

Voyager 2: Augmenting Visual Analysis with Partial View Specifications

Kanit Wongsuphasawat*, Zening Qu*, Dominik Moritz*, Riley Chang*, Felix Ouk*, Anushka Anand+, Jock Mackinlay+, Bill Howe*, Jeffrey Heer*

* University of Washington, Seattle, WA + Tableau Research, Seattle, WA
{kanitw, zqu, domoritz, rzc, felixouk, billhowe, jheer}@uw.edu {aanand, jmackinlay}@tableau.com

ABSTRACT

Visual data analysis involves both open-ended and focused exploration. Manual chart specification tools support question answering, but are often tedious for early-stage exploration where systematic data coverage is needed. Visualization recommenders can encourage broad coverage, but irrelevant suggestions may distract users once they commit to specific questions. We present *Voyager 2*, a mixed-initiative system that blends manual and automated chart specification to help analysts engage in both open-ended exploration and targeted question answering. We contribute two partial specification interfaces: *wildcards* let users specify multiple charts in parallel, while *related views* suggest visualizations relevant to the currently specified chart. We present our interface design and applications of the *CompassQL* visualization query language to enable these interfaces. In a controlled study we find that *Voyager 2* leads to increased data field coverage compared to a traditional specification tool, while still allowing analysts to flexibly drill-down and answer specific questions.

ACM Classification Keywords

H.5.2. Information Interfaces and Presentation: UI

Author Keywords

Data visualization; exploratory analysis; partial specification; visualization recommendation; mixed-initiative interfaces.

INTRODUCTION

Exploratory visual analysis is an iterative process that involves both open-ended exploration and focused analysis [10, 11, 30]. Analysts often begin with a broad overview [18], familiarizing themselves with the shape and structure of the data. Next, they can focus on more specific questions. Investigating these questions may spark exploration of potentially relevant factors and outcomes, in turn leading again to more focused analysis. In short, both the breadth and focus of an analyst’s “spotlight of attention” can vary throughout an analysis. However, existing tools provide interaction models designed primarily for either broad exploration or focused question-answering.

Traditional visual analysis tools (*e.g.*, [29, 34, 35]) provide specification interfaces for creating an expressive range of visualizations. Yet providing complete view specifications can be tedious, and require domain familiarity as well as design and analysis expertise. In response, features such as Tableau’s Show Me [15] automatically suggest visual encoding choices for selected data. Nevertheless, analysts must explicitly indicate which subsets of the data to view. If lacking either expertise or discipline, analysts might overlook important aspects of the data or fixate on specific questions prematurely.

Visualization recommender systems (*e.g.*, [19, 32, 39]) can facilitate broad exploration by suggesting data and views for analysts to browse. Our previous *Voyager* [39] system recommends both data fields and visual encodings, steerable via user selections, to promote increased data field coverage in early-stage exploration. However, existing recommender systems provide limited control over suggestions. Once analysts focus on specific questions, these systems can be insufficiently expressive and may distract with irrelevant suggestions.

Rather than treat analysis as a process with dichotomous “modes,” analysts may be better served by tools that support smooth gradations between open-ended and more focused phases of analysis. Towards this goal, we present *Voyager 2*, a new mixed-initiative tool that blends manual and automatic chart specification in a unified system. *Voyager 2* augments traditional visual analysis interfaces with two new *partial view specification* techniques. Using *wildcards*, analysts can precisely vary the properties of a specification to generate multiple charts in parallel, giving them control over sets of views aligned with their analysis goals. *Related views* automatically recommends charts based on the current user-specified *focus view*, promoting discovery of relevant data fields and alternative ways to summarize or encode the data. We also describe how the *CompassQL* visualization query language [38] enables specifications and recommendations in *Voyager 2*.

We evaluate *Voyager 2* in a controlled user study comparing it with PoleStar, a view specification tool modeled on Tableau. Analysis of usage logs and subject ratings finds that *Voyager 2* leads to increased data field coverage and higher ratings for open-exploration tasks. Meanwhile, subjects rate *Voyager 2* comparably with PoleStar for targeted question answering. Comparing these results with a prior comparative study of *Voyager* and PoleStar [39], we find that *Voyager 2* improves upon these prior systems in terms of supporting *both* open-ended and focused analysis.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

CHI’17, May 06–11, 2017, Denver, CO, USA.

Copyright is held by the owner/author(s). Publication rights licensed to ACM.

ACM 978-1-4503-4655-9/17/05 \$15.00

DOI: <http://dx.doi.org/10.1145/3025453.3025768>

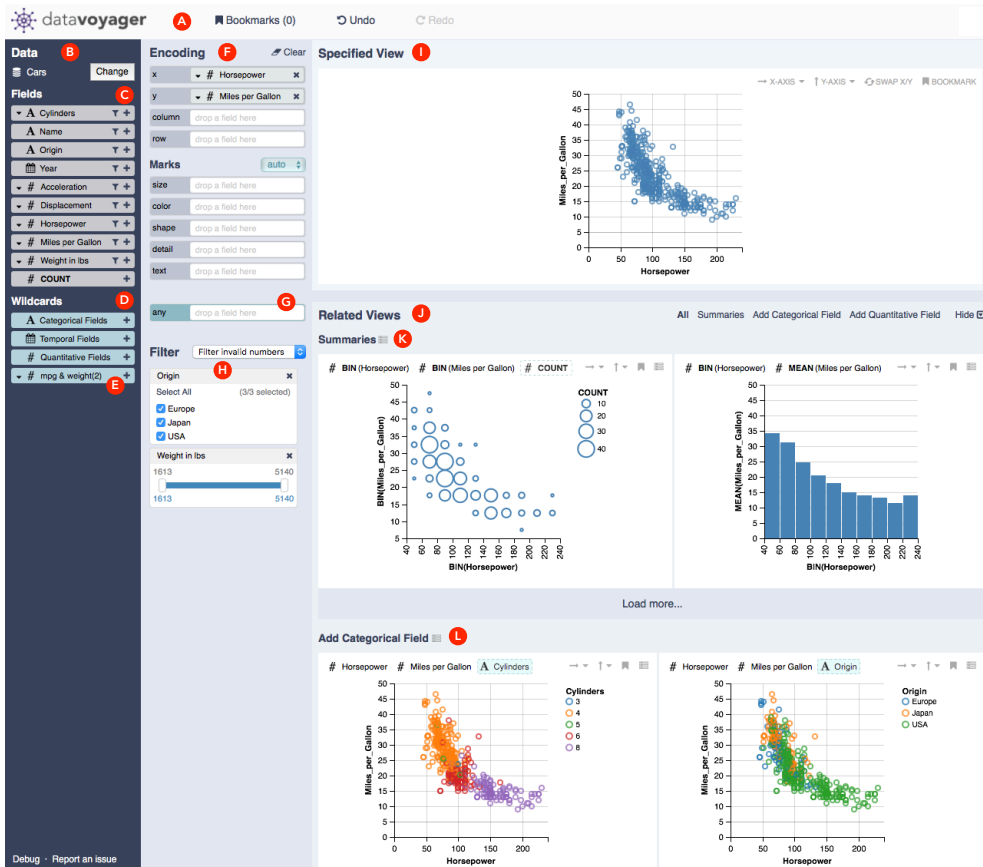


Figure 1. Voyager 2’s interface. The top panel (A) provides bookmark gallery and undo commands. The data panel (B) contains the dataset name, data fields (C), and wildcard fields (D). Wildcard fields let users create multiple views in parallel by serving as “variables” over an enumerated set of fields. Categorical, temporal, and quantitative field wildcards are provided by default, though users can manually author custom wildcards containing desired fields (E). The encoding panel (F) contains shelves for mapping fields to visual channels via drag-and-drop, and a control for selecting mark type. A wildcard shelf (G) lets users add fields without selecting a specific channel, allowing the system to suggest appropriate encodings. The filter panel (H) shows dynamic query controls for filtering. The primary focus view (I) displays the currently specified chart. Related views (J) show recommended plots relevant to the focus view. Related summaries (K) suggest aggregate plots to summarize the data. Field suggestions (L) show the results of encoding one additional field within the focus view.

