

Interactive Dynamics for Visual Analysis

Jeffrey Heer Stanford University, jheer@cs.stanford.edu

Ben Shneiderman University of Maryland, College Park, ben@cs.umd.edu

ABSTRACT

Visualization is regularly touted for its capacity to boost human visual performance, enabling users to more effectively perceive patterns in data. This capacity helps users to more rapidly carry out tasks such as assessing data quality, gaining awareness of data, and hypothesis formation. While most discussions of visualization focus on the data display, this article emphasizes visualization's second design component: interactive user controls to iteratively pose questions and then review, share and discuss their results. We propose a taxonomy of interactive dynamics for successful visual analytics systems. This 12-component taxonomy has three high-level categories: (1) *data & view specification*, (2) *view manipulation*, and (3) *analysis process & provenance*. Our taxonomy covers issues such as visualization creation, query widgets, direct manipulation, multi-view coordination, analysis history and collaboration. The taxonomy is intended to assist reasoning about interactive analysis tools by designers, researchers, professional analysts, procurement officers, educators and students. Validating and refining this framework is a community project that proceeds by feedback, critique and refinement.

1. INTRODUCTION

The increasing scale and availability of digital data provides an extraordinary resource for informing public policy, scientific discovery, business strategy and even our personal lives. But for such data to be most valuable, users must be able to make sense of it: to pursue questions, uncover patterns of interest, and identify (and potentially correct) errors. In concert with data management systems and statistical algorithms, analysis requires contextualized human judgments regarding the meaning and significance of the clusters, trends, and outliers discovered in data.

Visualization provides a powerful means of facilitating sense-making. By mapping data attributes to visual properties such as position, size, shape and color, visualization designers leverage perceptual skills to help users discern and interpret patterns within data [\cite{card:readings}](#). However, a single image typically provides answers to at best a handful of questions. Meaningful analysis consists of a *dialogue*: a conversation involving both people and data. Confusing widgets, hidden operations, incomprehensible displays or slow response times can limit the range and depth of topics considered, and may curtail thorough deliberation and introduce errors. To be most effective, visual analytics tools must support the fluent and flexible use of visualizations at rates resonant with the pace of human thought.

This article facilitates research, development, and procurement of visual analytics tools by considering how they promote analytic dialogues. We present a taxonomy of interactive dynamics that contribute to successful visual analytics systems. Our goal is to assist designers, researchers, professional analysts, procurement officers, educators and students in evaluating

and creating analysis tools. The taxonomy consists of twelve task types grouped into three high-level categories: (1) *data & view specification* (visualize, filter, sort & derive), (2) *view manipulation* (select, navigate, coordinate & organize), and (3) *analysis process & provenance* (record, annotate, share & guide). These categories incorporate the critical tasks that enable iterative visual analysis, including visualization creation, interactive querying, multi-view coordination, history and collaboration. Validating and refining this framework is a community project that proceeds by feedback, critique and refinement.

<i>Data & View Specification</i>	<p>Visualize data by choosing visual encodings.</p> <p>Filter out data to focus on relevant items.</p> <p>Sort elements to expose patterns.</p> <p>Derive values or models from source data.</p>
<i>View Manipulation</i>	<p>Select elements to highlight, filter or manipulate them.</p> <p>Navigate to examine high-level patterns and low-level detail.</p> <p>Coordinate views for linked, multi-dimensional exploration.</p> <p>Organize multiple windows and workspaces.</p>
<i>Process & Provenance</i>	<p>Record analysis histories for revisitation, review and sharing.</p> <p>Annotate patterns to document findings.</p> <p>Share views and annotations to enable collaboration.</p> <p>Guide novices through analysis tasks.</p>

Table {tab-taxonomy}: A Taxonomy of Interactive Dynamics for Visual Analysis.

Our focus on interactive elements presumes a basic familiarity with visualization design. The merits and frailties of bar charts, scatterplots, timelines and node-link diagrams, and of the visual encoding decisions that underlie such graphics, are certainly a central concern. However, we will largely pass over them here. A number of other articles and books address these topics in great detail \cite{card:readings, cleveland:elements, few:now, tuft:visual} and we recommend them to the interested reader.

Also, a brief note on terminology: throughout the article we adopt the term “analyst” to refer to someone making use of visual analysis tools. We do not mean to imply a specific person or role. Our notion of analyst encompasses anyone seeking to understand data: traditional analysts investigating financial markets or terrorist networks, scientists uncovering new insights about their data, journalists piecing together a story, and ordinary people tracking their money spent, electricity used, or blood pressure.

2. DATA & VIEW SPECIFICATION

To enable analysts to explore large data sets involving varied data types (e.g., multivariate, geospatial, textual, temporal, networked), flexible visual analysis tools must provide appropriate controls for specifying the data and views of interest. These controls enable users to selectively **visualize** the data, to **filter** out unrelated information to focus on relevant items, and to **sort** information to expose patterns. Analysts often need to **derive** new values from the input data, remove outliers and anomalies, or apply external metadata to group the data meaningfully.

2.1 Visualize

Perhaps the most fundamental operation in visual analysis is to specify a visualization of data: users must indicate which data are to be shown and how they should be depicted. Historically, this process required custom programming of a specific visualization component. Within user interfaces, such visualization “widgets” are often presented in a *chart typology*, a palette of available visualization templates (bar charts, scatter plots, map views, *etc.*) into which analysts can slot their data. This method of interaction will be immediately familiar to users of spreadsheet programs: one picks a chart type and assigns data variables to visual aspects such as the X/Y axes and the size or color of visualized marks. A chart typology has the benefits of simplicity and familiarity, but it also limits the types of possible visualizations and makes it cumbersome to try out different visualizations of the same data.

Some visualization system designers have explored alternative approaches. Classic scientific visualization systems and more recent platforms for artistic data use a *data flow graph*, in which the visualization process is deconstructed into a set of finer-grained operators for data import, transformation, layout, coloring, *etc.* Users interactively chain these operators together to construct novel displays. Through flexible combinations of operators, data flow models can enable a larger space of visualization designs. However, data flow systems are more complex than chart typologies and may be limited by the set of available operators. In many cases, novel designs require users with programming expertise to develop new “building blocks” for the system.

Other systems are based on *formal grammars* for visualization construction. These grammars constitute high-level languages for succinctly describing how data should be mapped to visual features. By combining a handful of such statements, users can construct complex, customized visualizations with a high-degree of design control. This approach is used by a number of popular data visualization frameworks, such as Wilkinson's Grammar of Graphics \cite{wilkinson:gg}, ggplot2 for R \cite{wickham:ggplot2}, and Protovis for HTML5 \cite{bostock:pv}. However, each of the above requires at least minimal programming ability. Tableau \cite{tableau} (née Polaris \cite{stolte:polaris}) provides an example of visualization specification by drag-and-drop operations: users place data variables on “shelves” corresponding to visual encodings such as spatial position, size, shape, and color (see Figure \{fig-tableau\}). The visual specification is then translated into an underlying formal grammar that determines both the visualization design and corresponding queries to a database. This approach leverages the expressiveness of formal grammars while avoiding the need for programming. Another advantage is that formal grammars can be augmented with automated design facilities: a system can generate multiple visualization suggestions from a partial specification \cite{mackinlay:apt, mackinlay:showme, roth:sage}. While both fluent and expressive, interfaces utilizing formal grammars often have steeper learning curves than the more familiar chart typology.

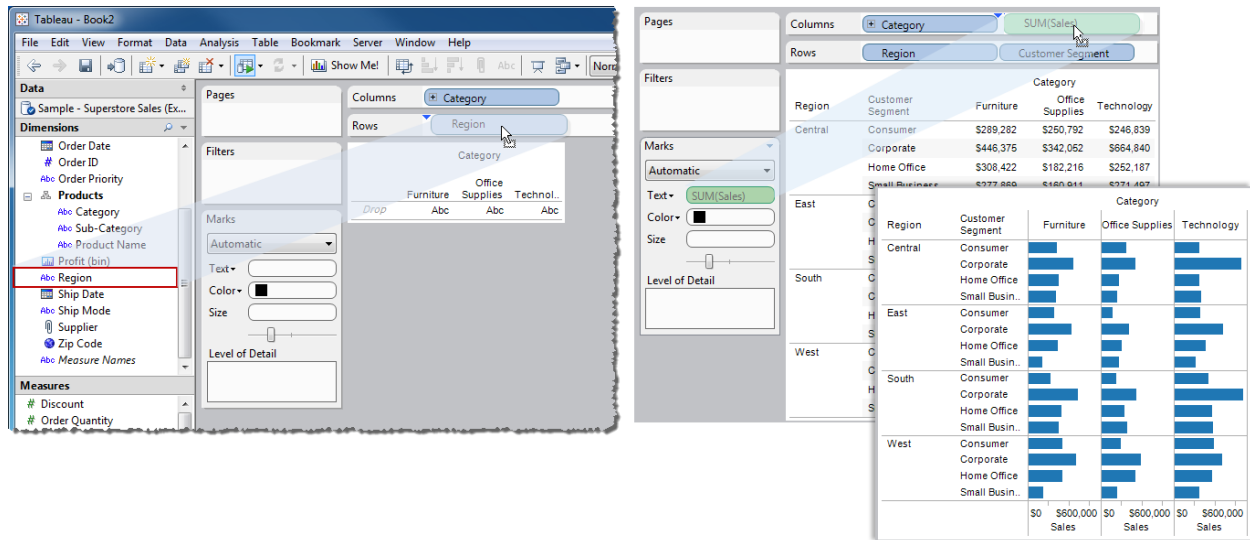


Figure {fig-tableau}: Visual encoding via drag-and-drop actions in Tableau. Analysts incrementally build up visualizations by mapping data variables to display properties such as position, size, color, and label text.

