of analysis* with automated aggregation of records. Table 1 shows a range of aggregations for various data types. The visual **form** of a summary can have few alternatives to reveal alternative semantics of a data attribute (Table 2). For example, categorical attributes with spatial areas (e.g. countries) can be shown as a list  (to emphasize sorted rankings), or on a map  (to emphasize spatial distributions). Relations in a multi-value categorical attribute can be seen in a set-pair matrix [39], and numeric summaries can include percentile aggregates. Table 2 also shows other features and tasks, such as navigation, search, and sorting, based on data and summary types.

The **aggregate metric** computes a numeric characteristic of the aggregated records. *Count*, the default metric, provides a familiar faceted data overview [34]. *Sum* and *average* metrics use the record values of a chosen numeric attribute. For example, considering a list of companies, the aggregate metric for each city can reveal the count of companies, the total number of employees, or the average growth (Figure 7). In contrast to Tableau [31] and Voyager [37] where *record count* is shown along with numeric record attributes, Keshif distinguishes and clarifies record count as an aggregate metric. Median and percentile characteristics of a record group can be shown using percentile aggregates. Therefore, Keshif achieves high expressiveness by revealing a wide range of data statistics on multiple common data types.

4.2 Linked Aggregate Selection Model

Keshif models three aggregate selection interactions for three complementary tasks. **Highlight**  (mouse-over) allows rapidly previewing characteristics of the records in the selected aggregate. **Filter**  (click) focuses on records within the selected aggregate by removing the records outside of the selection. It is an explicit, permanent selection compared to the highlighting selection for preview. The filtering criteria can be refined incrementally using multiple summaries and selections. **Compare**  (click on ) allows comparing characteristics of multiple record groups side-by-side. It is enabled by locking (capturing and storing) a highlighted selection. This allows explicit comparison of distributions across record groups without the need for memorization. In practice, we limit the number of compared selections to three in order to accommodate for perception. With the total selection representing all the records, Keshif can allow exploration of *six* record group distributions concurrently. All active data selections are shown as breadcrumbs in the header section for a quickly accessible overview of the exploration state.

To support the interaction flow, highlight (mouse-over) selection is designed to precede filtering (click) and comparison (click on lock). If an aggregate design does not reveal an explicit lock icon (e.g. no-value or region glyphs), comparison can be set by shift+click. To define a flexible range selection on an interval summary (to extend beyond fixed aggregate bins), mouse-move with an active modifier key (shift) is used. Lastly, to avoid unintentional triggering of highlighting and visual flickering on mouse-move across the screen, Keshif features a linear and thresholded delay based on the mouse speed. Slower, deliberate mouse moves immediately enable highlighting, while fast moves respond with a short delay.

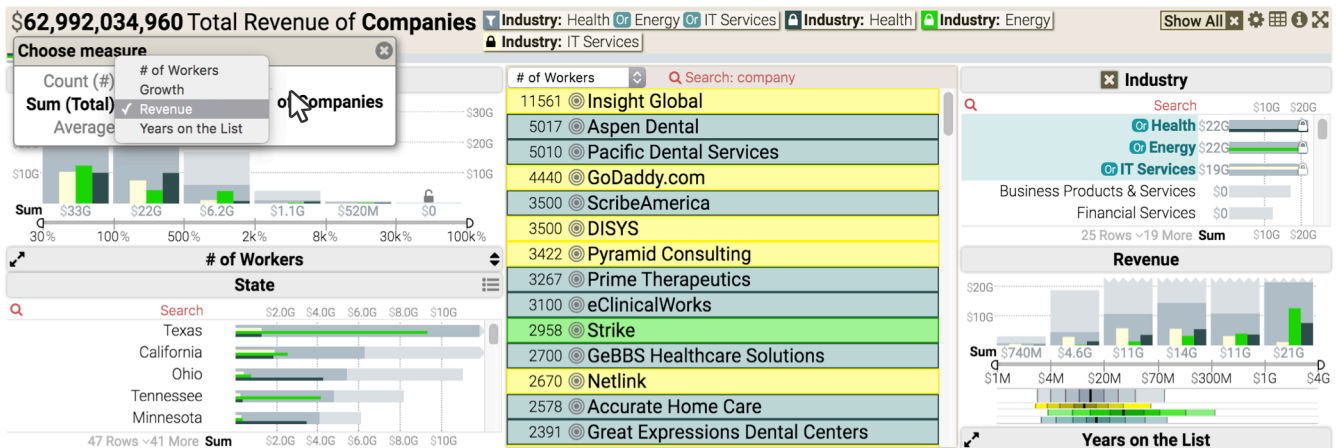



Fig. 3. Keshif dashboard showing a list of companies using three compare-selections. The total revenue is the aggregate metric, selected using the popup panel. The companies are filtered  on health or energy or IT services industries. Each industry is then selected for comparison. Companies that belong into respective industries are automatically color-coded in the middle record display. (Data from www.inc.com)