RELATED WORK

Voyager 2 draws on and extends prior research on exploratory search, visual analysis tools, and visualization recommenders.

Exploratory Search

Exploratory data analysis [10, 30] and exploratory search [16, 33] share a number of characteristics: users might be unfamiliar with their resources (such as datasets), undecided about their goals, or unsure about how to reach their objectives. To perform exploratory tasks, users may either query for specific information or browse to gain an overview and discover the unexpected. As they gain new information, users may clarify their goals and engage in alternative approaches.

Exploratory search tools may employ interfaces such as faceted browsers [40] and dynamic queries [27] to let users focus on interested items. With these interfaces, users express their intent in the form of *partial specifications* that convey criteria for desired items. For large collections, recommender systems [12] can populate a seed set to help users begin exploring or suggest relevant alternatives to selected items.

Our work attempts to support exploratory data analysis in an analogous fashion. In Voyager 2, users can apply wildcards to author partial specifications and query for views that satisfy given constraints. The system also suggests univariate summaries to help analysts begin an exploration, and recommends charts related to the current focus view. All of these interactions are included in one unified tool, enabled by a query language that supports partial specification of visualizations.

Visualization Specification

To facilitate data exploration, visual analysis grammars (e.g., [24, 34, 35]) can succinctly express a variety of charts, in part by letting users omit design details required by lower-level visualization languages (e.g., [5, 25]). To enable *interactive* specification, Tableau (formerly Polaris) [29] offers a graphical interface in which users drag-and-drop data fields onto visual encoding “shelves.” These interactions produce *complete* view specifications using the VizQL visual analysis grammar. Inspired by Tableau, Voyager 2 integrates a graphical specification interface with wildcards and recommendations. Interactions in Voyager 2 produce specifications in CompassQL, a generalization of the Vega-Lite grammar [24] to support *partial* view specifications. Given wildcard, grouping and ranking directives, CompassQL can produce a ranked collection of charts rather than just a single chart. We describe the design of CompassQL and its ability to express a variety of chart recommenders in an earlier paper [38]. In this paper, we illustrate how CompassQL enables a unified system of both complete and partial specifications in Voyager 2.

Visualization Recommendation

To suggest effective visual encodings for presenting data, Mackinlay’s APT [14] introduces a compositional algebra to enumerate the space of visualizations. APT then uses a logic program that codifies a set of *expressiveness* and *effectiveness* criteria inspired by Bertin [3] and Cleveland [7] to prune and rank visualizations. Sage [23] builds on APT with additional chart types and interactions. Tableau provides

“Show Me” features [15] for automatic mark and encoding decisions. Following these systems, Voyager 2 similarly applies expressiveness and effectiveness criteria to recommend encodings. However, Voyager 2 also suggests data fields and transformations, and allows users to specify constraints with *wildcards*. Other systems [6, 8, 41] recommend encodings based on a small taxonomy of predefined tasks. However, task-based approaches can be limited, as users may not have specific tasks in mind during exploratory analysis.

Prior work has also investigated statistical techniques to aid data field selection for a fixed set of encoding templates. The Rank-by-Feature Framework [26] orders histograms and scatter plots based on user-selected metrics. SeeDB [32] calculates deviation scores between data subsets to recommend aggregate views. Quality metrics [4] and scagnostics [1, 36] can also help identify interesting data fields. However, the use of fixed visualization templates limits the utility of these tools for exploratory analysis. In contrast, Voyager 2 recommendations span data fields, transformations, and visual encodings.

Multiple systems present chart galleries with varied data fields and encodings. Some can facilitate early stage exploration, but provide limited user control [19, 37]. Inspired by the Design Galleries project [17], Van den Elzen [31] introduces a system for browsing small multiples of alternative views driven by user-selected parameters. Zenvisage [28] also presents a chart gallery based on queried visual patterns. Voyager 2 similarly generates browseable galleries, but suggests *related views* relevant to the user’s focus without requiring parameter selection. *Wildcards* then enable users to generate multiple views by enumerating selected parameters.

The earlier Voyager system [39] presents a gallery of recommended charts to facilitate breadth-oriented exploration. Backing Voyager is the Compass recommender engine, which accepts user-selected data fields and summary functions to steer the recommendations. In a user study comparing Voyager with PoleStar (a tool based on Tableau), users examined significantly more unique sets of data fields when using Voyager. Subjects favored Voyager for open-ended exploration, but roundly preferred PoleStar for focused analysis, in part due to Voyager’s lack of support for creating arbitrary views.

Voyager 2 is a new mixed-initiative system that blends manual and automatic chart specification in a unified system by extending PoleStar’s interface with *wildcards* and *related views*. This unified system enables pivoting among multiple interaction methods. Users can construct arbitrary views as in Tableau, precisely specify multiple charts in parallel with wildcards, or browse suggested views related to their current focus chart. These interactions are realized using CompassQL, which enables a more expressive range of specifications than the original Compass engine.

USAGE SCENARIO

We start by describing how an analyst might use Voyager 2 to explore data. To aid comparison with the previous Voyager system [39], we use the same dataset of automobile statistics [21]. Upon loading the data, the analyst first wants to familiarize herself with the dataset. She attends to the uni-

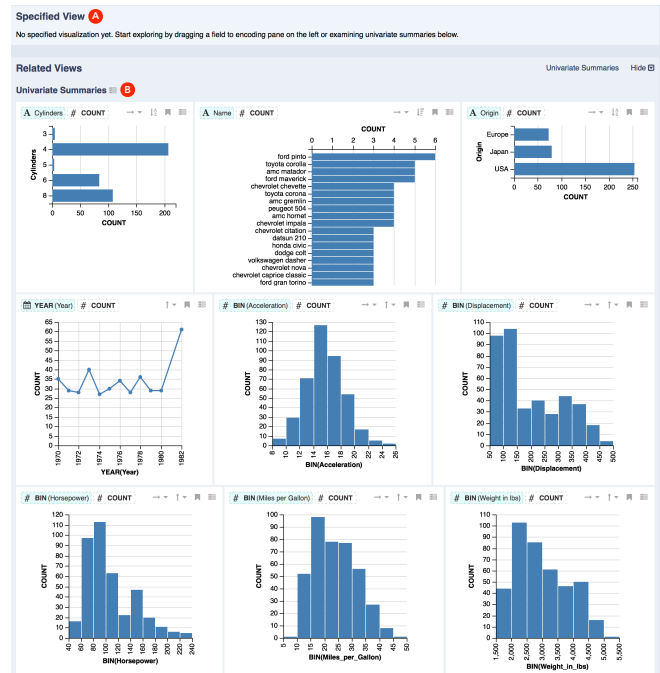


Figure 2. Upon loading a dataset, the *focus view* (A) is empty. The *related views* show univariate summaries (B) for all fields.

variate summaries in the *Related Views* panel (Figure 2). She sees that most, though not all, cars have an even number of cylinders. She also notes multiple records containing the same model name. The majority of cars originate from the USA, though others are from Europe or Japan; all made in the years 1970–1982. The histogram of *acceleration* appears normally distributed, while histograms of other quantitative fields more closely resemble log-normal distributions.

After assessing each univariate summary, the analyst wishes to examine potential associations among quantitative fields. To specify a set of bivariate plots for each pair of fields, she drags two *quantitative field wildcards* (Figure 1-D) to the wildcard “any” shelves (Figure 1-G). In turn, Voyager 2 automatically chooses encodings by mapping each pair of quantitative fields to the x and y axes to produce a gallery of scatter plots (Figure 3). Scrutinizing these plots, she observes a roughly quadratic relationship between *horsepower* and *miles per gallon* (Figure 3-A). As she is interested in investigating this relationship further, she bookmarks the plot and adds a text note documenting her observation. She then clicks the *specify* button (≡) to make this plot the new *focus view* (Figure 1-I).