Fortunately, these methods are not mutually exclusive. Users can apply a data flow system or formal grammar to define new components to include within a chart typology, leveraging the improved expressiveness of the former and the ease-of-use of the latter. Going forward, novel interfaces for visualization specification are still needed. Formal grammars that use graphical marks (e.g., rectangles, lines, plotting symbols, *etc.*) as their basic primitives provide a conceptual model compatible with interactive design tools. New tools requiring little to no programming might place custom visualization design in the hands of broader audiences.

2.2 Filter

Once data has been visualized, analysts often want to shift their focus among data subsets. As discussed later, analysts might directly select (e.g., “lasso”) elements in a display and then highlight or exclude them. Another option is to use a suite of auxiliary controls, or *dynamic query widgets* [cite{shneiderman:dq}], for controlling element visibility (see Figure {fig-dqwidgets}). The choice of appropriate widget is largely determined by the underlying data type. Categorical or ordinal data can be filtered using simple radio buttons or checkboxes (when the number of distinct elements is small), or scrollable lists, hierarchies, and search boxes with autocomplete (when the number of distinct elements is large, or contains arbitrary text). Quantitative data can be filtered using a standard slider for a single threshold value, or a range slider for specifying multiple points. When coupled with real-time updates to the visualization, these widgets allow rapid and reversible exploration of data subsets. Query controls can be further augmented with visualizations of their own; Figure {fig-dqhistogram} shows a range slider augmented with a histogram of the underlying values.

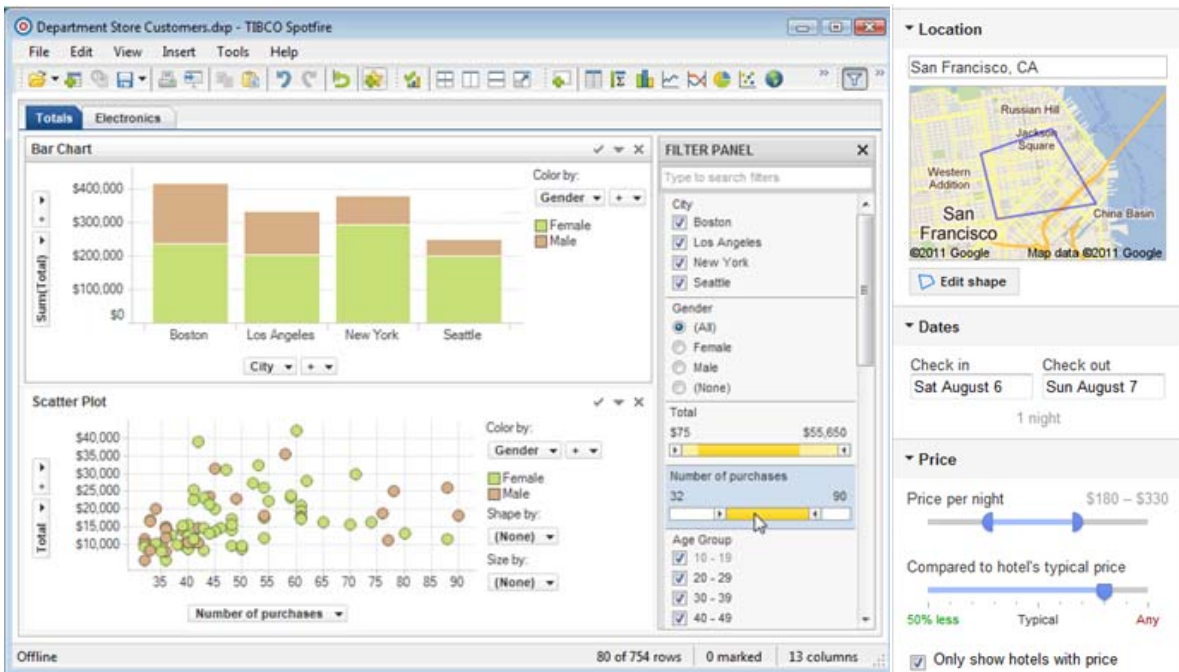


Figure {fig-dqwidgets}: Examples of dynamic query filter widgets. *Left:* Spotfire provides a variety of controls for filtering visualized data: checkboxes and radio buttons filter categorical variables, while range sliders filter numerical values. *Right:* Google Hotel Search provides widgets for geographic, date, and price ranges.

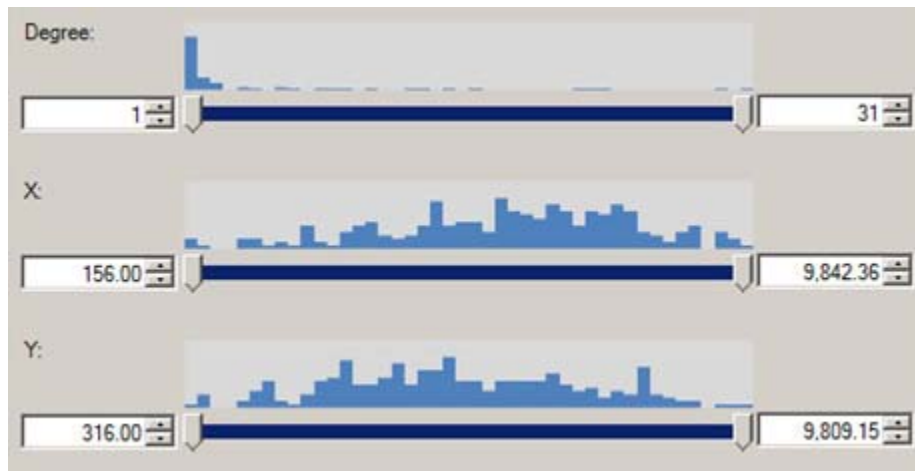


Figure {fig-dqhistogram}: Histogram Sliders in NodeXL. Selection widgets can be augmented with small visualizations showing the distributions of queried data attributes.

Expert users also benefit from more advanced functionality. For instance, a search box might support sophisticated query mechanisms, ranging in complexity from simple keyword search, to regular expression matching, to a full-fledged structured query language. While these additional mechanisms may not support rapid, incremental exploration as fluently as graphical widgets, they provide a means for expressing more nuanced criteria.

2.3 Sort

Ordering (or sorting) is another fundamental operation within a visualization. A proper ordering can effectively surface trends and clusters of values \cite{becker:trellis, yi:role}. The most common method of ordering is to sort records according to the value of one or more variables. Ordering becomes more complicated in the case of small multiples displays, in which both entire plots and the values they contain may be sorted to surface patterns. Sorting values consistently across plots (for example, by their marginal mean or median values) can reveal patterns while facilitating comparison among plots.

Some data types (e.g., multivariate tables, networks) do not always lend themselves to simply sorting by value. Such data requires more sophisticated *seriation* methods \cite{hahsler:seriation, wilkinson:gg, wilkinson:heatmap} that try to minimize some distance function among elements. An example is shown in Figure {fig-sortmatrix}, a matrix-based visualization of a social network. On the left, the raw order of data values exposes little structure; re-ordering the rows and columns to place connected actors in close proximity reveals underlying clusters of communities.

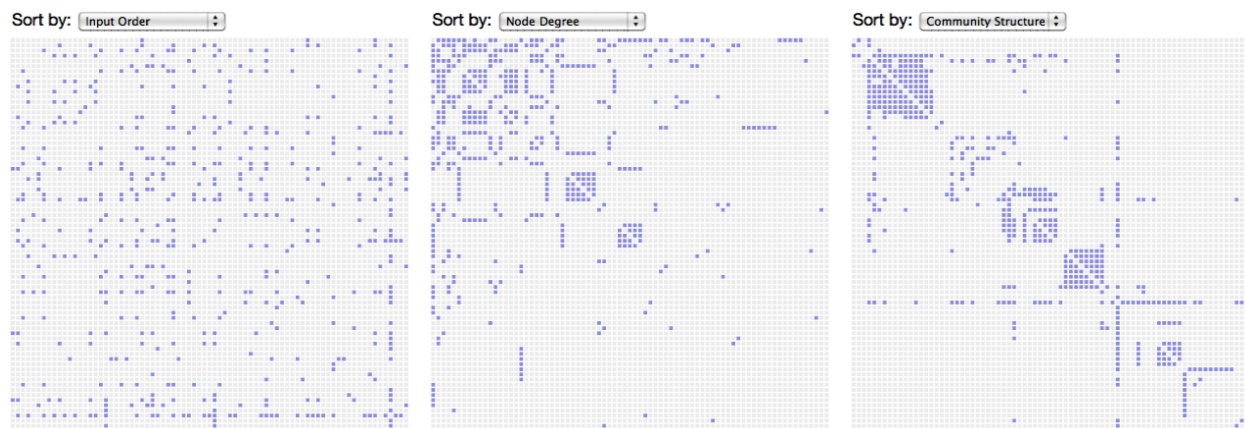


Figure {fig-sortmatrix}: Reorderable Matrices. *Left.* A matrix plot of a social network conveys little structure when the rows and columns (representing people) are sorted alphabetically. *Center.* Interactively re-ordering the matrix by node degree reveals more structure. *Right.* Sorting the matrix by network connectivity reveals community clusters.