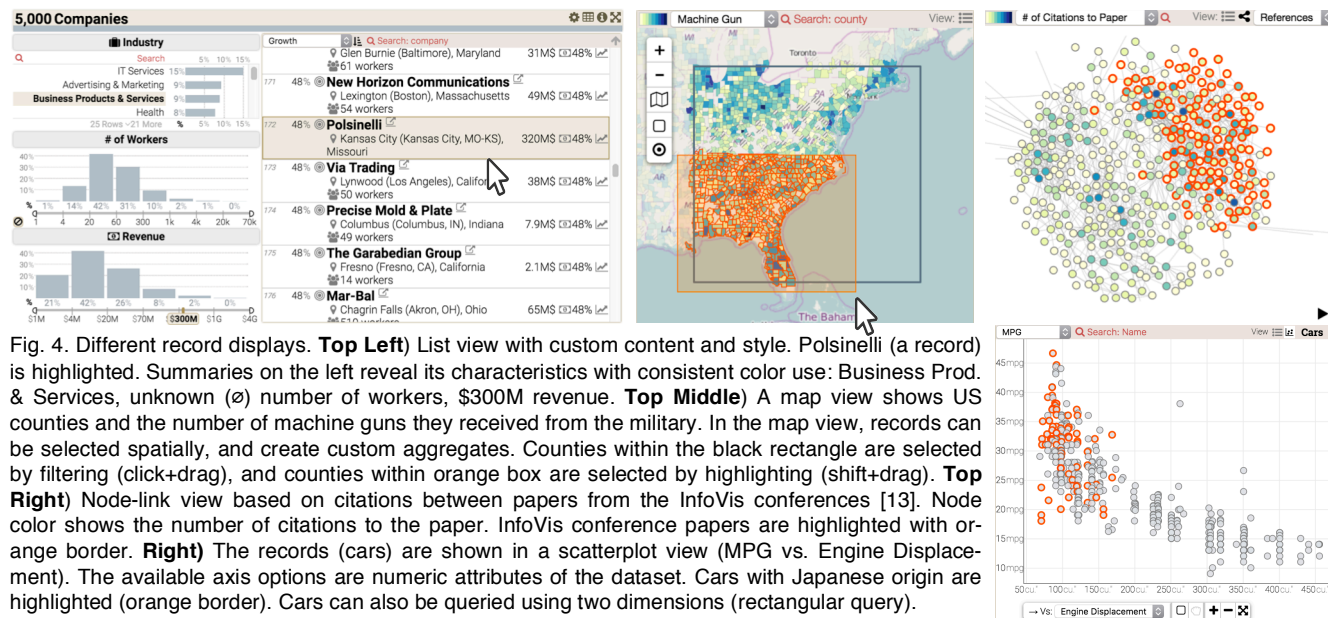


Fig. 4. Different record displays. **Top Left**) List view with custom content and style. Polsinelli (a record) is highlighted. Summaries on the left reveal its characteristics with consistent color use: Business Prod. & Services, unknown (\emptyset) number of workers, \$300M revenue. **Top Middle**) A map view shows US counties and the number of machine guns they received from the military. In the map view, records can be selected spatially, and create custom aggregates. Counties within the black rectangle are selected by filtering (click+drag), and counties within orange box are selected by highlighting (shift+drag). **Top Right**) Node-link view based on citations between papers from the InfoVis conferences [13]. Node color shows the number of citations to the paper. InfoVis conference papers are highlighted with orange border. **Right**) The records (cars) are shown in a scatterplot view (MPG vs. Engine Displacement). The available axis options are numeric attributes of the dataset. Cars with Japanese origin are highlighted (orange border). Cars can also be queried using two dimensions (rectangular query).

4.3 Visual Encodings for Aggregates

Aggregates are visualized by color-coding the selection type (Total , Filtered , Highlighted , Compared), and other visual encodings based on the data type, with two quantitative measure **scale mode** alternatives (Table 1 and 3). *Absolute scale* encodes absolute values, and is shared across all summary aggregates. *Part-of scale* encodes highlighted/compared values as a percentage of filtered records per each aggregate. The scale mode is set by clicking on the chart measure axis, a design that avoids additional control-specific UI. Filtered selection is emphasized by using it to set the summary scale range. Compared selections are shown side-by-side. Highlighted and compared values are within the scale limits when *count* and *sum* metrics are used, as the subset measures less than the filtered set. However, this relation does not hold under *average* metric. In this case, Keshif updates the measure scale to cover compare selection values, but not of highlight selection since frequent scale changes on rapid highlights would be distracting.

Aggregate **measure labels** can be shown in absolute or percentage value under *count* or *sum* metrics. For example, 377 companies in health industry out of 5,000 total companies can be labeled as 8% (Figure 4, Left). Clicking the #-% icons on the chart corner changes the label mode.

Our design is also differentiated by what it excludes and avoids, a process of purposeful elimination of limited or inferior alternatives. Comparisons are shown side-by-side rather than stacked since stacking only works when selected record groups are exclusive, which Keshif does not guarantee, i.e. groups can be compared across summaries or multi-valued categorical data. Side-by-side placement on a shared baseline also accurately reveals each selection and differences across, unlike stacked designs which best reveal totals. We avoided categorical word-clouds because of their limitations in perceptual accuracy, well-defined ordering, and limited support for multiple selections. We avoided parallel coordinates because they of their complexity for data analytics novices.

Record Form	Encoding by numeric attribute	Filtered-Out Records	Highlight / Compare Encoding
List	Sort	Removed	Fill color
Map	Fill color	Transparent (or removed)	Border color
Node-Link	Position (X-Y)		
Scatterplot	Position (X-Y)		

Table 4. The form and visual encoding used by the record display for visualizing records. The default is list view. Map view is enabled if the records define a spatial area. Node-link view is enabled if the records have an attribute that refers to other records. The form can be switched during data exploration.

4.4 The Record Display, Encodings, and Selection

The **record display** shows records individually either as a list, on a map if the records define spatial areas, as a node-link diagram if the records are explicitly interconnected, or as a scatterplot to show relations across two numeric attributes (Figure 4). The visual encodings for records are summarized in Table 4. Mouse-over on a record reveals its attribute values in all summaries (Figure 4, left), an on-demand design that automatically highlights the record within the context of distributions of all records. In the node-link view, mouse-over also highlights connected records. All details of a single record can be seen in a pop-up panel by selecting the record as well.

4.5 Calculated Attributes for Transforming Data

Raw attributes of a dataset may need to be transformed for flexible and effective data representation and analysis. Keshif supports this by **calculated attributes**, functions that return a calculated data feature given a record. Among many uses, calculated attributes can enable \blacklozenge parsing or tokenizing text values (such as splitting "A;B;C" on ";" to generate ["A", "B", "C"]), \blacklozenge processing data without modifying the data source (such as converting values, e.g. "10k" to 10,000 and "20M" to 20,000,000), \blacklozenge extracting time components (Figure 1, *Day of Week* is extracted from *Date*), \blacklozenge combining multiple records features into a multi-valued categorical data type for compact summarization (Figure 5, the services held by the nominees are combined from multiple raw attributes), \blacklozenge

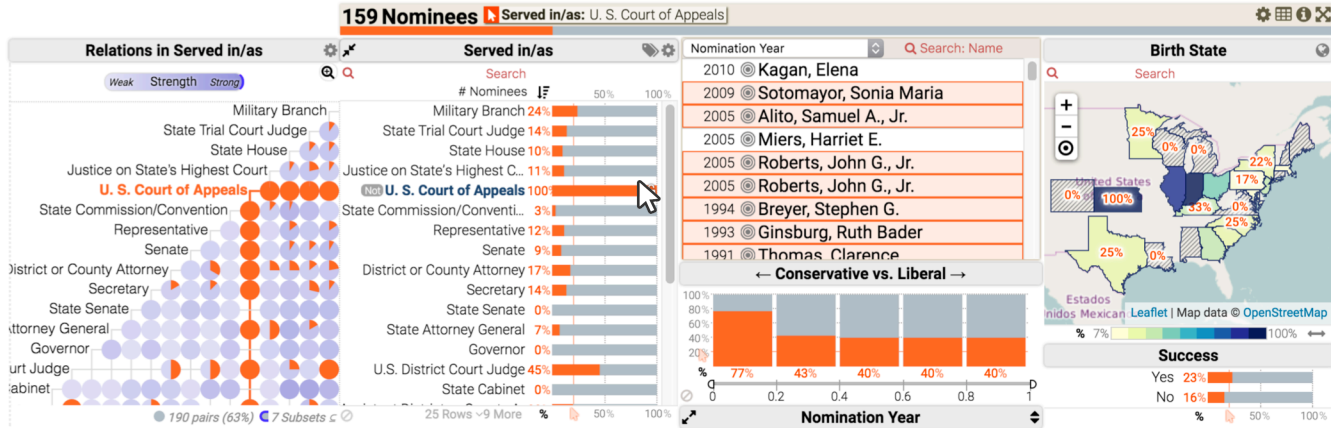


Fig. 5. Keshif dashboard showing the U.S. Supreme Court nominees. Scales show part-of (%) relations. The set matrix [33] shows relations across positions that the nominee served in before the nomination. Nominees that served in U.S. Court of Appeals are highlighted by mouse-over. The map view uses color-coding to show the percentage of nominees served in the selected position among all candidates from that state. Some states do not have a nominee (gaps). Some states (dashed regions) have none (0%) that served in selected position.