The first section of *Related Views* now shows summary plots (Figure 1-K), including a 2D histogram of *horsepower* and *miles per gallon*. The second section presents variants of the focus scatter plot with additional color-coded categorical fields (Figure 1-L). The analyst notices that the USA is the *origin* of cars with high *horsepower* and low *miles per gallon*. To dig deeper, she makes this plot the new focus view and examines its related summary plots (Figure 4-A). She confirms that the mean *horsepower* of American cars is higher than other regions, while the mean *miles per gallon* is lower. Below, she also sees a suggested trellis plot partitioned by *origin* as an alternative to color-coding (Figure 4-B).



Figure 3. Dropping two *quantitative field wildcards* onto the *wildcard shelves*. Voyager 2 automatically chooses encodings, producing scatter-plots that show bivariate relationships between all quantitative fields.

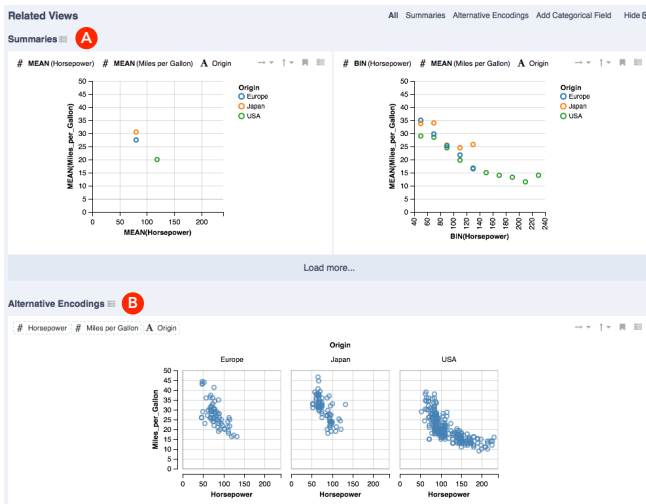


Figure 4. Setting the focus view to the colored scatterplot of *horsepower*, *miles per gallon*, and *origin* in Figure 1-L. The related views panel then displays summaries (A) and alternative encodings (B) of the focus view.

At this point, the analyst wonders how *origin* affects other characteristics of the cars. She clears the encoding shelves and adds *origin* and a *quantitative field wildcard* to the shelves (Figure 5-A). To see both raw distributions and mean summaries, she also applies a *wildcard function* to the quantitative field wildcard (Figure 5-A, 5-C). She examines the resulting gallery of strip plots and aggregate bar plots in Figure 5-D to assess the relationships between *origin* and all quantitative fields at multiple levels of detail.

The analyst continues her analysis, exploring various aspects of the data. As she discovers insightful views, she bookmarks and annotates them with notes so that she can subsequently share her findings with colleagues.



Figure 5. Mapping a *quantitative field wildcard* to *x* and *origin* to *y* (A) produces a gallery of plots. A *wildcard function* enumerates no function (*none*) and *mean* (B-C), generating strip plots of raw values and bar charts of mean values (D). The ? in (A) denotes the wildcard function.

DESIGN CONSIDERATIONS

The original Voyager paper [39] proposed a set of considerations to guide the design of recommendations for a faceted chart browser. Here, we revisit those considerations (C1-C6) and propose extensions (C7, C8) in the context of a unified tool that blends manual and automatic chart specifications.

C1. Show data variation, not design variation. *Recommendations should prioritize views of different fields and transformations over different encodings of the same data.* Related views and wildcards in Voyager 2 continue to follow this approach, encouraging broader data exploration [18]. However, Voyager 2 also provides precise control over visual encodings.

C2. Allow interactive steering to drive recommendations. *The system must provide controls for users to indicate their intent.* Voyager lets users select data fields and summary functions of interest, but does not permit encoding specification. Voyager 2 blends recommenders into a manual specification tool, granting more control to analysts. Users can create arbitrary encodings as in Tableau and browse *related views* based on the current focus, or query a set of views with *wildcards*.

C3. Use expressive and effective visual encodings. *Recommendations should apply perceptual design principles [3, 7, 14].* Voyager 2 continues to apply *expressiveness* criteria to exclude misleading encodings and uses *effectiveness* metrics to rank suggestions as a part of the underlying CompassQL language.

C4. Promote reading of multiple charts in context. *Present related charts such that effort spent reading one chart can aid reading of the next.* Akin to Voyager, Voyager 2 aligns charts and makes their axis ranges consistent to ease comparison when possible. Voyager 2 also applies this consideration for suggesting summary views or views with additional fields by preserving the visual encodings of the focus chart. It also clusters suggestions into groups to provide local consistency.

C5. Prefer fine-tuning to exhaustive enumeration. *Rather than show all possible charts, include simple interactions to view chart variants.* Voyager 2 preserves lightweight interactions for sorting, scale transforms, and axis transposition of recommended charts. In addition, its unified specification interface enables fine-tuning of visual encoding mappings.

C6. Enable revisitation and follow-up analysis. *Provide bookmarking and export features to enable sharing and recall.* Voyager 2 provides similar support for undo, bookmarks and chart export. Bookmarks now include support for text notes.

For Voyager 2, we further extend these considerations:

C7. Use automation to extend user focus. *Ground suggestions in the current context of analysis.* While Voyager presents browseable recommendations, Voyager 2 instead augments manual specification. *Related views* elaborate on the user's focus view, while *wildcards* allow precise control over chart exploration. To transition from browsing to follow-on analysis, users can make any suggested or enumerated views the new focus, or interact with the shelves to modify the view.

C8. Avoid redundant suggestions. *Presenting many similar recommendations may overwhelm or distract users.* In addition to promoting data variation over design variation, Voyager 2 groups suggestions into selectable categories, limits the default number of suggestions per category, and prunes the space of visual encodings to suggest distinct designs.

THE VOYAGER 2 INTERFACE DESIGN

We now present the interface design of Voyager 2, which allows users to pivot among manual specification, *wildcard* specification, and browsing *related views* in a unified system.

Basic Interactions for Manual Chart Specification

Figure 1 shows the Voyager 2 interface. The top panel provides buttons for undo, redo, and opening a bookmark gallery (C6). The left column contains the *data* panel, which lists data fields and wildcards, ordered by data type and then by name. A *count* field is provided to aggregate the number of records.

The middle column contains the *encoding* and *filter* panels. The *encoding* panel (Figure 1-F) provides controls for specifying visual encodings and data field transformations (C2). To visualize data, analysts can drag-and-drop a data field onto an encoding channel shelf (e.g., *x*, *y*, *color*). By default, the system encodes raw (unaggregated) values. Via a drop-down menu (Figure 5-B), analysts can select a transformation such as an aggregation, binning, or time unit function.

We also provide automatic encoding features akin to Tableau's *Add to Sheet* and *Automatic Mark* [15]. Analysts can double-click a field or click its *add to shelf* (+) button to add it to an automatically-selected encoding channel (C3). By default, the system automatically picks an appropriate mark type; analysts can override this choice using a drop-down menu. To enable dynamic query filters [27] as in Figure 1-H, a user can drag a field onto the *filter* panel or click the filter button (▼).

As encodings are modified, the *focus view* (Figure 1-I) and *related views* (Figure 1-J) panels in the rightmost column update accordingly. Each view contains toolbar buttons for modifying axes, sorting, and bookmarking (C5, C6). Users can hover over a chart element to reveal a tooltip describing the underlying data point. For large views that require scrolling, users can activate a local scroll bar by hovering for 500ms (to disentangle local and global scrolling), akin to Voyager [39].

Specifying Views in Parallel with Wildcards

Wildcards let analysts specify multiple charts in parallel by authoring partial specifications (C2, C7). In response, Voyager 2 presents a *specified gallery*, showing charts that satisfy the wildcard constraints, as in Figures 3, 5, and 6. The specified gallery and other wildcard interfaces use teal background to distinguish them from other items. To avoid overwhelming users (C8), the system does not show *related views* when wildcards are in play.