2.4 Derive

As an analysis proceeds in iterative cycles, users may find that the input data is insufficient: variables may need to be transformed, or new attributes derived from existing values. Common cases include normalization, log transforms, and the like. While users can derive new values prior to importing data for visual analysis, the overhead of moving between tools stymies fluid, iterative exploration. As a result, visual analytics tools should include facilities for deriving new data from input data. Most often this functionality is provided via a *calculation language*, similar to those found in spreadsheets. Other examples include support for statistical methods, ranging from simple summary statistics (mean, median, variance) to model fitting (regression curves) and data transformation (group-by aggregation, outlier suppression). Beyond these basic functions that provide much-needed flexibility, hypothesis testing methods (t-tests, ANOVA) amplify the benefits of smooth integration of statistics and visualization.

Looking forward, improved derivation methods present a promising frontier for visual analytics research. How can visual tools support flexible construction of more advanced models or derived values? Using programming-by-demonstration methods, analysts might annotate patterns (e.g., of network intrusion events \cite{xiao:hr}) from which the system can generalize pattern recognition rules. Or, visual tools might automatically fit applicable statistical models to the data based on the current visualization state. For example, the nesting of variables within common “pivot” displays could be mapped to the structure of a linear model. More principled frameworks that wed visualization to modeling and forecasting are still emerging.

3. VIEW MANIPULATION

Once analysts have created a visualization through data and view specification actions, they should be able to manipulate the view to highlight patterns, investigate hypotheses and drill-down for more details. Analysts must be able to **select** elements or data regions to highlight, filter, or operate on them. Large information spaces require analysts to scroll, pan, zoom and otherwise **navigate** the view to examine both high-level patterns and fine-grained details. Multiple, coordinated visualizations regularly provide clearer insights into multi-dimensional data than isolated views. Analysts must be able to **coordinate** rich, multi-view displays and **organize** their dashboards and work spaces, so that filtering applies to all displays at once and selection in one display causes highlighting in the others (see Figure {fig-dqwidgets}).

3.1 Select

Pointing to an item or region of interest is common in everyday communication because it indicates the subject of conversation and action. In the physical world, people coordinate their gestures, gaze, and speech to indicate salient elements. For example, different hand gestures can communicate angle (oriented flat hand), height (horizontal flat hand), intervals (thumb and index finger in “C” shape), groupings (circling a region), and forces (accelerating fist) \cite{hill:deixis}. In visual analysis, reference (or *selection*) remains of critical importance, but is realized through a more limited set of actions, such as clicking or lassoing items of interest.

Common forms of selection within visualizations include mouse hover, mouse click, region selections (e.g., rectangular and elliptical regions, or free-form “lassos”), and area cursors (e.g., “brushes” \cite{becker:brushing} or dynamic selectors such as the bubble cursor \cite{grossman:bubble}). These selections often determine a set of objects to be manipulated, such as highlighting, annotation, filtering, or accessing details-on-demand. The context of interaction must be taken into account when choosing a selection method. For instance, responding to hover events to provide details-on-demand is inappropriate when using touch-based input on a tablet or mobile phone.

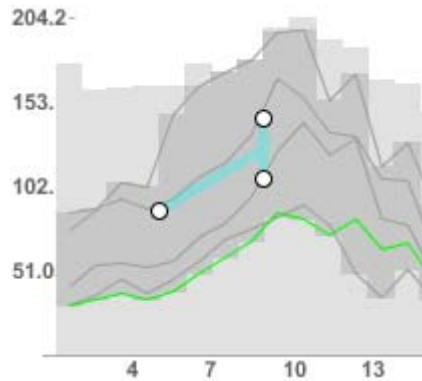


Figure {fig-timesearcher}: Querying Time-Series by Slope in TimeSearcher. An angular selection tool specifies a target slope (rate of change) and tolerance for a collection of stock prices. The widget operates directly on the visualization: dragging the widget from left to right interactively queries other time windows.

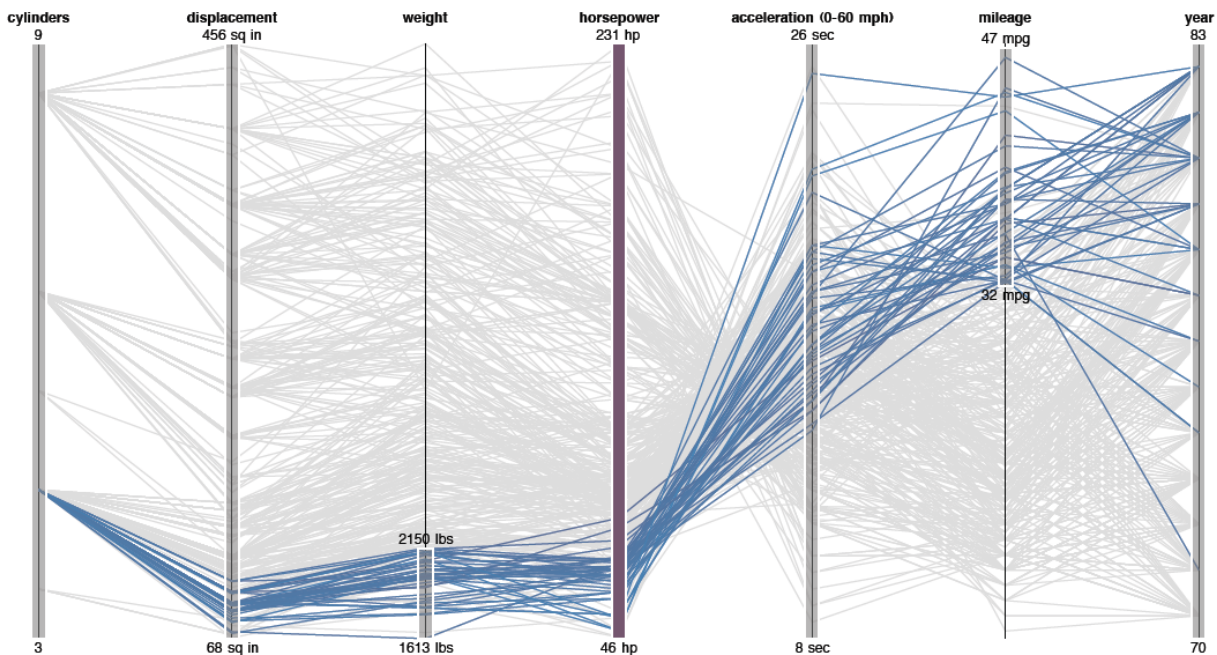


Figure {fig-parallel}: Selection Queries in Parallel Coordinates. Parallel coordinates plot multidimensional data as line segments among parallel axes. Here, a user has dragged along the axes to create interactive selections that highlight automobiles with low weight and high mileage.

Selections can also vary in terms of their expressive power. Most interfaces support selections of a collection of items. While this approach is easy to implement, it does not allow users to specify higher-level criteria. A more powerful, albeit more complex, approach is to support selections as queries over the data. Maintaining query structure increases the expressiveness of visualization applications. For instance, rather than directly select the contained elements, drawing a rectangle in a chart may specify a range query over the data variables represented by the X- and Y-axes. The resulting selection criteria can then be saved and applied to dynamic data (updating elements may enter or exit a query region) or to a completely different visualization. Examples include querying stock price changes in TimeSearcher

\cite{hochheiser:timesearcher} (Figure \ref{fig-timesearcher}) and attribute ranges in parallel coordinates displays \cite{inselberg:parallel} (Figure \cite{fig-parallel}).

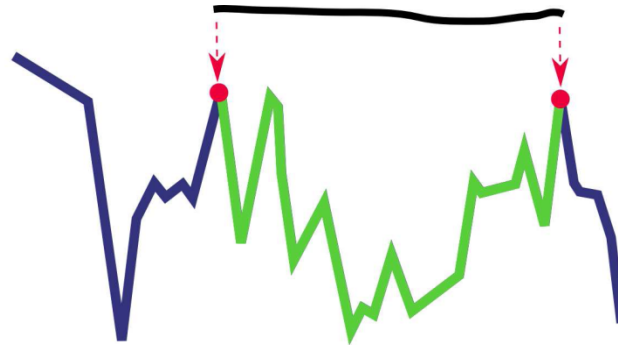


Figure {fig-ink}: Perceptual Interpretation of Ink Annotations \cite{kong:ink}. Matching input strokes (black) to a time-series segmentation model enables rapid annotation of perceptually-salient regions such as spans, peaks, and slopes.