merging data from external resources (*Author Countries* summary in a publication dashboard can lookup the country from an author table while iterating on the authors of a single paper), and \blacklozenge defining rich HTML markup for individual records in the record display (Figure 4, Left). Keshif also includes graphical shortcuts for common calculated attributes, such as extracting (i) time components, (ii) the set degree (from a set-typed attribute), and (iii) multiple categories by tokenization.

4.6 Saved Selections for Flexible Queries

With Keshif’s design basis on faceted browsing, filters on different summaries are combined with AND, and temporal / numeric summaries support only a single range filter. To overcome these query limitations, we introduce the **saved selection** feature. At any point, filtered records can be saved and stored as a new category in the saved selections summary. Saved selections can be selected with AND/OR/NOT queries individually (Figure 6), and even analyzed as set relations [39] as saved selections represent a multi-valued categorical grouping of records. This feature can also be used to bookmark result sets.

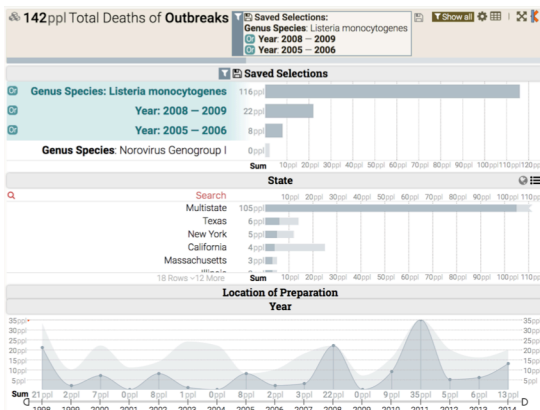


Fig. 6. Keshif dashboard showing foodborne outbreaks in the U.S. Keshif improves upon query expressiveness using saved selections. Top summary shows record groups that were previously saved after filtering. Using saved selections summary, filters across summaries (i.e. year and species) can be combined with OR, and multiple ranges can be selected in a time (year) summary, as shown above.

4.7 Authoring and Sharing Keshif Dashboards

Authoring enables the conversion of raw data to its exploratory form in dashboards, as well as modifying existing dashboards to explore different perspectives. In graphical authoring mode, the available attributes panel (Figure 7) shows the attributes that are not inserted to the dashboard. Each attribute includes a small visualization thumbnail showing its distribution overview. Attributes can be added to, removed from, and moved across four panels by drag-and-drop. To further simplify the placement (a non-exploratory task), double-clicking on an available attribute adds its summary to a panel based on its data type (such as categorical: left, interval: right, time: wide bottom). Calculated attributes can be added by specifying their name and custom function. Our prototype can load datasets from cloud services, local servers, or from the local file system. It can parse CSV and JSON files, and Google Sheets tables automatically. Other types of structured data formats can be parsed through customized data load callback functions.

4.8 Dashboard Configuration Specification

The configuration of a Keshif dashboard expresses *what* is being visualized and explored, not *how*. A configuration sets the data source and describes the summaries and the



Fig. 7. Dashboards are authored using drag-and-drop from available attributes panel to create summaries in four panels (left, right, middle, bottom), or to list records individually. In this view, *US Gross Sales* of movies is dragged and placed between *Creative Type* and *IMDB Rating*. When dragging, the dashboard layout reveals drop zones across all panels, and in-between summaries.

record display. It can be noted in a compact, human-readable textual JavaScript object, or JSON. Keshif detects the attribute data type (categorical, numeric, timestamp) on data load, and automatically visualizes it. In our current implementation, some metadata such as spatial areas of categories or records, setting a referencing attribute to enable node-link view, and specifying ordinal categories (e.g. large > medium > small) need to be defined using the textual configuration. Records can also be individually styled using CSS. We plan to extend metadata detection and graphical authoring capabilities to support these customizations in the future. Our configuration API has been discussed in more detail in an earlier paper [40].

An integration with a third party API (GitHub) enables saving a dashboard configuration on the cloud as a simple repository with a unique ID (anonymously, or with user credentials as public or private). This allows revising and forking for collaboration. The dashboards can be shared with a unique URL, and embedded into web pages using Keshif as a web-service, or directly injected into page DOM with the dashboard configuration.

4.9 Implementation

Keshif is implemented using modern web standards (JavaScript, HTML, CSS) and the D3 library [4]. The Keshif prototype (github.com/adilyalcin/keshif) includes 11kloc of JavaScript, and 4kloc of LESS (CSS pre-processor) style. Keshif maintains doubly-linked indexing between records and aggregates. Filtering and selection states are cached and incrementally updated to reduce computation time. To support datasets with thousands of records, the record list view inserts records on demand using infinite scrolling pattern, reducing number of page elements and improving the rendering speed. The current implementation enables interactive exploration of ~200,000 aggregated records without a computation backend. Scaling up the number of aggregates (e.g. categories), or the number of records in a selection, lowers query time performance.

With its simple configuration, minimal dependencies and cloud data source integrations, Keshif can be integrated, self-hosted, or used as a web service to enable rapid tabular data exploration with minimal technical skills and training. Its implementation allows configuration and customizations using basic web programming.

5 APPLICATIONS

In this section, we describe how data authoring and exploration using Keshif fits in a workflow, and its application to multiple data domains with real-world datasets.