Each plot in the specified gallery includes *preview capsules* in its top-left corner. The capsules display the visualized data fields and transformation functions, which are indicated using bold capitalized text (e.g., **MEAN**). Hovering over a capsule triggers an *encoding preview*, in which the shelves transiently update to show the specification of the selected plot and highlight corresponding fields. Encoding previews can help analysts understand the visual encodings used and help novice users learn how to construct particular types of charts.

Akin to a single focus view, the top-right corner for each plot in the specified gallery contains buttons for chart modification and bookmarking. In addition, the *specify* button (☰, intended to look like encoding shelves), allows users to assign a plot as the new focus view (C7). Like preview capsules, the *specify* button triggers an encoding preview upon mouse hover.

To aid comparison across charts (C4), Voyager 2 uses the global minimum and maximum values of a data field as its axis range by default even for aggregated plots, making axis ranges for the same data field consistent across plots. Users can disable this behavior using the toolbar.

As view specification primarily involves selection of data fields, transformations and visual encodings, we provide the following kinds of wildcards to enable partial specification:

Wildcard fields let analysts construct views that treat data fields as free variables. Voyager 2 provides preset wildcard fields for *categorical*, *temporal*, and *quantitative fields* (Figure 1-D). These wildcards represent all fields of a particular data type. Analysts can simply drag-and-drop a wildcard field onto an encoding shelf to create multiple charts in parallel. For example, Figure 3 uses wildcard fields to produce a gallery of scatter plots involving all pairs of quantitative fields.

To create a wildcard for a specific set of fields, analysts can author a *custom wildcard field* by dragging desired fields to the wildcard list. Figure 1-E shows a custom wildcard field for *miles per gallon* and *weight in lbs*. To avoid incongruous views, Voyager 2 prevents analysts from creating custom wildcard fields containing multiple data types.

Wildcard functions allow simultaneous specification of data field transformations. Checking the wildcard checkbox in an encoding shelf's dropdown menu (Figure 5-B) enables a wildcard function. Once enabled, the radio buttons for each function become checkboxes, allowing analysts to select which functions to include (Figure 5-C). For example, selecting *none* and *mean* produces charts with raw data and mean summaries, as shown in Figure 5-D. The encoding shelf capsule (Figure 5-A) denotes the use of wildcard functions with "?".

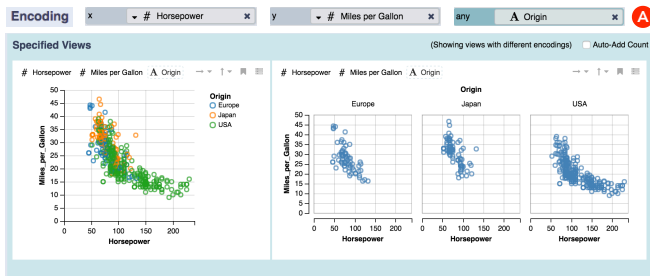


Figure 6. Wildcard shelf (A) allows users to consider alternative ways to encode *origin* (by using color or by faceting).

Wildcard shelves enable ambiguous assignment of data fields to encoding channels. Analysts can drag a field or wildcard field onto a wildcard shelf (Figure 1-G). In Figure 6, an analyst uses a wildcard shelf to add the *origin* field to a scatterplot of *horsepower* and *miles per gallon*. To avoid redundant views (C8), Voyager 2 only produces distinctly different encodings. For example, the output gallery only shows a column-based trellis plot and excludes a row-based trellis plot.

Applying multiple wildcards at the same time varies multiple parameters. For example, Figure 5 applies both a wildcard field and a wildcard function to see both distributions and mean values of each *quantitative field* conditioned on *origin* values. However, applying too many wildcards may produce a large number of views. Similar to Voyager, Voyager 2 prioritizes showing data variation over design variation (C1). When analysts use wildcard shelves together with wildcard fields or functions, Voyager 2 selects only the top-ranked encoding channel for the fields on the wildcard shelves.

Related Views to Promote Data Coverage

To promote data coverage and surface alternative ways to summarize and encode the data (C7), Voyager 2 presents multiple groups of *related views* based on the current focus view. Each group header (Figure 1-K, 1-L) denotes the type of suggestion. To avoid overwhelming analysts with only one kind of suggestion (C8), Voyager 2 displays a limited number of views in each section and provides a “show more” button (Figure 4-A, bottom) for expanding the list. Users can also focus on a specific suggestion type via tab navigation. Each related view has interfaces similar to a view in the specified gallery. Voyager 2 supports the following types of related views:

Univariate summaries (Figure 2) show distributions for all data fields when no focus view is specified. Rather than starting with a blank screen, univariate summaries help users familiarize themselves with the different data fields (C1), following suggested practices for exploratory analysis [18]. To maintain consistency with the *data* panel (C4), these plots are ordered by data type and then by field name.

Summaries present aggregate plots to augment a focus view showing raw data. For example, Figure 1-K presents summary plots for the scatterplot between *horsepower* and *miles per gallon* in Figure 1-I. For quantitative fields, we calculate *mean* values or *bin* the fields to create histograms. To help users see distributions of discrete fields, we automatically add *count* to the summaries if the focus view has no quantitative fields. To determine top summary plots, Voyager 2 prefers views

with fewer transformations to ease interpretation, and then sorts enumerated functions in the same order as the function drop-down dialog (Figure 5-B) to facilitate chart reading (C4).

Field suggestions show plots that contain one additional field. The goal is to help analysts consider other relationships that they might otherwise overlook (C1). To produce these views, Voyager 2 adds the suggested fields to the most effective channel still available (C3). To facilitate interpretation (C4), Voyager 2 keeps existing encodings of the focus view constant and groups views that add fields of the same data type together. For instance, Figure 1-L shows a group of recommended views that use the *color* channel to encode suggested categorical fields in the scatterplot between *horsepower* and *miles per gallon* (Figure 1-I). For consistency, the system orders the views by field name (C4). To avoid overplotting (C3), Voyager 2 only provides field suggestions if there is an empty position shelf (*x* or *y*) available or if none of the non-positional channels (*color*, *size*, or *shape*) have been assigned.

Alternative encodings display other options for visualizing the same data (data fields and transformations identical to the focus view) when there are multiple effective encodings. For example, Figure 4 shows that analysts can encode *origin* using a partitioned trellis plot in addition to using a *color* encoding.

Refining Specifications of Related Views with Wildcards

Any section presented under *related views* can also be expressed using a *wildcard* encoding. To support refinement of suggestions (C7), each related view section header contains a specify button (☰). The button generates an *encoding preview* upon hover. Upon clicking the button, the encoding shelves are assigned a wildcard-based specification that generates the selected subset of related views. Thus, the specify button provides a mechanism for conducting focused analysis with a *collection* of recommended views, and may also help analysts learn how to use and interpret wildcards.

COMPASSQL: ENABLING PARTIAL SPECIFICATION

Both recommendations and specifications in Voyager 2 use the CompassQL visualization query language [38] as an underlying representation. In this section, we first briefly describe the design of CompassQL. We then illustrate how interactions in Voyager 2 produce CompassQL queries.

The CompassQL Visualization Query Language

A CompassQL query (Figures 7-8) consists of (1) a partial chart specification and enumeration constraints, and (2) directives for grouping, choosing, and ranking suggestions. Given a query as input, the CompassQL engine first uses a backtracking algorithm [22] to enumerate complete view specifications that satisfy all constraints. The engine then clusters and ranks candidates using the provided directives to produce results.

Partial Specification. A specification in CompassQL (*spec*) has a similar structure to a Vega-Lite unit specification [24], but allows replacing concrete values with enumeration specifiers (or “wildcards”), indicating that certain properties should be determined by the query engine. For instance, we can replicate Tableau’s *Automatic Mark* [15] by setting the mark type of a specification to a wildcard (?) as in Figure 7-A.