Designing more expressive selection methods remains an active area of research. For example, researchers have proposed methods to map mouse gestures over a time-series visualization to select perceptually salient data regions such as peaks, valleys and slopes \cite{kong:ink} or to query complex patterns of temporal variation \cite{holz:relax}. Initial selections can also be used as a starting point for more complex selections, as users might click a representative object and then formulate a broader selection based on the object’s properties (e.g., “select all items blue like this one”). Of course, selection need not be limited to the mouse and keyboard: input modalities such as touch, gesture and speech might enable new, effective forms of selection.

3.2 Navigate

How users navigate a visualization is in part determined by where they start. One common pattern of navigation adheres to the visual information seeking mantra: “Overview first, zoom and filter, then details-on-demand” \cite{shneiderman:eyes}. Analysts may begin by taking a broad view of the data, including assessment of prominent clusters, outliers and also potential data quality issues. These orienting actions can then be followed by more specific, detailed investigations of data subsets.

Of course, starting with an expansive overview is not always advisable. A legal analyst researching for an upcoming trial may be wise to forego an overview of the entire history of U.S. court decisions. Instead, the analyst might start with the legal decisions most relevant to the current case, perhaps determined by keyword search, and expand her investigation to other, cited decisions. This form of navigation can be summarized as “Search, Show Context, Expand on Demand” \cite{vanham:doi}.

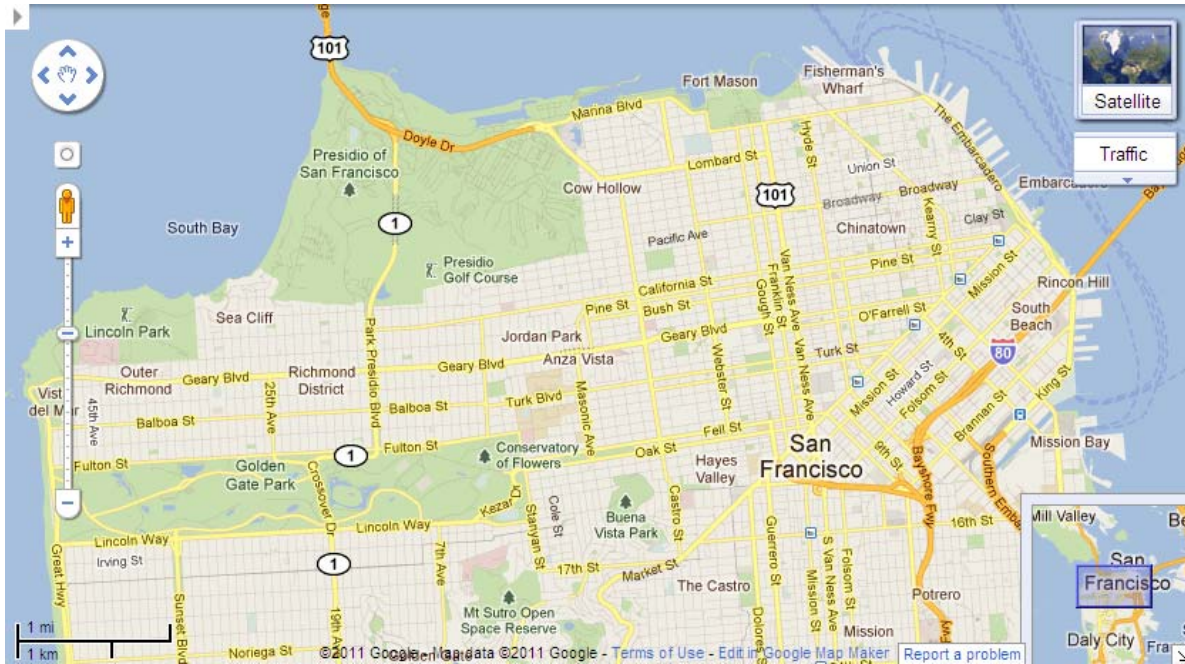


Figure (fig-map): Zoomable Tile Map. Google Maps provide controls for pan (drag) and zoom (slider and scroll wheel) and include an overview display (bottom-right) for showing the larger geographic context.

Visualizations often function as "viewports" onto an information space. Analysts then need to manipulate the viewport to navigate the space. Common examples include scrolling or panning a display via scroll bars or mouse drag, and zooming among different levels using a zoom slider or scroll wheel. Zooming need not follow a strict geometric metaphor: *semantic zooming* \cite{bederson:padpp} methods can modify both the amount of information shown and how it is displayed as users move among levels of detail (Figure {fig-datelens}). Additionally, dynamic query widgets, such as range sliders for the X and Y axes of a scatter plot, can filter the visible data range and thus provide a form of zooming within a chart.

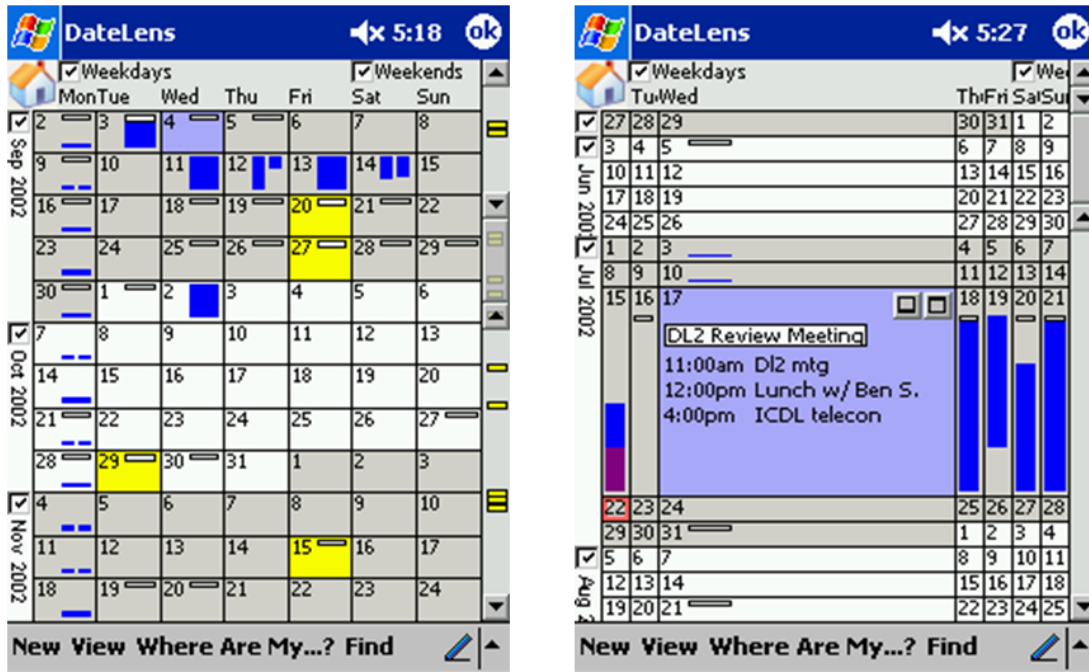


Figure (fig-datelens): DateLens Focus+Context Calendar. As users navigate from months to days to hours, the display magnifies selected regions. Semantic zooming reveals more details within focal regions.

To further aid navigation, researchers have developed a variety of *focus plus context* methods, sometimes called “bifocal” views [spence:bifocal], that provide a detailed view of a high interest data region while retaining surrounding context to help keep analysts oriented. A second key idea is the use of *overview and detail* displays. For example, a geographic visualization (Figure {fig-map}) might include a large zoomed-in map (the detail), while a smaller, zoomed-out map includes a rectangle showing the position of the zoomed-in view within the broader terrain (the overview). In this case, the detail view provides the focus, and the overview provides context. The benefits are highest when the zoom factor (ratio of overview to detail view) is 5-20 [zoom:factor]. When larger zoom factors are needed, intermediate overviews have proven to be helpful.

A different approach is to use *distortion* or magnification techniques that transform the entire display region such that contextual regions are demagnified. A simple example is the MacOS X dock, which uses 1D fisheye distortion to show common applications; more sophisticated methods employ distortion in multiple dimensions. While often visually intriguing, complex distortion methods have yet to prove their worth in real-world applications: viewers can become disoriented by non-linear distortions and show little performance improvement over simpler methods such as zooming [nekrasovski:nav].