5.1 An Example Use Case

A local newspaper wants to run a story on the homicide victims in their city to inform its readers and policy makers. The journalists track ten years of reported cases, describing the location, motive of the murderer, police investigation status, as well as the name, age, gender, and race of the victim in a spreadsheet. They add the neighborhood of each homicide to reveal spatial trends as a regional overview. They also prepare a GeoJSON file describing the neighborhood areas, indexed by neighbor-

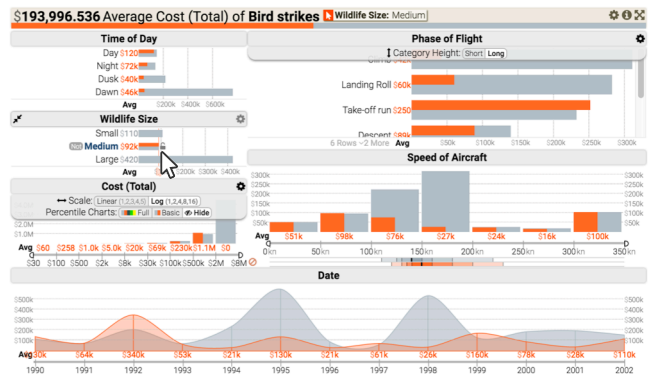


Fig 8. Exploring BirdStrikes dataset. The aggregate measure function is the *average* of *cost*. Medium size birds are highlighted. The highlight selection shows the average damage per each aggregate related to medium-size birds. The average cost is not steady over time. Medium birds cause damage with much less cost at dawn and dusk, compared to average from all types of birds.

hood name. Then, to explore this structured data rapidly, they import it into Keshif.

Keshif first reveals the number of homicide victims (2,294) and the list of attributes with simple distribution previews. Interested in demographics, the journalists add age and sex summaries, which immediately reveal that victim’s ages were commonly between 20-40 (1.4k), and that most victims were male (2.0k). Setting measure labels to percentages calculates that 62% were between ages 20-40, and 89% were male. To analyze if female victims had different age characteristics, they filter to female victims. They notice that only 44% were between ages 20-40, and the distribution change reveals that female victims are older overall. They re-support this observation by clearing the filtering, enabling percentile chart of age distributions, and comparing females and males. Using the 1D percentile distributions, they note that the median age of female was 31, while the median was 26 for male victims. They take a note of these numbers for their news story.

Next, the journalists are interested to see temporal trends in motives. They quickly preview the most common motives by mouse-move: arguments, drugs, retaliation and robbery, and observe their temporal trends. Knowing that the city had been taking measures to reduce drug violence, they highlight drugs again, and find out that over ten year period, the number of drug-related homicides decreased 84% in absolute numbers (49→8). However, they also notice an overall decreasing trend in homicides. Therefore, they lock-select drug related homicides for comparison, and change to *part-of* scale. This reveals that the relative ratio of drug related offenses dropped 21%→8%, a smaller, yet still significant decrease. They note these trends may be due to the new drug policies and policing in the city.

They save the homicide victims dashboard with selected attributes, and share the link with another colleague. When she opens the link, she notices that they might not have analyzed the manner of homicide and its relation to neighborhoods. She first adds the manner summary, and then adds the neighborhood summary and changes to map view to study spatial trends. To explore patterns, she moves the mouse across different manner categories and

observes the changes on the map. She quickly notices homicides with stabbing seem different than the overall distribution: Central city regions have more victims of stabbing. She sends a note to her colleagues, along with a link of the updated dashboard to reproduce the result.

Collecting many insights over the process, along with other resources from public officials, interviews and high-profile cases, the journalists are ready to write their story. They create simple annotated charts with constrained interactions utilizing other tools to highlight the individual trends they observe, and link them in their story. They finally embed the Keshif dashboard in the end of their news story to make data freely explorable. They invite readers to look at their own neighborhood and to find the information valuable to them in the fully interactive dashboard. This sample workflow is based on the example at keshif.me/demo/dc_homicides, and can be reproduced live and online.

5.2 Applications on Multiple Domains

We have created public Keshif dashboards for 160+ datasets (www.keshif.me) across many domains including journalism, surveys, transportation, cities, food, finance, entertainment, politics, and personal data, some of which are exemplified in this paper. The range of datasets demonstrates the generalizability and flexibility of its model and implementation. While importing, studying, and testing many tabular datasets with a wide range of data characteristics, structure and formats, we incrementally refined Keshif’s design, features, and implementation over three years. These sample public data dashboards are created and maintained using the JavaScript-based configurations by the lead author of this paper.

6 EVALUATION

We evaluated Keshif using insight-based evaluation [28] with visual analytics novices in a short-term, casual, open-ended data exploration study with short training. Our goal was to understand insight characteristics and the exploration process, and how these results relate to the proposed model. We aimed to recruit visual analytics novices as they are most impacted by barriers in specifying visual encodings and unconventional visualizations, thus would benefit more from a streamlined exploration flow. Our participants used *only* the graphical interface of Keshif (not the API) to explore the data by authoring (creating and editing) dashboards. Thus, our participants did not use the JavaScript API. Our results are comparable to the evaluation of Voyager [37] at high-level, showing that less-skilled participants could reach insights rapidly using Keshif, comparable to participants in other studies with more skills using more sophisticated tools.

6.1 Study Design

Participants. We recruited 6 participants using public message boards: 4 female, 2 male, 5 aged 18-24 (4 of them students outside computer or information science departments), 1 aged 40-49. Participants were not skilled in visual data exploration, and had not received formal training on visualization. None had used Tableau or simi-

lar visual analytics environments. All had used Excel before. The five younger participants had created charts and analyzed data with Excel, and other tools they had used include SPSS (3), Stata (2), and Graphpad (1), showing their background in statistical analysis. They had not analyzed the studied datasets before, they were not domain experts, and they had not used Keshif before.

Datasets. We used two datasets (movies and birdstrikes – Figure 8) for the study, also used in the evaluation of Voyager [37]. They are chosen for real-world interest to a general audience, of similar complexity and data types. The movies dataset includes 3,201 movie records with 15 attributes (7 categorical, 1 temporal, 8 numeric), including title, director, genre, sales figures, and IMDB / Rotten Tomatoes ratings. The birdstrikes dataset is a redacted version of the FAA wildlife airplane strike database with 10,000 records and 14 attributes (8 categorical, 1 spatial region, 1 temporal, 4 numeric).

Training. The sessions began with a 6-minute video tutorial (1, 2) using a dataset of companies, followed by a warm-up exploration of this dataset for 6 minutes. We also provided 23-page printed slides from the video training. The facilitator answered questions about tool features based on what is covered on the training material.

Study Procedure. We asked participants to explore a dataset, and specifically to “get a comprehensive sense of what the dataset contains and verbally note interesting patterns, trends or other insights”. Their exploration started with the data imported to an empty dashboard. The participants performed an unguided, self-driven exploration without assigned tasks for 15 minutes for each of the two datasets in a think-aloud protocol. Half of the participants explored the movies dataset first, and the others explored the birdstrikes dataset first. After exploring a dataset, participants completed a survey focusing on insight-based metrics. Participants also completed a survey on demographics and data analytics experience.