<p>A Focus View (Figure 1-I)</p> <pre> { "spec": { "data": {"url": "cars.json"}, "mark": "?", "encodings": [{ "channel": "x", "field": "Horsepower", "type": "quantitative" }, { "channel": "y", "field": "Miles_per_Gallon", "type": "quantitative" }] }, "groupBy": "similarEncodings", "chooseBy": "effectiveness", "orderBy": "effectiveness" } </pre>	<p>B Wildcard Specification (Figure 3)</p> <pre> { "spec": { "data": {"url": "cars.json"}, "mark": "?", "encodings": [{ "channel": "?", "field": "?", "type": "quantitative" }, { "channel": "?", "field": "?", "type": "quantitative" }] }, "groupBy": "transformedFields", "chooseBy": "effectiveness", "orderBy": "fieldOrder" } </pre>
---	---

Figure 7. CompassQL queries: focus view and wildcard specification.

CompassQL wildcards can include user-defined constraints. For example, a *custom* wildcard field constrains the underlying CompassQL field property to a particular set. Besides user-defined constraints, CompassQL implicitly applies a set of *expressiveness* constraints (listed in supplemental material) to suggest sensible and non-misleading views (C3). Derived from Voyager’s Compass engine [39], these constraints concern both perceptual expressiveness [14] and the expressiveness of transformation functions. For example, CompassQL excludes charts that use *shape* to encode a quantitative field (as *shape* does not convey magnitude), and avoids inapplicable functions (e.g., by not applying *time units* to non-temporal fields).

Grouping. For a given set of constraints, there may be many charts with the same data or similar encodings. To reduce redundancy (C8), a query may include a *groupBy* property to define a key function for grouping similar charts. A key function for Voyager 2 may determine sets of identical fields, sets of identical fields and transformations (*transformedFields*), or sets of *similarEncodings*. For similar encodings, transposed charts and assignments of a data field to a non-positional channel (e.g., *color*, *size*, *shape*) are each clustered together.

Choosing and Ordering. The *chooseBy* property defines a scoring function for selecting a top representative (or “exemplar”) of each cluster group, while *orderBy* specifies a scoring function for ordering per-group exemplars in the recommendation list. The *fieldOrder* function sorts enumerated fields using an order consistent with the *data* panel (C4). To ease interpretation of enumerated transformation functions (C4), *functionOrder* prefers views with fewer transformations and then sorts enumerated functions in the same order as the function drop-down dialog (Figure 5-B). To suggest perceptually effective encodings (C3), Voyager 2 uses *effectiveness* scores derived from Compass [39].

Generating Queries for View Specifications

Similar to PoleStar [39] and Tableau [29], Voyager 2 maintains a one-to-one mapping between the interface and the underlying specification. For example, Figure 7-A shows a query of the focus view in Figure 1-I. Wildcards in Voyager 2 also map directly to wildcards in CompassQL queries. For instance, using wildcard shelves and fields in Figure 3 produces a CompassQL query in which the corresponding encoding channels and fields are wildcards (Figure 7-B).

To prioritize data variation (C1), Voyager 2 groups views with identical *transformedFields* if a specification has a wildcard field or function (Figure 7-B). However, if there is no wildcard field or function, Voyager 2 shows design variations and groups views with *similarEncodings* (Figure 7-A).

For each group of similar views, Voyager 2 chooses an exemplar with the top perceptual effectiveness scores (C3). It then orders these exemplars to produce query results. To facilitate chart reading (C4), Voyager 2 first ranks them by *fieldOrder* if there is a wildcard field (Figure 7-B), and ranks them using the *functionOrder* scores if there is a wildcard function. It then ranks the exemplars based on their perceptual effectiveness (C3), as in Figure 7-A.

Generating Queries for Recommending Related Views

To provide related views, Voyager 2 uses the following methods to generate CompassQL queries from a focus chart.

To produce *univariate summaries* (Figure 2) when the focus view is empty, we use a fixed query template that includes a count field and a *wildcard field* with a *wildcard function* (Figure 8-A). To make each output view contain a group-by field for aggregating count, an expressiveness constraint implicitly limits the wildcard function to *bin* for a quantitative field, to time unit functions for a temporal field, or to *none* for a nominal field. Both fields are mapped to wildcard channels of a wildcard mark, letting the system pick appropriate encodings and orientation. Akin to Voyager, Voyager 2 prefers the y-axis for encoding nominal fields, resulting in horizontal labels that are easier to read. To avoid redundancy (C8), Voyager 2 clusters univariate summaries with identical fields and chooses an exemplar with the top effectiveness score for each field (C3). It then sorts the exemplars by *fieldOrder* (C4).

To generate *summaries* for an unaggregated focus view as in Figure 1-K, we add a wildcard function (*fn*), constrained to mean or bin, to each quantitative field (Figure 8-B). For each temporal field, we add a wildcard function with varying time units. The mark type, left as a wildcard, is automatically determined. The *autoAddCount* flag automatically aggregates the count of records for plots with only discrete fields. Voyager 2 groups summary views with identical sets of *transformedFields* (C1), chooses exemplars based on effectiveness (C3), and sorts views with enumerated functions using the *functionOrder* scores (C4).

To provide *field suggestions* (Figure 1-L), we augment the focus view’s specification with a wildcard field constrained to a fixed data type (Figure 8-C). To provide an appropriate level of detail for the added field, we apply a wildcard to constrain *fn* to *none*, *bin* or *mean* for quantitative fields, and to varying time units for temporal fields. Voyager 2 groups views with identical fields (C8), chooses exemplars with the highest effectiveness (C3), and sorts them by *fieldOrder* (C4).

To suggest *alternative encodings* (Figure 4-B), we replace the encoding channels and mark type with wildcards (Figure 8-D). To show distinctly different designs (C8), Voyager 2 clusters *similarEncodings* and excludes charts similar to the focus view from the query results. Per-cluster exemplars are chosen and sorted based on their perceptual effectiveness (C3).

A Univariate Summaries (Figure 2)

```
{
  "spec": {
    "data": {"url": "cars.json"},
    "mark": "?",
    "encodings": [{
      "channel": "?",
      "fn": "?",
      "field": "?",
      "type": "?"
    }],
    {
      "channel": "?",
      "fn": "count",
      "field": "*",
      "type": "quantitative"
    }
  ],
  "groupBy": "fields",
  "chooseBy": "effectiveness",
  "orderBy": "fieldOrder"
}
```

B Summaries (Figure 1-K)

```
{
  "spec": {
    "data": {"url": "cars.json"},
    "mark": "?",
    "encodings": [{
      "channel": "x",
      "fn": ["bin", "mean"],
      "field": "Horsepower",
      "type": "quantitative"
    }],
    {
      "channel": "y",
      "fn": ["bin", "mean"],
      "field": "Miles_per_Gallon",
      "type": "quantitative"
    }
  ],
  "groupBy": "transformedFields",
  "chooseBy": "effectiveness",
  "orderBy": "functionOrder",
  "config": {"autoAddCount": true}
}
```

C Field Suggestions (Figure 1-L)

```
{
  "spec": {
    "data": {"url": "cars.json"},
    "mark": "?",
    "encodings": [{
      "channel": "x",
      "field": "Horsepower",
      "type": "quantitative"
    }],
    {
      "channel": "y",
      "field": "Miles_per_Gallon",
      "type": "quantitative"
    }],
    {
      "channel": "?",
      "field": "?",
      "type": "nominal"
    }
  ],
  "groupBy": "fields",
  "chooseBy": "effectiveness",
  "orderBy": "fieldOrder"
}
```

D Alternative Encodings (Figure 4-B)

```
{
  "spec": {
    "data": {"url": "cars.json"},
    "mark": "?",
    "encodings": [{
      "channel": "?",
      "field": "Horsepower",
      "type": "quantitative"
    }],
    {
      "channel": "?",
      "field": "Miles_per_Gallon",
      "type": "quantitative"
    }],
    {
      "channel": "?",
      "field": "Origin",
      "type": "nominal"
    }
  ],
  "groupBy": "similarEncodings",
  "chooseBy": "effectiveness",
  "orderBy": "effectiveness"
}
```

Figure 8. CompassQL queries for different categories of related views.