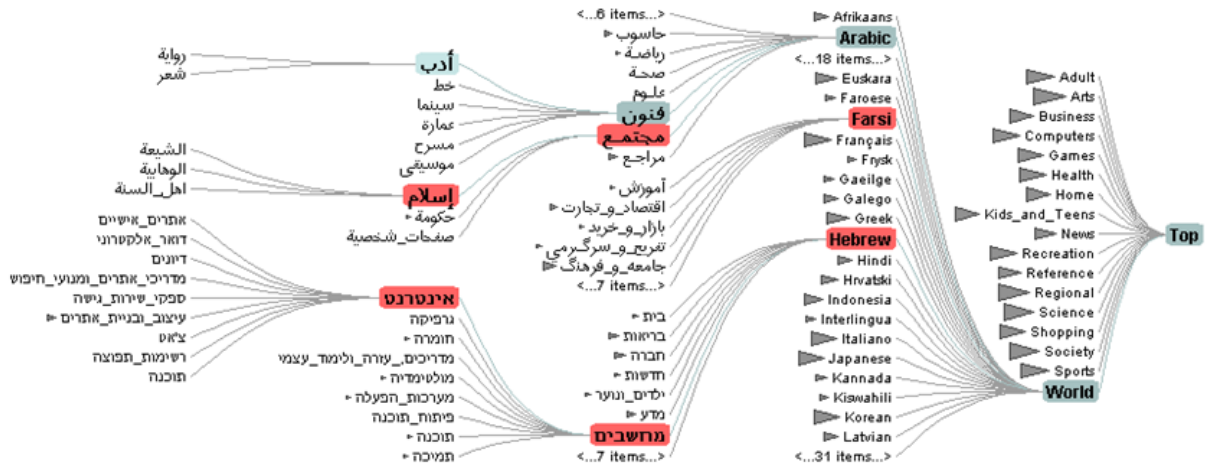


Figure {fig-doitree}: Degree-of-Interest Tree showing a 600k element taxonomy \cite{heer:doi}. A model of the user's current interest filters the display to the most relevant elements. Low-interest elements are elided but still accessible through aggregate representations. The interest estimates update as an analyst explores the taxonomy, initiating animated transitions between different views of the data.

In addition to manipulating display space, focus plus context methods can be applied directly to the data itself. The goal is to identify which data elements are currently of high interest (focus), which are of high importance regardless of the current focus (context), and which can safely be removed from view. *Degree-of-Interest* (DOI) functions \cite{furnas:followup, heer:doi, vanham:doi} calculate scores for information content based both on general importance (e.g., top-level categories within a hierarchy, or nodes with high centrality in a graph) and current interest (e.g., as indicated by mouse clicks, search queries, or proximity to other high interest elements). The distribution of DOI scores can then be used to selectively control the visibility of elements based on the current view size and context of interaction, as in Figure {fig-doitree}. As users click on or search for different elements, the DOI scores dynamically update to reveal relevant unseen data or hide irrelevant detail.

Visualizations can provide cues to assist users' decisions of where and how to navigate. The controls for view manipulation have often been invisible, such as zooming/panning by mouse movement. Improved strategies facilitate discovery by users, and provide visible indication of settings in legends or other ways, such as scroll bar positions, that provide informative feedback. An important challenge is to show selected items, even when they are not in view. For example, the results of a text search that are not currently in view might be shown by markers in the scroll bar \cite{zellweger:lights} or the periphery of the display \cite{baudisch:halo, gustafson:wedge}.

3.3 Coordinate

Many problems require *coordinated multiple views* that enable users to see their data from different perspectives. A public policy analyst studying educational attainment might produce a bar chart of people's ages, a map of locations, a textual list with education history, and a scatter plot showing income vs. education. By selecting a single item or a group in one view,

analysts might then see related details or highlighted items in the other views. This powerful approach to exploring multivariate data also enables drilling down into subgroups, marking of sets, and export of selections.



Figure {fig-multiples}: Small Multiples Display. The visualization shows employment figures by economic sector in the state of Minnesota. The repetition of the chart form supports comparison among sectors. Plotting all the data in one chart would otherwise clutter and obscure individual trends.

Multi-view displays can facilitate comparison. For example, Tufte \cite{tufte:visual} advocates the use of *small multiples*: a collection of visualizations placed in close spatial proximity and typically using the same measures and scales. As in Figure {fig-multiples}, these small multiples, also called *trellis views*, enable rapid comparison of different data dimensions or time-slices. Alternatively, multi-view displays can use a variety of visualization types -- such as histograms, scatter plots, maps, or network diagrams -- to show different projections of a multidimensional data set (Figure {fig-improvise}). Though comparing multiple visualizations requires viewers to orchestrate their attention and mentally integrate patterns among views, this process is often more effective than overburdening a single visualization with too many dimensions.

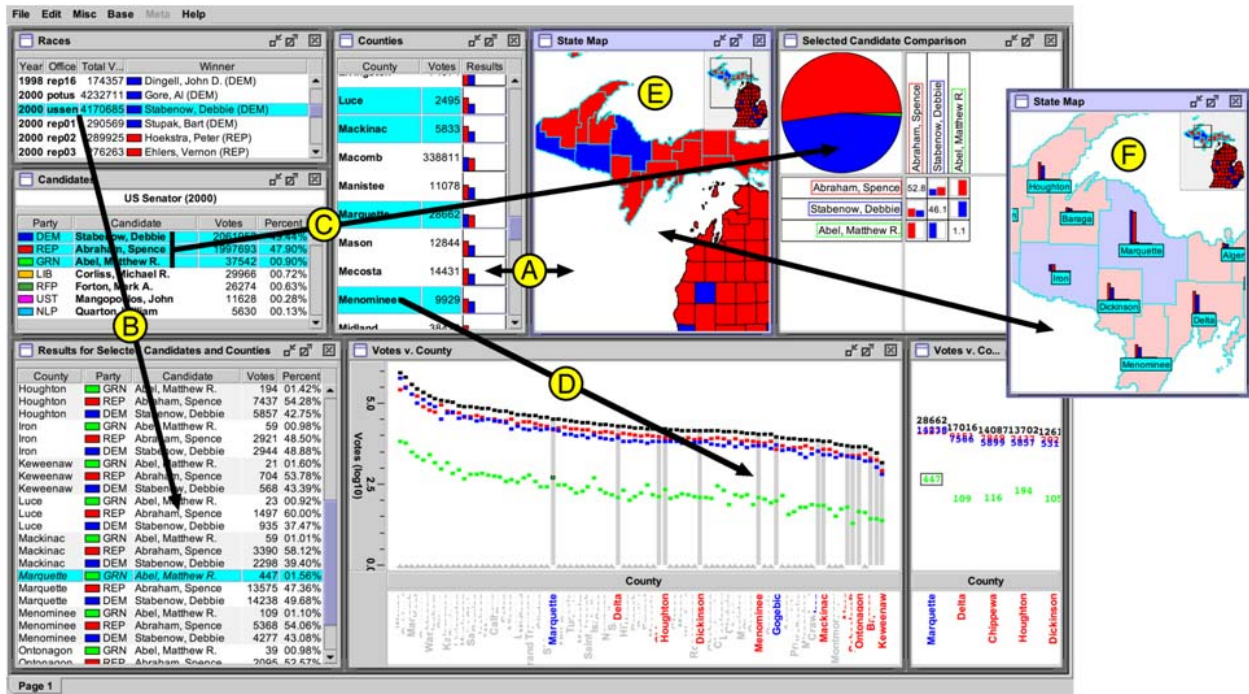


Figure (fig-improvise): Multiple coordinated views in Improvise \cite{weaver:improvise}. An analyst constructs a complex patchwork of interlinked tables, plots and maps to analyze the outcomes of elections in Michigan.

Multi-view displays may be comprised of diverse components. Accompanying elements such as legends, histogram sliders, and scroll bars with highlighting markers can all provide multiple views onto the data. Automatically generated legends and axes are important for providing accurate annotations for analysts and meaningful explanations when visualizations are shared. Legends and axes can also become control panels for changing color palettes, marker attributes, variable ranges or provenance information \cite{riche:legends}.

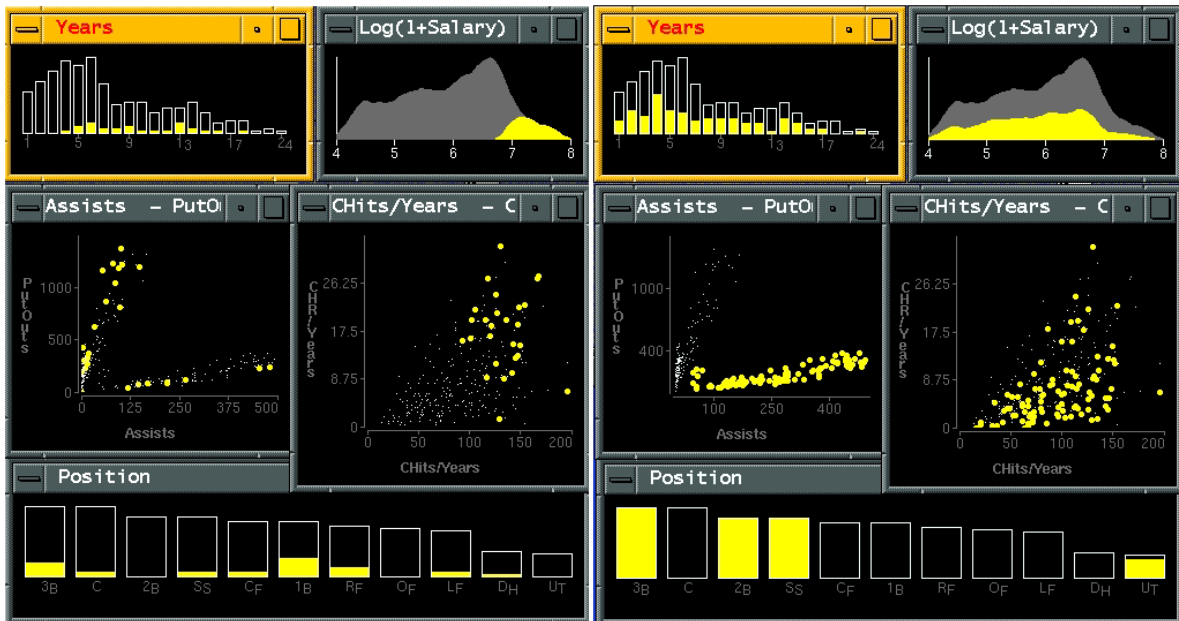


Figure {fig-brushing}: Brushing and linking of baseball statistics in GGobi. As analysts make selections in one plot, the corresponding elements highlight in the others. *Left:* selecting high income players (top-right plot) shows little dependence on career length or fielding ability, but correlates with hitting performance. *Right:* selecting the cluster of players who make more assists than put-outs (middle-left plot) reveals a strong dependence on position.