We did not ask the participants to formulate any questions before the exploration, as doing so might have biased them toward premature fixation on those questions. However, at the end of training, we verbally encouraged them to consider (i) changing the axis mode, (ii) changing the measure function, (iii) using compare selections, and (iv) using the map view (if available) so that they could form richer goals and reach wider insights. In our pilot studies, we observed these features were not utilized by novices in self-driven exploration. We did not enforce these recommendations so the participant remained in full control. Per the think-aloud protocol, the facilitator encouraged communication by asking questions such as “What are you thinking right now?” and “Can you explain in more detail?” when communication stopped or the feedback was vague.

Each study session took about an hour. We compensated participants with \$10 cash. All sessions were held in a university lab using Google Chrome on a Macbook Pro with a 15-inch retina display, and a mouse for interaction. During the studies, we captured the screen and the audio. Surveys results on exploration experience and participant background are also part of our data collection.

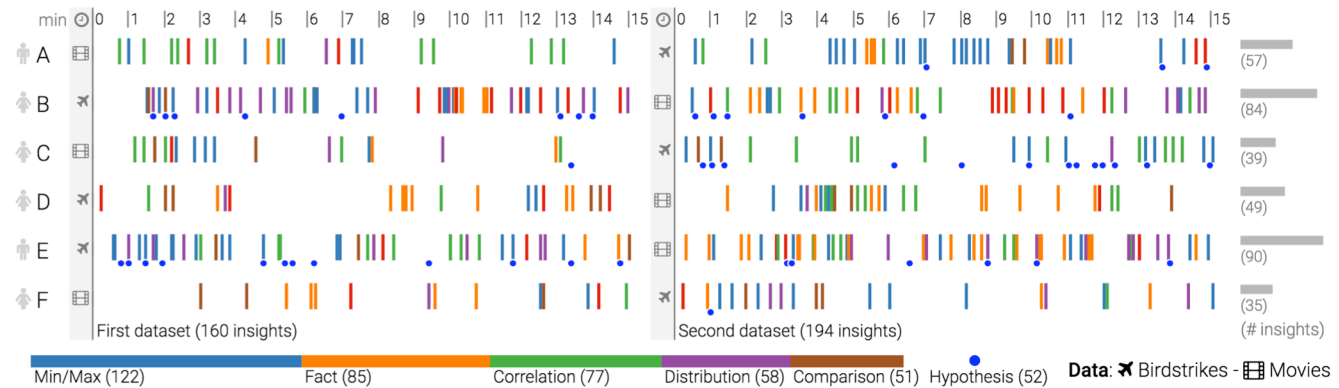


Fig. 9. The timeline of the data insights of the participants in the user study. Each participant explored two datasets for fifteen minutes. Insights are color coded into categories: fact, min/max, distribution, comparison, correlation. Insights with multiple features are colored red. The colored legend shows the total count per each insight category. There are 354 insights and 52 hypotheses noted in the 180 minutes total.

Our evaluation shares Voyager’s study structure [37] in terms of datasets and the open-ended exploration task. However, (i) we recruited visual analytics novices instead of experienced participants, (ii) we limited exploration to 15 minutes per dataset instead of 30 minutes (a more casual use), (iii) we provided shorter training (6 vs. 10 minutes), and (iv) we followed insight-based evaluation with think-aloud protocol instead of using bookmarked charts. We did not compare Keshif and another tool side-by-side because the tools we considered differed substantially in their naturally supported tasks, available charts, and learning time. For example, Voyager does not support interactive linked selections, filtering, and maps. Visualizations in Keshif are not based on an abstract visualization grammar, and do not support the many chart variations of Voyager. Introducing Tableau to a novice in 6 minutes to start rich self-driven data exploration is challenging given its many options and settings, even using ShowMe feature. Keshif does not model data exploration as navigation of recommended charts and relations, but as a data dialogue on automated visualization with few yet expressive modes and interactions. Nevertheless, side-by-side evaluations of task-based performance remain interesting to explore in future work.

6.2 Insight Coding

To detect the insights, the primary author transcribed the verbal feedback of the participants. Using the transcripts, we identified statements that presented an insight on the data content as a single, cohesive proposition. We did not consider statements at a strictly visual level as an insight (such as “there is a peak”), unless participant related it to the data content. We also did not consider restatement of a previous insight as a new insight. Then, we coded attributes of each insight using two passes on the transcripts and the video captures. In the second pass, we extended insight categorizations, and confirmed existing codes. We also noted *hypothesis* statements as a question or an explanation of a trend that can neither be confirmed nor denied given the dataset. A hypothesis commonly relates to participants prior experience and knowledge. The first author consulted the second author to resolve issues. The insight coding results are accessible and explorable as a Keshif dashboard at bit.ly/1Vbs40c.

We coded each insight on its insight-based characteristics and the interface state at the time of insight.

- **Text:** What is the insight?
- **Time:** When was it noted? (seconds elapsed)
- **Correctness:** Was it correct?
- **Feature:** Was it describing a fact, min/max, distribution, comparison or correlation?
- **Data types directly relevant to the insight** (summary type (categorical, numerical, time, map), individual record, etc.)
- **Selection state** (the number of filtered, highlighted, compared summaries)
- **Measure function** (count, sum, or average)
- **Measure label** (absolute or percent-age)
- **Axis mode** (absolute or part-of)
- **Dataset** (movies or birdstrikes)
- **Dataset order** (first or second)
- **Participant ID**

Next, we describe the insight categories we encoded:

- **Fact** describes a property of a record, an aggregation, or a basic observation that does not describe a trend. Examples include “84 of them are causing minor damage”, “That was Delta Airlines”, and “it is an adventure movie”.
- **Min/Max** describes the most/least common feature in the data. Examples include “B737-300 cause the most bird strikes”, “Dramas typically make between 20 and 300M”, and “[Movies were released] Mostly during this time period, between 2004 and 2007.”
- **Distribution** focuses on the variations and trends within a data attribute. Examples include “So, the comedy movie ratings.... it is kinda spread out, they are not that consistent.”, and “It has a large variety of genres, from drama to action, horror.”
- **Comparison** describes two or more specific aggregates, records, or selections. Examples include “[Beloved] has a higher Rotten Tomatoes rating than it does IMDB rating.”, “[After filtering] All of a sudden Dallas falls way down”, and “So the average cost, is, I guess it’s around the same [as the overall trend].”
- **Correlation** describes relations across attributes in a dataset. The relation may be based on a subset of the data. For example, “not many of that (highest grossing) were rated R” relates gross sales with the R rating, describing a

trend. “It looks like they gave pretty good scores to original screen plays” is another example.