EVALUATION: VOYAGER 2 VERSUS POLESTAR

We conducted a user study to assess Voyager 2’s ability to support both breadth- and depth-oriented analysis. We compared Voyager 2 with *PoleStar* [39], a specification interface modeled on Tableau [29], a state-of-the-art tool for visual analysis. Our study design isolates *wildcards* and *related views* as the only difference between conditions. To this aim, we extended PoleStar to support filtering as well as *automatic mark* and *add to shelf* features [15]. To facilitate cross-study comparison between Voyager and Voyager 2, our design mirrors a previous study [39] comparing Voyager and PoleStar. We hypothesized that Voyager 2 would lead to higher data field coverage than PoleStar, and that, unlike Voyager, Voyager 2 would enable analysts to flexibly drill-down and answer specific questions.

Study Design

Our study employed a 2 (interface) × 2 (dataset) mixed design. Each subject conducted two exploratory analysis sessions, each with a different, counterbalanced tool and dataset.

Datasets. To facilitate cross-study comparison, we reused the datasets from a prior study [39]: statistics about motion pictures (“movies”) and a redacted version of FAA wildlife airplane strike records (“birdstrikes”). The movies dataset contains 3,201 records and 15 fields (7 nominal, 1 temporal, 8 quantitative). The birdstrikes dataset has 10,000 records and 14 fields (8 nominal, 1 geographic, 1 temporal, 4 quantitative).

Participants. We recruited 16 participants (11 female, 5 male), including 11 graduate students, 2 researchers, and 3 software engineers. All subjects had prior data analysis experience: all had used Excel, 9 had used Tableau or PowerBI, 13 had used Python/matplotlib, and 11 had used R/ggplot. All subjects had neither analyzed the study datasets before, nor had they used Voyager 2 or PoleStar. However, some subjects found basic interactions in both tools familiar due to their experience with Tableau. Each subject spent approximately 2 hours in our study. They received a \$15 gift certificate as compensation.

Study Protocol. Before each session, we provided a 15 minute tutorial of the tool using the automobile dataset [21]. We then briefly introduced subjects to the experimental dataset. We

asked participants to “comprehensively explore the data.” We also asked them to bookmark insightful views and add text notes describing their rationale. To encourage accountability, subjects were told they would need to verbally summarize their insights at the end of each session, using their bookmarked views. We used a think-aloud protocol, asking participants to verbalize their thought process during the session. We did not ask participants to formulate specific questions before the session, as we did not want to bias them toward premature fixation on a specific set of questions. Subjects had 30 minutes to explore the dataset in each session, but could choose to end the session early once satisfied with their exploration.

We held all sessions in a laboratory setting. Participants ran both tools in Google Chrome on a Macbook Pro with a 15-inch retina display (2,880 × 1,980 pixels). After completing the analysis sessions, participants completed an exit survey.

Collected Data. An experimenter observed each session and took notes. We recorded audio and the screen for later review. Both visualization tools recorded interaction logs, including all input and application events. Finally, the exit survey included Likert scale ratings and subjects’ rationales for their ratings.

Analysis of Usage Logs

We use linear mixed-effects models [2] to analyze usage log data. We include visualization tool and session order as fixed effects, and include intercept terms for dataset and subject as random effects (representing per-dataset and per-subject bias). Following common practice, to assess significance we use likelihood-ratio tests that compare a full model to a reduced model in which the fixed effect in question has been removed.

Voyager 2 promotes increased data field coverage. As in the previous Voyager study [39], we want to assess the breadth of exploration. We consider the number and percentage of unique field sets both shown and interacted with. Following prior work [9, 13, 39], we use mouse hover and interaction with a chart’s toolbar as proxies to assess if a user has examined a chart. This approximation provides a conservative estimate, as users may scrutinize charts without direct interaction.

In Figure 9, the visualization tool used significantly affects the number of unique field sets shown ($\chi^2(1) = 43.380, p < 0.001$) and interacted with ($\chi^2(1) = 26.999, p < 0.001$). On average, subjects were exposed to 102 unique field sets in Voyager 2 (over a baseline of 23 for PoleStar) and interacted with 41 unique field sets (over a baseline of 17). Comparing across studies, in prior work Voyager led to 3.2x more field sets seen and 1.5x more field sets interacted with on average. Here, the numbers for Voyager 2 are 4.4x and 2.4x: evidence that Voyager 2 similarly supports breadth-oriented analysis. While the prior study found an effect due to tool presentation order, we do not find such an effect in this study.

For the percentage of unique fields, Voyager 2 users were on average exposed to 98% of all fields in a dataset, over a baseline of 80% for PoleStar ($\chi^2(1) = 17.476, p < 0.001$), and interacted with 93% of fields versus a baseline of 79% ($\chi^2(1) = 10.644, p < 0.001$). In other words, subjects overlooked 20% of the fields in a dataset on average when using PoleStar, while only 7% were overlooked using Voyager 2. Moreover, 12/16 (75%) subjects interacted with at least 90% of all fields in Voyager 2, while only 5/16 (31%) did in PoleStar.

Bookmark rate is unaffected by visualization tool. Analyzing the count of bookmarked views, we find no effect due to tool ($\chi^2(1) = 0.381, p = 0.537$). Echoing prior work [39], we find that users bookmark views at similar rates, despite increased exposure from recommendations. We do not find an effect due to presentation order ($\chi^2(1) = 1.675, p = 0.196$).

Plots in related views are the most interacted with and bookmarked. From the total of 233 minutes that subjects interacting with charts in Voyager 2, they spent 109 minutes (46.8%) with related views, 41 minutes (17.7%) with views created with wildcards, and 83 minutes (35.5%) with views created without wildcards. Drilling down, 8/16 (50%) subjects spent the majority of their time interacting with related views. Time spent interacting with either related views or wildcard views accounts for the majority of time for 14/16 (87.5%) subjects.

To assess if these interactions led to notable discoveries, we analyzed the source of bookmarked views. For 144 total charts bookmarked in Voyager 2, 62 (43.1%) are from related views, 32 (22.2%) are from views created with wildcards, and 50 (34.7%) are from views created without wildcards. Bookmarks from related views or from views created using wildcards account for the majority of bookmarks for 11/16 (68.8%) subjects. This result suggests that related views and wildcards contributed to finding interesting views.

Analysis of User Ratings

In the exit survey we asked subjects to reflect on their experiences and to provide ratings for both tools and features.

Voyager 2 excels at exploration and supports focused analysis. Subjects rated the tools they found more valuable for exploration and question answering, using a symmetric 7-point scale (Figure 10). For exploration, 15 subjects find Voyager 2 “more” or “much more valuable”, while only one finds PoleStar “somewhat more valuable” ($\mu = 2.19, \sigma = 0.981, t(15) = 8.919, p < 0.001$). For question answering, no participant has a

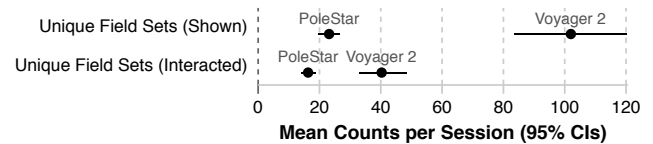
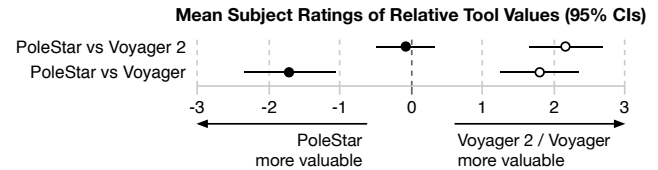


Figure 9. Mean counts and 95% CIs of unique field sets shown and interacted with. Users view and interact with more fields using Voyager 2.