Multi-view displays can also enable interactive exploration across views. *Brushing and linking* is the process of selecting ("brushing") elements in one display to highlight (or hide) corresponding data in the other views \cite{becker:brushing}. Each visualization can thus serve as an input channel for revealing patterns across a data set (Figure \ref{fig-brushing}). Linked selection enables rich, multidimensional reasoning by allowing analysts to assess how patterns in one view project onto the others. Analysts may wish to coordinate views in variety of ways \cite{north:snap, weaver:improvise}: selecting elements in one view might highlight matching records in other views, or instead provide filtering criteria to remove information from the other displays. Linked navigation provides an additional form of coordination: scrolling or zooming one view can simultaneously manipulate other views.

Studies of how analysts construct multi-view displays and specify coordination behaviors (e.g., highlighting, filtering) provide designers with an understanding of how to build more effective tools. In addition, if designers ensure that rich multi-view displays stay understandable, analysts are more likely to make compelling insights. Newcomers to an analysis, or even seasoned analysts simply returning from a coffee break, may become confused by the number of views and the potentially complicated set of coordinated queries between them. Visual analytics systems that provide access to coordination settings and replay the history of view construction enhance understanding.

3.4 Organize

When analysts make use of multiple views they face the corresponding challenge of managing a collection of visualizations. As in traditional window-based interfaces, analysts may wish to open, close, maximize, and layout different components. As purely manual window manipulation can be tedious, visual analytics tools can simplify the organization of visualization views, legends, and interactive controls. Many visual analysis systems provide facilities for creating visualization dashboards -- a collection of non-overlapping, often linked, views -- by manually arranging a set of separately-constructed, individual views. More advanced systems might aid this process through automated support \cite{bell:space, kandogan:ew}. Useful methods include suggesting related views of interest (e.g., of data variables correlated to the visualized attributes), automatic (re)sizing as views are added or removed, and layout routines to place related views in spatial proximity.

A related issue arises when organizing different analysis trails, akin to managing multiple worksheets in a spreadsheet application. Effective view organization should also support analysts in maintaining and communicating an overview of their analysis sessions. As described in the next section, managing the analysis process constitutes a critical aspect of interactive analysis.

4. PROCESS & PROVENANCE

Visual analytics is not limited to the generation and manipulation of visualizations -- it is fundamentally about the process of iterative data exploration and interpretation. As a result, visual analytics tools should provide facilities for scaffolding the analysis process. Tools should preserve analytic provenance by keeping a **record** of user actions and insights so that the history of work can be reviewed and refined. Textual logs of activity have benefits, but visual overviews of activity can be more compact and comprehensible. If analysts can **annotate** patterns, outliers and views of interest, they can document their observations, questions and hypotheses. In modern systems, analysts are empowered to **share** results and discuss with colleagues, coordinate the work of multiple groups, or support processes that may take weeks and months. Moreover, analysis tools can explicitly **guide** novices through common analysis tasks or provide progress indicators for experts.

4.1 Record

When analyzing data with visualizations, users regularly traverse the space of views in an iterative fashion. Exploratory analysis may result in a number of hypotheses, leading to multiple rounds of question-answering. Analysts can generate unexpected questions that may be investigated immediately or revisited later. After conducting analysis, analysts may need to review, summarize, and communicate their findings, often in the form of reports or presentations.

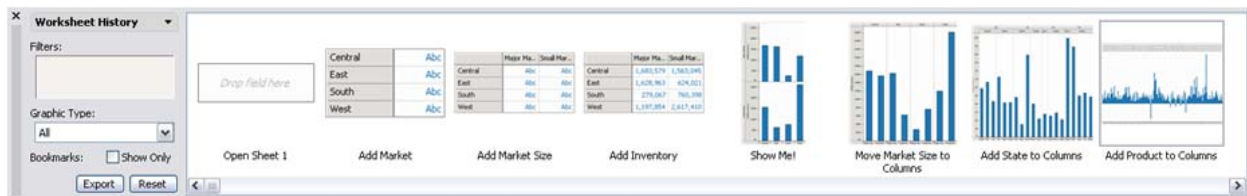


Figure {fig-history}: Visual Analysis History \cite{heer:history}. A “comic strip” display retraces the steps taken in a visual analysis of business operations data.

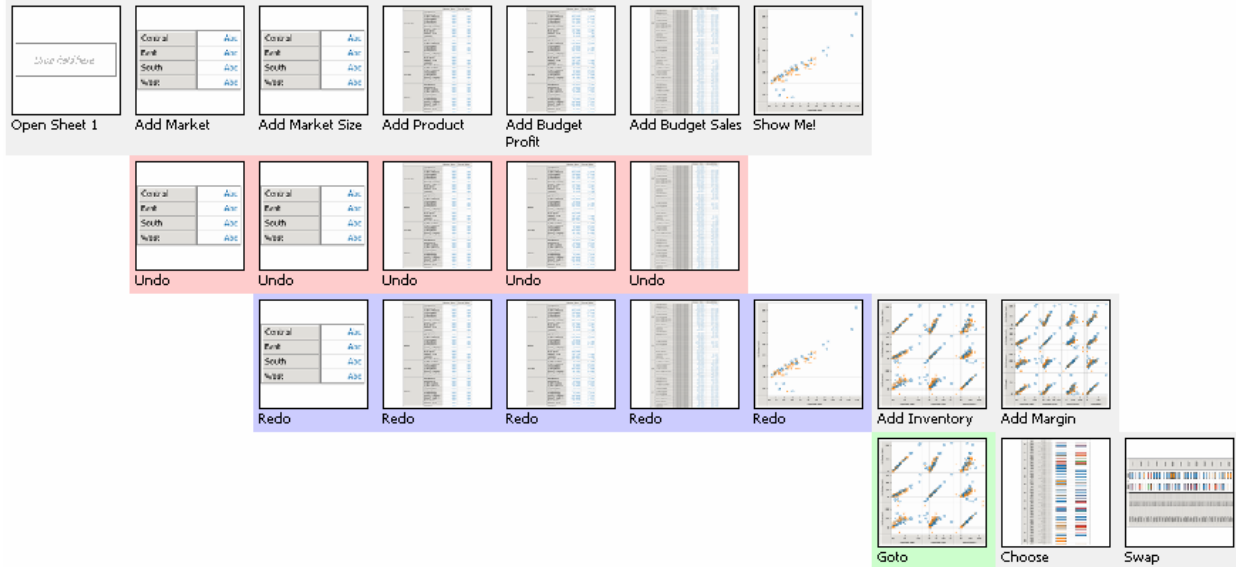


Figure {fig-branching}: Behavior Graph of Branching Analysis \cite{heer:history}. Reading the graph in a snake-like fashion (first left-to-right, then right-to-left) reveals patterns of iterative exploration, branching, and backtracking in an analysis.

To support iterative analysis, visual analysis tools can record and visualize users' *interaction histories*. At a minimum, applications should provide basic undo and redo support. While low-level input such as mouse and keyboard events are easy to capture, histories become much more valuable when they record high-level semantic actions. By modeling the space of user actions (e.g., view specifications, sorting, filtering, zooming, *etc.*), richer logs can be constructed and visualized \cite{derthick:history, heer:history, jankun:model, scheiddeger:prov}. Common visual representations of analytic actions include both chronological ("timeline") and sequential ("comic strip") views (Figure \ref{fig-history}). As shown in Figure \ref{fig-branching}, visual histories also reveal the hierarchical patterns of branching histories. Techniques for "chunking" related actions together can further reduce clutter \cite{kurlander:history, heer:history}.

Visual histories can support a range of interactions. First, histories provide a convenient mechanism to revisit prior analysis states and resume incomplete explorations. Adding metadata such as comments, tags, or ratings to states can facilitate later review and sharing. Interactive histories can also capture a repeatable sequence of operations that can be named and saved as a reusable macro. This powerful feature enables analysts who are dealing with thousands of datasets to automate their efforts in a reliable manner. Histories might spur sharing: analysts can export selected analysis trails to external media, ranging from screenshots to interactive presentations. Finally, histories also provide a means to study users and model analytic processes \cite{jankun:model, scheiddeger:prov}.