The verbal statements may not reveal the details of observations and analytical thinking of the participant in the think-aloud protocol. Overall, we did not expect the participants to articulate the complete exploration state, but to share important aspects of the insight clearly. While encoding insight correctness, we had a permissively positive basis. For example, when the participant noted, “the most strikes are in Pittsburgh region” on a filtered data, we consider it *correct*, even though the filtering criterion is not stated. An incorrect statement example is “Portland has all their hits being the one species of bird”, because Portland has a variety of birds contributing to its birdstrikes. We also encoded some statements as partially correct when the trends could not be easily confirmed, or statements were vague. Examples include “Comedies make that much out of that much money”, and “the worldwide sales (...) definitely move”. Facts on personal experience are not coded for correctness. We assessed the confidence in the insights in post exploration survey.

The coding of the interface state (selections and visual modes) enables understanding how the tool is used and at which stages the insights were obtained/shared. However, the insight may not relate to all such states. For example, when there are multiple compare selections, the insight may describe one distribution rather than a comparison across multiple distributions. Lastly, an insight might relate to multiple data types. For example, “Comedy was one of the top grossing in the US” relates to both genre (categorical) and numeric (gross sales), while describing a min/max feature. We noted type as “map” when the map view was used to describe the location in the insight.

6.3 Analysis and Results

The temporal overview and characteristics of the insights of our study participants is shown in Figure 9. Our participants reached 35 to 90 insights each across the two sessions, with ~2 insights/minute on average. In comparison, Voyager [37] reports 12.5 bookmarked charts in average per 30-minute data exploration session by skilled participants using the same datasets (and 10 charts in average for a drag-and-drop visual specification). Studying the effect of display size across two conditions (targeting very large displays), Reda et al. [24] report about ~1.2 insights/minute. Their participants were mostly computer science graduate students. Liu and Heer’s study [18] on the effect of interaction latency using imMens [19] system with 16 participants skilled in visual analytics (R and Tableau) report a throughput of ~1.9 insights/minute, based on observations or generalizations on two datasets explored 30 minutes each. Our participants had no visual analytics experience, and achieved high insight throughput with little training in shorter time.

During the studies, we noticed that personal differences were a big factor in the variances. To quote the participant with the lowest number of insights (F): “I personally would have gained more from this experience if I was asked to perform specific tasks. (...) I’m not one who necessarily feels inclined to just play on my own. Some people are, some people aren’t.” Therefore,

each individual may not be inclined to reach data insights or perform well when unguided, a challenge in broadening public use of data exploration.

Insights of our participants most commonly described the min/max features in the data attributes (34%). 79% of these insights were on categorical data, suggesting that auto-sorting of categories influenced the exploration outcomes. 24% of the insights included simple facts, 38% of which were on individual records (such as a movie). Correlations were also common (22%), as they also include statements that relate two attributes by first selecting an aggregate on one, and observing the trends in the other. Comparisons were the least common type of insight (14%). Note that an insight may have multiple types. 28% of our coded insights had more than one feature. Our analysis shows the variation in the types of insights communicated by our participants. Arguably, their experience in statistical analysis (through course and personal work, as suggested by tools they had used before) may have guided them to look for and report detailed insights, even though they were not skilled in visual analysis.

The participants most frequently had insights under the default settings that create a familiar faceted interface with absolute record counts and basic distributions. 96% of the insights were made under absolute axis mode, 92% were made with absolute measure label, and 90% were made under the count aggregate measure. Interestingly, the participant with the most insights (E) used the default settings throughout. In contrast, 78% of the insights were reported with some active data selection. Highlighting was active for 34% of the insights, and filtering was active for 55%. However, comparison was less common, only 18% across all reported insights.

Our results show that non-default, less familiar settings for expressive richness are more likely to lead to incorrect statements. Insights made under average or sum measures were incorrect 24% and 20% of the times respectively, compared to only 5% for the default count measure. A substantial difference in accuracy was observed for compare selection as well. 35% of the incorrect or partially correct insights had at least one compare selection at the time the insight was shared, another significant trend in our data. The compare selection on locking interaction is an unfamiliar design compared to filtering and highlighting actions, which may explain the lower accuracy when it was used.



Fig. 10. Results from post-exploration surveys, focusing on the self-evaluation of data exploration experience. Each question includes 12 responses, across six participants on each dataset they explored. The color shows agreement, and the answers are aligned on neutral response, and sorted by mostly-positive agreements first.

The survey results are summarized in Figure 10. Participants collectively agreed they could reach more insights given more time using Keshif. Participants also positively noted they could observe detailed relations and trends, although not comprehensively. The least positive feedback was the perceived value of their insights, and the strongest negative feedback was on familiarity with the domains. This supports that being unfamiliar with a datasets or domain is likely to lower the value of insights. The confidence and value in exploration also reflect our participant’s low confidence in the provided data. Participant C noted, “*I didn’t know where the list came from, how the data was collected (...) I don’t know how much value they have to me, because I don’t know how much I can trust them [dataset].*” Participants responded more positively to their exploration being influenced by what they learn, rather than being targeted. General comments on the usability of Keshif were among the positive feedback.

Our results suggest a learning affect over time with improvement of insight outcomes and satisfaction. More insights were reported in the second session compared to the first (194 vs. 160, an increase of ~20%). Survey results (Figure 10) show that participants were more comfortable in using Keshif in the second session as well (4.3 vs. 5.3 average on 7-point Likert scale answers).

7 LIMITATIONS AND FUTURE WORK

While we have so far presented the motivations and features of Keshif, and its characteristic casual use by novices, we now focus on the limitations of our context and potential future work.

Visualization design. We presented the aggregate glyph designs for visualization on selected common data types rather than a design basis applicable to a wide range of chart and data types. However, our design features and components can be applied to new data types and semantics as appropriate. For example, spatial points can be aggregated on a map using the circular glyphs of set-pair matrix. Our design could be extended to support aggregate hierarchies to represent categorical hierarchies, and merge aggregates for higher-level overviews. However, our goal is not to contribute an absolute and generalized design space, as we believe extensions with new chart types and/or tasks will potentially require re-evaluating and adjusting the proposed strategies.

Data model. An exploratory Keshif dashboard presents a single entity type (tabular data). Additional tables can be referenced in calculated attributes. This data model is basic and consistent, and requires raw, unaggregated data. Our data model and visual summarization and querying strategies can be extended to support more complex data types for record attributes, such as time-series data or rich spatiotemporal datasets.

User and task context. In our design, we targeted users with an intrinsic motivation to understand tabular datasets. While dashboards can be forked, refined, and shared, we do not propose a complete model for synch or asynch collaboration in data exploration. Our model does not present solutions for provenance of insights or inter-

face use. Based on our focus on exploratory process of data understanding rather than data presentation, we did not design Keshif to support adding custom annotations, or exporting charts.