- **Focused Question Answering:** Users rated Voyager 2 comparably to PoleStar but roundly preferred PoleStar to Voyager in the prior study.
- **Open-ended Exploration:** Users roundly preferred Voyager 2 to PoleStar, akin to the prior study in which users roundly favored Voyager over PoleStar.

Figure 10. Mean subject ratings and 95% CIs of relative tool value for question answering and open-ended exploration (symmetric 7-point scale) from our study and [39]. Voyager 2 has higher overall ratings than Voyager and PoleStar in terms of supporting both analysis phases.

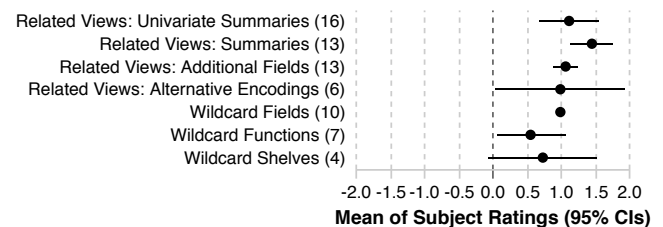


Figure 11. Mean usefulness ratings and 95% CIs for Voyager 2 features on a symmetric 5-point scale (-2 not useful, +2 very useful) show that subjects find both related views and wildcards features useful. Labels include the number of subjects who used and rated each feature.

strong preference: 7 subjects are neutral while 5 and 4, respectively, find PoleStar or Voyager 2 “somewhat more valuable” (-1, +1) ($\mu = -0.063, \sigma = 0.772, t(15) = -0.324, p = 0.751$). In prior work [39], subjects preferred Voyager for exploration, but strongly favored PoleStar for question answering. Compared to Voyager and PoleStar, Voyager 2 has higher overall ratings for supporting both analysis phases.

When asked how comprehensive they believed their analysis to be, subjects reported similar confidence levels for both tools (Voyager 2: $\mu = 0.375, \sigma = 1.258$; PoleStar: $\mu = 0.375, \sigma = 1.628; W = 135.5, p = 0.786$). Although Voyager 2 has more features, user ratings indicate comparable ease of use (Voyager 2: $\mu = 1.313, \sigma = 1.352$; PoleStar: $\mu = 1.188, \sigma = 1.223; W = 117, p = 0.680$).

Participants use wildcards less, but find both related views and wildcards helpful when used. Participants also rated the usefulness of various Voyager 2 features using a symmetric 5-point scale. Participants could give negative ratings if they found a feature “distracting”, or indicate “did not use” if they did not use a certain feature in the session. As shown in Figure 11, subjects found the Voyager 2 features useful. Though used less frequently, wildcards were well received. Subjects with more analysis expertise particularly appreciated the ability to explore multiple fields in a controlled fashion.

Participant Feedback: Balancing Automation & Control

In their free text comments, subjects described how Voyager 2 (especially *related views*) aided exploration and learning:

"I feel more confident using Voyager [2]. It helped me to learn. PoleStar feels scarier and like using the older SPSS tools. Voyager [2] seems more learner friendly."

"Voyager 2 does all the summary charts for you, whereas I spent most of my time building similar summary charts in PoleStar."

"The related view suggestion function in Voyager [2] accelerates exploration a lot."

"I liked that Voyager [2] showed me what fields to include in order to see a specific graph. With PoleStar, I had to do a lot of trial and error and couldn't express what I wanted to see."

Some participants also raised concerns for how recommendations might shape an analyst's thought process:

"These related views are so good but it's also spoiling that I start thinking less. I'm not sure if that's really a good thing."

Subjects also commented on the use of *wildcards*:

"I found wildcards useful when I wanted a quick view to compare all the categorical or quantitative variables."

"Creating my own wildcard field made it easier for me to compare the output graphs side by side."

"I wasn't sure in advance sometimes if the graph I wanted to see should use Bin or Median/Mean, so I used the Wildcard Function to compare, pick, and learn for future use."

"The wildcard seems to offer the same functionality as some of the related views but would give me more control over what I wanted to see. I think this would be really useful if I knew the tool well and had been using it for several weeks."

One participant with a strong statistics background also noted that, though she made heavy use of related views, with more experience she would prefer the control afforded by wildcards.

Summary

Together, our results indicate that Voyager 2's partial specification interfaces facilitate broader exploration, tool learning, and enable both serendipitous and controlled discovery. They also suggest the need for more study of these interfaces. Wildcards were viewed as powerful aids for analysis that might become increasingly useful with additional training. Related views were praised for accelerating analysis and suggesting otherwise overlooked directions, but also raise questions about the degree to which analysts might rely on them. Does the ready presence of relevant suggestions erode an analyst's independent thought process, and if so, to what ends?

CONCLUSION & FUTURE WORK

We contribute Voyager 2, a visual analysis tool that combines manual and automatic chart specification in a single unified system. We introduce two partial specification interfaces: *wildcards* let users precisely vary the properties of a specification to generate multiple charts in parallel, while *related views*

recommends visualizations relevant to the user's current focus. Both specifications and recommendations in Voyager 2 are represented using CompassQL [38], a visualization query language based on Vega-Lite [24]. Our controlled study evaluates this unified system approach. Comparing our results to a prior study [39], we find that Voyager 2 improves upon both a prior traditional specification interface (PoleStar) and a chart recommendation browser (Voyager) in terms of supporting *both* open-ended exploration and focused question answering.

There remains important future work for balancing manual and automatic chart specification in visual analysis tools. One important research topic is the design and evaluation of more sophisticated and scalable visualization recommenders. In terms of expressivity, we plan to further extend CompassQL to support additional summary plots. For example, box plots and layered charts can provide more statistical information, including variability and trend lines, in addition to the central tendency of a distribution. Moreover, as the number of data fields increases, sorting suggestions according to statistical measures (e.g., prevalence of outliers, strength of association with the focus view, partitioning quality [1]) may be preferable to sorting by data type and field name. While CompassQL supports extension via user-defined functions [38], the selection of appropriate statistical measures, end-user validation, and efficient, scalable computation remain as challenges.

We also plan to extend Voyager 2's interface. We intend to add wildcard interfaces for other properties such as binning parameters and scale types. While Voyager 2's design takes reading of multiple charts in context into account (C4), more work is needed to formalize and evaluate this goal. For example, one might improve visualization design criteria to consider consistency between a set of visualizations presented together [20]. Moreover, Voyager 2 is primarily designed to work on a single desktop or laptop monitor. Professional analysts might utilize larger displays and benefit from more optimized layouts for displaying specification results and recommendations.

Finally, this paper focuses on a unified system blending manual and automatic chart specification, and evaluates the system as a whole. We leave detailed studies of isolated aspects of the system, such as wildcards, as important future work. Longitudinal studies might help us better understand how analysts learn and apply partial specification interfaces in their own work. By recording and modeling longer-term analysis activities, we might characterize analytic strategies and gain more insight into the benefits and potential drawbacks of introducing increased automation into the data exploration process.

To support future research, Voyager 2 and CompassQL are available as open-source software at <https://vega.github.io>.

ACKNOWLEDGMENTS

We thank both the anonymous reviewers and the UW Interactive Data Lab for their helpful comments. This work was supported by the Intel Big Data ISTC, DARPA XDATA, the Moore Foundation, and the UW eScience Institute.