4.2 Annotate

The goal of visual analysis is to generate new insights from data. Often this goal requires that interactive visualizations serve not only as data exploration tools, but also as a means for recording, organizing, and communicating insights gained during exploration. One option discussed previously is to allow textual annotation of states within a visual history. More

expressive annotations are possible through direct interaction with the view, as analysts may wish to "point" to specific elements or regions within a visualization; explanatory text or links to other views can be associated with these annotations \cite{heer:voyagers}.

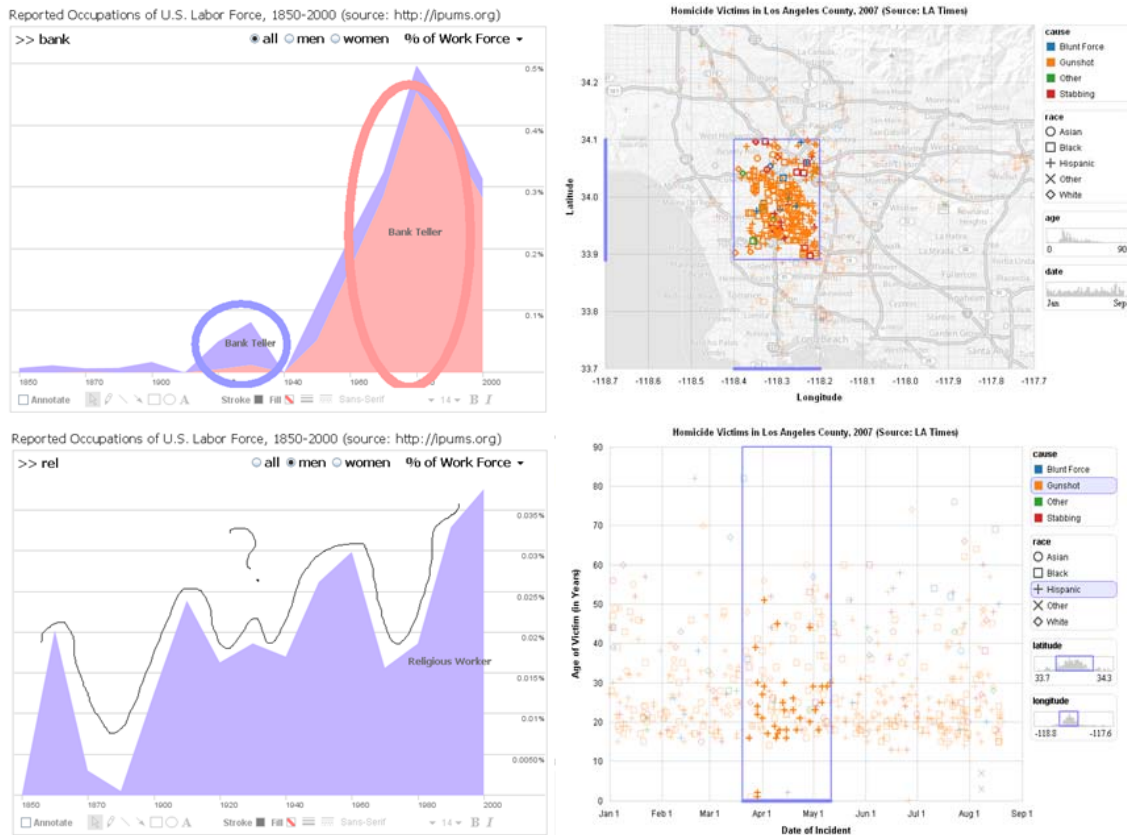


Figure {fig-annotate}: Annotated Visualizations. *Left:* Freeform graphical annotations of occupational data. The top annotation highlights a gender reversal among bank tellers using color-coded ellipses; the bottom annotation expresses confusion regarding the erratic percentage of religious workers. *Right:* Data-aware annotations. In the top charts, selection queries anchor annotations of crime data. The bottom chart shows annotations transferred across a change in visual encodings: the selected geographic range is now conveyed using histogram sliders.

Freeform *graphical annotations* provide one expressive form of pointing \cite{heer:voyagers}, shown on the left side of Figure {fig-annotate}. Drawing a circle around a cluster of items or pointing an arrow at a peak in a graph can direct the attention of viewers. The angle or color of the arrow or shape of the hand-drawn circle may communicate emotional cues or add emphasis. Although such drawings allow a high degree of expression, they lack any explicit tie to the underlying data. Freeform annotations can persist over geometric transformations such as panning and zooming, but if they are not "*data-aware*" they may become meaningless in the face of operations such as filtering or drill-down.

Annotations can be made data-aware when realized as selections (see the right side of Figure {fig-annotate}). These selections can be represented by a set of selected elements, a declarative query, or both. Data-aware annotations allow a pointing intention to be reapplied to different views of the same data, enabling reuse of references across different choices of visual

encodings. Data-aware annotations may also enable users to search for all commentary or visualizations that reference a particular data item. As data-aware annotations are machine-readable, they might also be used to export selected data or aggregated to identify data subsets of high interest.

4.3 Share

Researchers in visual analytics often focus on the perceptual and cognitive processes of a single analyst. In practice, real-world analysis is also a social process that may involve multiple interpretations, discussion, and dissemination of results \cite{heer:voyagers, ibm:manyeyes}. The implication is clear: to fully support the analysis life-cycle, visual analytics tools should support social interaction. At minimum, tools must be able to export views (png, jpg, ppt, etc.) or data subsets (csv, json, xls, etc.) for sharing and revisitation. An important capability is to export the settings for the control panels, so other analysts can see the same visualization. Figure {fig-senseus} shows one example of a collaborative visual analysis tool incorporating view sharing, annotation and discussion.

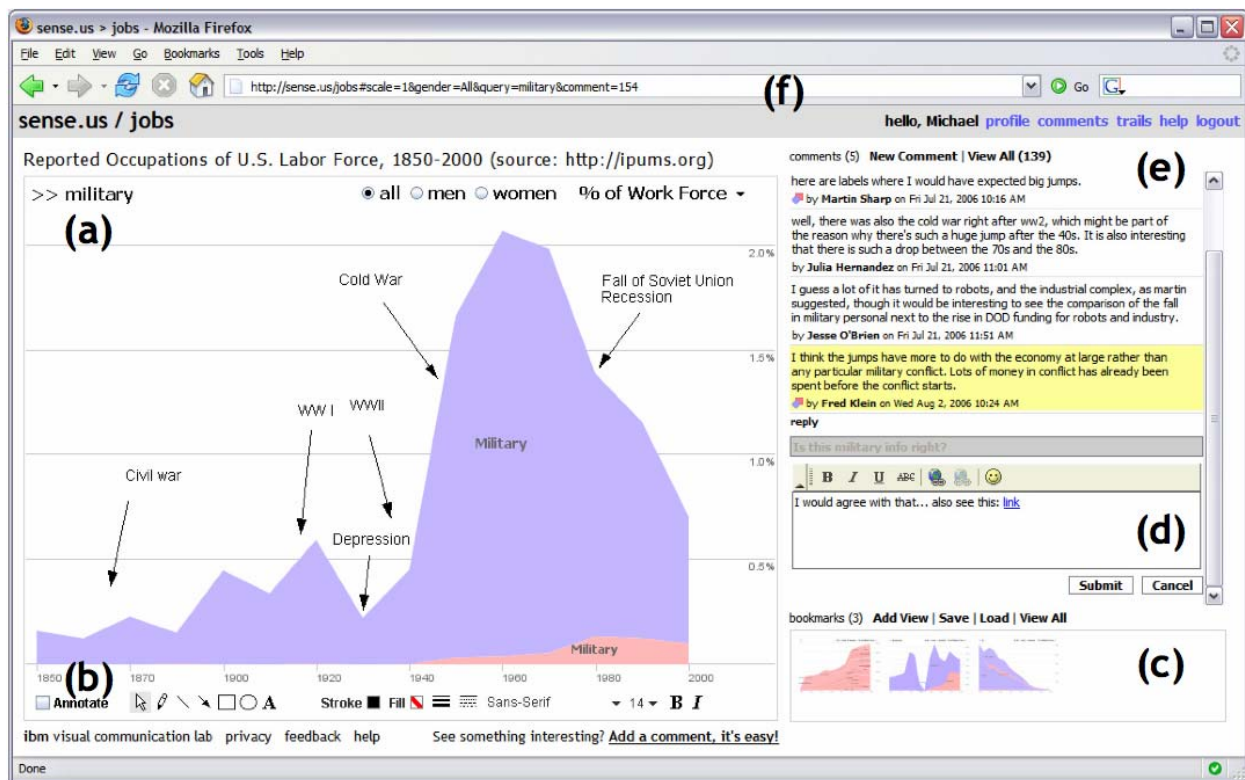


Figure {fig-senseus}: Collaborative Visual Analysis in Sense.us \cite{heer:voyagers}. (a) An interactive visualization, with a graphical annotation for the currently selected comment. The visualization is a stacked time-series visualization of the U.S. labor force, broken down by gender. Here the percentage of the work force in military jobs is shown. (b) A set of graphical annotation tools. (c) A bookmark trail of saved views. (d) Text-entry field for adding comments. Bookmarks can be dragged onto the text field to link views to a comment. (e) Comments attached to the current view. (f) Shareable URL for the current application state. The URL is updated automatically as the visualization state changes.