Evaluation. Our evaluation with novices in a casual setting focused on understanding the insights generated from rich raw data and patterns of use with Keshif. Future work could target different well-defined problem spaces and specific features to compare performance and preferences side-by-side across Keshif and other tools. Given our think-aloud protocol, a facilitator was in the room during studies, and encouraged verbal communication throughout. Therefore, our results reflect what the participants chose to share regarding their observations, and may not present learning outcomes comprehensively. In addition, data analysis approach and outcomes differ significantly from one person to another, and our results reflect six first-time users. We believe future studies will generate more observations and understanding of individual differences, and a more thorough understanding of the insights and the process of data exploration.

Expressiveness and design in practice. Results from our user study and our own experiences with other users suggest people are not likely to change default features, and prefer the main faceted browsing design with filtering, highlighting and comparing, even when reminded to explore data in more depth. Influencing behavior from casual to rich, in-depth exploration remains a challenge, especially for novice audiences. This may require more comprehensive training and/or new interface approaches, as well as improving data literacy and critical thinking capacities. We also should note that unconventional design elements, such as rapid mouse-over selections, can be distracting or overwhelming for new, casual users. In Keshif, we aimed to design for a meaningful analytical outcome for every interaction. We also tried to reduce unintentional highlight selections using a delay that responds to pointer speed. However, existing practices have not yet formed norms of intention for such implicit actions. On the other hand, many of our users have expressed the strong utility of, and their preference for, the rapid highlight selection. We believe wider deployment and continuing adoption of similar designs can adjust expectations, and intentions, of a wide audience over time. In the meantime, systems may allow the users to state preferences (opt in/out) for new designs.

Authoring. While Keshif offers a graphical interface for authoring and exploration, features such as calculated attributes, API customizations and custom data loading callbacks target a more technically skilled audience. While informal feedback from external users with novice coding skills noted that Keshif can be learned and used through example dashboard code configurations, we are looking forward to extending graphical features for authoring while maintaining Keshif’s minimalist design.

Form factor. We designed Keshif for desktop/laptop form factors with pointer-based (mouse/touchpad) interaction. Keshif does not aim to scale to small (mobile) displays or large displays effectively, and it is not designed for touch interaction, which could be a future work.

Computation. Keshif is currently implemented to run on a web browser as a client-side tool. While the lack of a server query backend limits computational scalability, it also makes Keshif easy to deploy, maintain, and integrate with existing data sources and web pages. The aggregated visualizations can support larger datasets by design. Future work on server-based computation includes development of remote and scalable data backends, data transmission, and rapid distributed query models.

8 CONCLUSION

In this paper, we presented Keshif, a new automated and systematic design for rapid and expressive tabular data exploration. The search space for visualizations is constrained by aggregate summaries and a record display with design based on data type, and the search space for interaction to reveal detailed data distributions is constrained by multi-modal linked selections. In addition, aggregate measure metrics, visual scale modes, calculated attributes, and saved selections expand data exploration capabilities while maintaining a minimal, consistent, and fully synchronized setting. This approach also enables simple authoring of visual data dashboards from raw data for a streamlined raw data exploration. Our web-based implementation runs in a web browser without a server backend, and can support 200k+ records. We validate our system by (a) presenting samples from 160+ public datasets imported to Keshif across many domains, (b) discussing a sample use case in journalism domain, and (c) reporting on results from an insight-based user study with visual analytics novices under a short-term casual use. The results from our novice participants supports that exploration space created by Keshif can be rapidly learned and interactively used to explore new datasets, with the volume, range and characteristics of insights comparable to skilled users on advanced tools as reported in other studies.

ACKNOWLEDGMENT

We wish to thank Ben Shneiderman, Catherine Plaisant, Stephen North, Fan Du, Deok Gun Park, Karthik Badam, and our anonymous reviewers for their discussions and suggestions. This work was partially supported by the National Socio-Environmental Synthesis Center (SESYNC) through a grant from the U.S. National Science Foundation (# DBI-1052875) to the University of Maryland. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

- [1] C. Ahlberg and B. Shneiderman, "Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 1994, pp. 313–317.
- [2] F. Beck, S. Koch, and D. Weiskopf, "Visual Analysis and Dissemination of Scientific Literature Collections with SurVis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 180–189, Jan. 2016.
- [3] A. Bigelow, S. Drucker, D. Fisher, and M. Meyer, "Reflections on How Designers Design with Data," in *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*, New York, NY, USA, 2014, pp. 17–24.
- [4] M. Bostock, V. Ogievetsky, and J. Heer, "D3: Data-driven documents," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2301–2309, Dec. 2011.
- [5] N. Boukhelifa, J. C. Roberts, and P. J. Rodgers, "A coordination model for exploratory multiview visualization," in *International Conference on Coordinated and Multiple Views in Exploratory Visualization, 2003. Proceedings, 2003*, pp. 76–85.
- [6] M. Brehmer, J. Ng, K. Tate, and T. Munzner, "Matches, Mismatches, and Methods: Multiple-View Workflows for Energy Portfolio Analysis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 449–458, Jan. 2016.
- [7] W. S. Cleveland and R. McGill, "Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods," *Journal of the American Statistical Association*, vol. 79, no. 387, pp. 531–554, Sep. 1984.
- [8] M. Dork, S. Carpendale, C. Collins, and C. Williamson, "Vis-Gets: Coordinated Visualizations for Web-based Information Exploration and Discovery," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1205–1212, Nov. 2008.
- [9] S. van den Elzen and J. J. van Wijk, "Small Multiples, Large Singles: A New Approach for Visual Data Exploration," *Computer Graphics Forum*, vol. 32, no. 3pt2, pp. 191–200, Jun. 2013.
- [10] M. Golemati, C. Halatsis, C. Vassilakis, A. Katifori, and G. Lepouras, "A Context-Based Adaptive Visualization Environment," in *Tenth International Conference on Information Visualization, 2006*, pp. 62–67.
- [11] L. Grammel, M. Tory, and M.-A. Storey, "How Information Visualization Novices Construct Visualizations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 943–952, Nov. 2010.
- [12] S. Greene, E. Tanin, C. Plaisant, B. Shneiderman, L. Olsen, G. Major, and S. Johns, "The end of zero-hit queries: query previews for NASA's Global Change Master Directory," *International Journal on Digital Libraries*, vol. 2, no. 2–3, pp. 79–90, Sep. 1999.
- [13] D. F. Huynh, D. R. Karger, and R. C. Miller, "Exhibit: Lightweight Structured Data Publishing," in *Proceedings of the ACM Conference on World Wide Web*, New York, NY, USA, 2007, pp. 737–746.
- [14] P. Isenberg, F. Heimerl, S. Koch, T. Isenberg, P. Xu, C. Stolper, M. Sedlmair, J. Chen, T. Möller, and J. Stasko, *Visualization Publication Dataset*. 2015.
- [15] A. Key, B. Howe, D. Perry, and C. Aragon, "VizDeck: Self-organizing Dashboards for Visual Analytics," in *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, New York, NY, USA, 2012, pp. 681–684.
- [16] B. C. Kwon, S. H. Kim, S. Lee, J. Choo, J. Huh, and J. S. Yi, "VisOHC: Designing Visual Analytics for Online Health Communities," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 71–80, Jan. 2016.
- [17] B. chul Kwon, B. Fisher, and J. S. Yi, "Visual analytic road-