REFERENCES

1. Anushka Anand and Justin Talbot. 2016. Automatic Selection of Partitioning Variables for Small Multiple

- Displays. *IEEE Transactions on Visualization and Computer Graphics (Proc. InfoVis)* 22, 1 (2016), 669–677.
2. Dale J. Barr, Roger Levy, Christoph Scheepers, and Harry J. Tily. 2013. Random effects structure for confirmatory hypothesis testing: Keep it maximal. *Journal of memory and language* 68, 3 (2013), 255–278.
 3. Jacques Bertin. 1983. *Semiology of graphics: diagrams, networks, maps*. University of Wisconsin press.
 4. Enrico Bertini, Andrada Tatu, and Daniel Keim. 2011. Quality metrics in high-dimensional data visualization: an overview and systematization. *IEEE Transactions on Visualization and Computer Graphics (Proc. InfoVis)* 17, 12 (2011), 2203–2212.
 5. Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. 2011. D3: Data-Driven Documents. *IEEE Transactions on Visualization and Computer Graphics (Proc. InfoVis)* 17, 12 (2011), 2301–2309.
 6. Stephen M. Casner. 1991. Task-analytic approach to the automated design of graphic presentations. *ACM Transactions on Graphics (TOG)* 10, 2 (1991), 111–151.
 7. William S. Cleveland and Robert McGill. 1984. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *J. Amer. Statist. Assoc.* 79, 387 (1984), 531–554.
 8. David Gotz and Zhen Wen. 2009. Behavior-driven visualization recommendation. In *Proceedings of the 14th international conference on Intelligent user interfaces*. 315–324.
 9. Spence Green, Jeffrey Heer, and Christopher D. Manning. 2013. The efficacy of human post-editing for language translation. In *Proc. ACM Human Factors in Computing Systems (CHI)*.
 10. Jeffrey Heer and Ben Shneiderman. 2012. Interactive Dynamics for Visual Analysis. *Commun. ACM* 55, 4 (April 2012), 45–54. <https://idl.cs.washington.edu/papers/interactive-dynamics>.
 11. Jeffrey Heer, Frank Van Ham, Sheelagh Carpendale, Chris Weaver, and Petra Isenberg. 2008. Creation and collaboration: Engaging new audiences for information visualization. In *Information Visualization*. Springer, 92–133.
 12. Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. 2004. Evaluating Collaborative Filtering Recommender Systems. *ACM Trans. Inf. Syst.* 22, 1 (Jan. 2004), 5–53.
 13. Jeff Huang, Ryen White, and Georg Buscher. 2012. User see, user point: gaze and cursor alignment in web search. In *Proc. ACM Human Factors in Computing Systems (CHI)*.
 14. Jock Mackinlay. 1986. Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics* 5, 2 (1986), 110–141.
 15. Jock Mackinlay, Pat Hanrahan, and Chris Stolte. 2007. Show Me: Automatic Presentation for Visual Analysis. *IEEE Transactions on Visualization and Computer Graphics (Proc. InfoVis)* 13, 6 (2007), 1137–1144.
 16. Gary Marchionini. 2006. Exploratory search: from finding to understanding. *Commun. ACM* 49, 4 (2006), 41–46.
 17. Joe Marks, Brad Andalman, Paul A. Beardsley, William Freeman, Sarah Gibson, Jessica Hodgins, Thomas Kang, Brian Mirtich, Hanspeter Pfister, Wheeler Ruml, and others. 1997. Design galleries: A general approach to setting parameters for computer graphics and animation. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 389–400.
 18. David S. Moore and George P. McCabe. 1989. *Introduction to the Practice of Statistics*. WH Freeman/Times Books/Henry Holt & Co.
 19. Daniel B. Perry, Bill Howe, Alicia M.F. Key, and Cecilia Aragon. 2013. VizDeck: Streamlining exploratory visual analytics of scientific data. In *Proc. iSchool Conference*.
 20. Zening Qu and Jessica Hullman. 2016. Evaluating Visualization Sets: Trade-offs Between Local Effectiveness and Global Consistency. In *Proceedings of the Beyond Time and Errors on Novel Evaluation Methods for Visualization*. ACM, 44–52.
 21. Ernesto Ramos and David Donoho. 1983. ASA Data Exposition Dataset. (1983). <http://stat-computing.org/dataexpo/1983.html>.
 22. Francesca Rossi, Peter Van Beek, and Toby Walsh. 2006. *Handbook of constraint programming*. Elsevier.
 23. Steven F. Roth, John Kolojejchick, Joe Mattis, and Jade Goldstein. 1994. Interactive graphic design using automatic presentation knowledge. In *Proc. ACM Human Factors in Computing Systems (CHI)*. ACM, 112–117.
 24. Arvind Satyanarayan, Dominik Moritz, Kanit Wongsuphasawat, and Jeffrey Heer. 2017. Vega-Lite: A Grammar of Interactive Graphics. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)* (2017). <https://idl.cs.washington.edu/papers/vega-lite>.
 25. Arvind Satyanarayan, Ryan Russell, Jane Hoffswell, and Jeffrey Heer. 2016. Reactive Vega: A Streaming Dataflow Architecture for Declarative Interactive Visualization. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)* (2016). <https://idl.cs.washington.edu/papers/reactive-vega-architecture>.
 26. Jinwook Seo and Ben Shneiderman. 2005. A rank-by-feature framework for interactive exploration of multidimensional data. *Information Visualization* 4, 2 (2005), 96–113.
 27. Ben Shneiderman. 1994. Dynamic queries for visual information seeking. *Software, IEEE* 11, 6 (1994), 70–77.

28. Tarique Siddiqui, Albert Kim, John Lee, Karrie Karahalios, and Aditya Parameswaran. 2017. Effortless Data Exploration with zenvisage: An Expressive and Interactive Visual Analytics System. *International Conference on Very Large Data Bases (VLDB)* (2017).
29. Chris Stolte, Diane Tang, and Pat Hanrahan. 2002. Polaris: A System for Query, Analysis, and Visualization of Multidimensional Relational Databases. *IEEE Transactions on Visualization and Computer Graphics* 8, 1 (2002), 52–65.
30. John W. Tukey. 1977. Exploratory data analysis. *Reading, Ma* 231 (1977), 32.
31. Stef van den Elzen and Jarke J. van Wijk. 2013. Small multiples, large singles: A new approach for visual data exploration. *Computer Graphics Forum* 32, 3pt2 (2013), 191–200.
32. Manasi Vartak, Samuel Madden, Aditya Parameswaran, and Neoklis Polyzotis. 2014. SeeDB: Automatically Generating Query Visualizations. *Proceedings of the VLDB Endowment* 7, 13 (2014), 1581–1584.
33. Ryen W. White and Resa A. Roth. 2009. Exploratory search: Beyond the query-response paradigm. *Synthesis Lectures on Information Concepts, Retrieval, and Services* 1, 1 (2009), 1–98.
34. Hadley Wickham. 2009. *ggplot2: Elegant Graphics for Data Analysis*. Springer.
35. Leland Wilkinson. 2005. *The Grammar of Graphics*. Springer.
36. Leland Wilkinson, Anushka Anand, and Robert L. Grossman. 2005. Graph-Theoretic Scagnostics.. In *IEEE Transactions on Visualization and Computer Graphics (Proc. InfoVis)*, Vol. 5. 21.
37. Graham Wills and Leland Wilkinson. 2010. Autovis: automatic visualization. *Information Visualization* 9, 1 (2010), 47–69.
38. Kanit Wongsuphasawat, Dominik Moritz, Anushka Anand, Jock Mackinlay, Bill Howe, and Jeffrey Heer. 2016a. Towards a general-purpose query language for visualization recommendation. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*. ACM. <https://idl.cs.washington.edu/papers/compassql>.
39. Kanit Wongsuphasawat, Dominik Moritz, Anushka Anand, Jock Mackinlay, Bill Howe, and Jeffrey Heer. 2016b. Voyager: Exploratory Analysis via Faceted Browsing of Visualization Recommendations. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)* (2016). <https://idl.cs.washington.edu/papers/voyager>.
40. Ka-Ping Yee, Kirsten Swearingen, Kevin Li, and Marti Hearst. 2003. Faceted Metadata for Image Search and Browsing. In *Proc. ACM Human Factors in Computing Systems (CHI)*. 401–408.
41. Michelle X. Zhou and Min Chen. 2003. Automated generation of graphic sketches by example. In *IJCAI*, Vol. 3. 65–71.