- blocks for novice investigators,” in *IEEE Conference on Visual Analytics Science and Technology (VAST)*, 2011, pp. 3–11.
- [18] Z. Liu and J. Heer, “The Effects of Interactive Latency on Exploratory Visual Analysis,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 2122–2131, Dec. 2014.
- [19] Z. Liu, B. Jiang, and J. Heer, “imMens: Real-time Visual Querying of Big Data,” in *Proceedings of the 15th Eurographics Conference on Visualization*, Aire-la-Ville, Switzerland, Switzerland, 2013, pp. 421–430.
- [20] Z. Liu, J. Stasko, and T. Sullivan, “SellTrend: Inter-Attribute Visual Analysis of Temporal Transaction Data,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 1025–1032, Nov. 2009.
- [21] J. Mackinlay, P. Hanrahan, and C. Stolte, “Show Me: Automatic Presentation for Visual Analysis,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1137–1144, Nov. 2007.
- [22] K. Madhavan, N. Elmqvist, M. Vorvoreanu, X. Chen, Y. Wong, H. Xian, Z. Dong, and A. Johri, “DIA2: Web-based Cyberinfrastructure for Visual Analysis of Funding Portfolios,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 1823–1832, Dec. 2014.
- [23] C. North and B. Shneiderman, “Snap-together visualization: a user interface for coordinating visualizations via relational schemata,” in *Proceedings of the working conference on Advanced visual interfaces*, New York, NY, USA, 2000, pp. 128–135.
- [24] K. Reda, A. E. Johnson, M. E. Papka, and J. Leigh, “Effects of Display Size and Resolution on User Behavior and Insight Acquisition in Visual Exploration,” in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, New York, NY, USA, 2015, pp. 2759–2768.
- [25] D. Ren, T. Hollerer, and X. Yuan, “iVisDesigner: Expressive Interactive Design of Information Visualizations,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 2092–2101, Dec. 2014.
- [26] J. C. Roberts, “State of the Art: Coordinated Multiple Views in Exploratory Visualization,” in *Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization, 2007. CMV '07, 2007*, pp. 61–71.
- [27] S. F. Roth, J. Kolojechick, J. Mattis, and J. Goldstein, “Interactive Graphic Design Using Automatic Presentation Knowledge,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 1994, pp. 112–117.
- [28] P. Saraiya, C. North, and K. Duca, “An insight-based methodology for evaluating bioinformatics visualizations,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 4, pp. 443–456, Jul. 2005.
- [29] A. Satyanarayan and J. Heer, “Lyra: An Interactive Visualization Design Environment,” *Computer Graphics Forum*, vol. 33, no. 3, pp. 351–360, Jun. 2014.
- [30] B. Shneiderman, “The eyes have it: a task by data type taxonomy for information visualizations,” in *IEEE Symposium on Visual Languages*, 1996, pp. 336–343.
- [31] C. Stolte and P. Hanrahan, “Polaris: A System for Query, Analysis and Visualization of Multi-Dimensional Relational Databases,” in *IEEE Symposium on Information Visualization*, Washington, DC, USA, 2000, p. 5–.
- [32] E. Tufte, *The Visual Display of Quantitative Information*, 2nd edition. Cheshire, Conn: Graphics Pr, 1983.
- [33] L. Tweedie, B. Spence, D. Williams, and R. Bhogal, “The Attribute Explorer,” in *Conference Companion on Human Factors in Computing Systems*, New York, NY, USA, 1994, pp. 435–436.
- [34] M. Vartak, S. Madden, A. Parameswaran, and N. Polyzotis, “SeeDB: Automatically Generating Query Visualizations,” *Proc. VLDB Endow.*, vol. 7, no. 13, pp. 1581–1584, Aug. 2014.
- [35] F. B. Viegas, M. Wattenberg, F. van Ham, J. Kriss, and M. McKeon, “ManyEyes: a Site for Visualization at Internet Scale,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1121–1128, Nov. 2007.
- [36] C. Weaver, “Building Highly-Coordinated Visualizations in Improvise,” in *IEEE Symposium on Information Visualization, 2004. INFOVIS 2004, 2004*, pp. 159–166.
- [37] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer, “Voyager: Exploratory Analysis via Faceted Browsing of Visualization Recommendations,” *IEEE Transactions on Visualization and Computer Graphics*, vol. PP, no. 99, pp. 1–1, 2015.
- [38] M. A. Yalçın, N. Elmqvist, and B. B. Bederson, “Cognitive Stages in Visual Data Exploration,” in *Proceedings of the BELIV Workshop: Beyond Time and Errors - Novel Evaluation Methods for Visualization*, New York, NY, USA, 2016.
- [39] M. A. Yalçın, N. Elmqvist, and B. B. Bederson, “AggreSet: Rich and Scalable Set Exploration using Visualizations of Element Aggregations,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 688–697, Jan. 2016.
- [40] M. A. Yalçın, N. Elmqvist, and B. B. Bederson, “Keshif: Out-of-the-Box Visual and Interactive Data Exploration Environment,” in *Proc. of IEEE VIS 2016 Workshop on Visualization in Practice: Open Source Visualization and Visual Analytics Software*.
- [41] K.-P. Yee, K. Swearingen, K. Li, and M. Hearst, “Faceted metadata for image search and browsing,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2003, pp. 401–408.



Mehmet Adil Yalçın received the Ph.D. degree in 2016 from University of Maryland, College Park, MD, USA, the M.Sc. degree in 2010 from Bilkent University in Ankara, Turkey, and the B.Sc. degree in 2008 from Middle East Technical University in Ankara, Turkey.



Niklas Elmqvist received the Ph.D. degree in 2006 from Chalmers University of Technology in Göteborg, Sweden. He is an associate professor in the College of Information Studies at University of Maryland, College Park, MD, USA. He was previously an assistant professor in the School of Electrical & Computer Engineering at Purdue University in West Lafayette, IN. He is a senior member of the IEEE and the IEEE Computer Society.



Benjamin B. Bederson received the Ph.D. degree in 1992 from New York University, USA. He is a professor in the Computer Science Department at University of Maryland, College Park, MD, USA, and the Associate Provost of Learning Initiatives and Executive Director at Teaching and Learning Transformation Center of the same university. He was elected to the CHI Academy in 2012. He is an ACM Distinguished Scientist.