A simple but effective aid to collaboration is view sharing via *application bookmarking*: a visual analytics system should be able to model and export its internal state \cite{heer:voyagers, ibm:manyeyes}. View sharing often takes the form of a URL or similar identifier that allows a collaborator to quickly navigate to a view of interest. Seeing an identical view provides collaborators with a common ground for discussion. Annotation methods can be applied within such views to further collaboration. One challenge for effective view sharing concerns how to handle dynamic data: should a bookmarked view maintain a snapshot to historical data, provide access to the most current data, or both?

Another method of sharing and dissemination is to *publish* a visualization. Commercial tools such as Spotfire and Tableau can publish visualization dashboards as interactive web pages. These web-based components provide a subset of interactive functionality (e.g., selection, search, and drill-down) to enable some amount of follow-up analysis. Services such as IBM's Many-Eyes \cite{ibm:manyeyes} can be used to embed visualization applets in external web sites. Publishing is particularly important for engaging larger audiences. Visualizations embedded within a blog or discussion forum can reach an established audience and may foster discussion more effectively than a centralized visualization site \cite{danis:place}.

Other collaborative concerns depend on the context of use. Are collaborators working *synchronously* (same time) or *asynchronously* (different time)? Are they *co-located* (same place) or *distributed* (different place)? Each of these configurations may require specialized strategies that consider the division of work, access control, presence indicators, and activity awareness \cite{heer:voyagers, isenberg:esv}.

4.4 Guide

The exploration process is well understood for some traditional domains. For example, a very simple workflow might remove incomplete data items, sort, select high value items, and report on these selections. However, analysts may need to develop new strategies that are formalized to guide newcomers and provide progress indicators to experts. Visual analysis systems can incorporate *guided analytics* to lead users through workflows for common tasks.

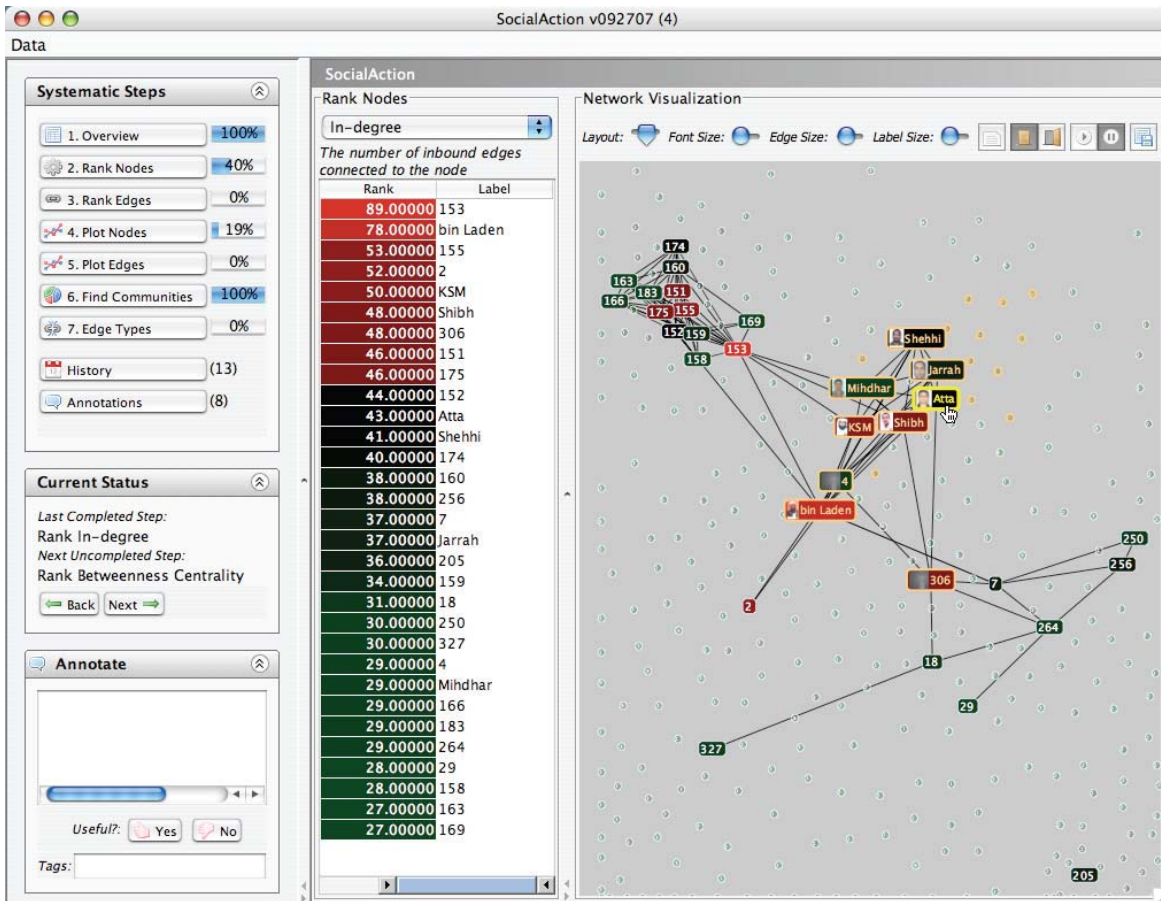


Figure {fig-socialaction}: Systematic yet flexible analysis in SocialAction \cite{perer:syf}. The panel on the left suggests common steps to structure social network analysis and provides progress indicators.

Some processes are clearly linear, but many visual analytics tasks require richer *systematic yet flexible* processes that allow users to take excursions while keeping track of what they have done. For example, SocialAction \cite{perer:syf} organizes social network analysis into a sequence of activities (e.g., rank nodes, plot nodes, find communities); the system allows analysts to selectively skip steps if they please and keeps a record of which steps have been completed (see Figure {fig-socialaction}). In a related vein, experts often develop visualizations that are used by less knowledgeable team members, in much the same way that spreadsheet macros enable specialists to encode accounting or business practices for others. More research is needed to identify effective visual analytics processes and enable expert analysts to author reusable workflows.

Budget Forecasts, Compared With Reality

Just two years ago, surpluses were predicted by 2012. How accurate have past White House budget forecasts been?

1 2 3 4 5 6 NEXT ▶

Latest forecast

Today, with a better understanding of the severity of the economic downturn, the deficit situation is much more dire.

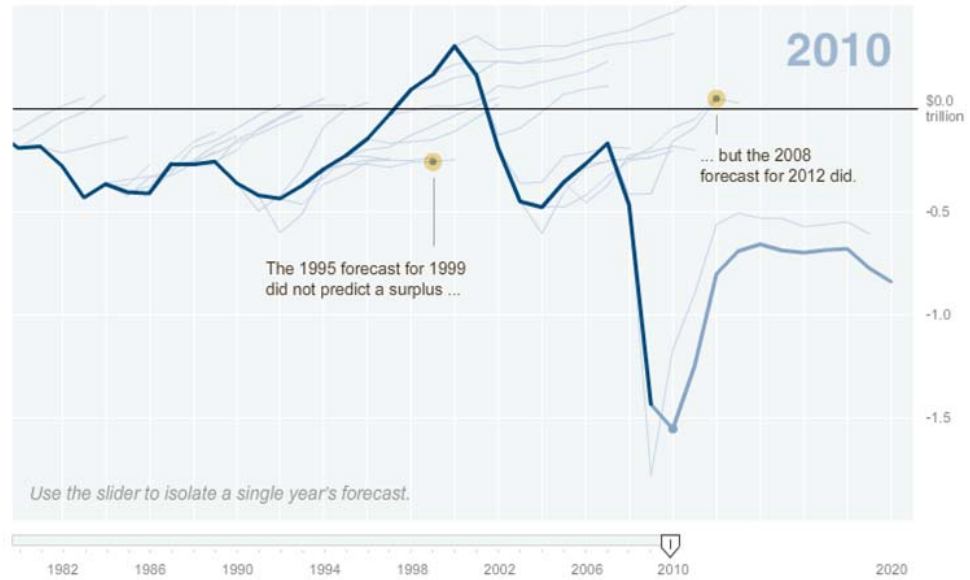


Figure {fig-nytimes}: Data storytelling by the New York Times. Through staging and annotations, this interactive graphic guides the reader through decades of budget predictions while also permitting interactive exploration.

In recent years, journalists have been experimenting with different forms of *narrative visualization* \cite{segel:narrative} by structuring interactive graphics to tell stories with data. Visualizations from the New York Times, Washington Post, the Guardian, and other news sources often lead the viewer step-by-step through a linear narrative, guided by supporting text and annotations (see Figure {fig-nytimes}). At a story's conclusion, the visualizations provide interactive controls for further exploration. These narrative structures both communicate key observations from the data and cleverly provide a *tacit tutorial* of the available interactions by animating each component along with the story. By the time the presentation opens up for freeform exploration, the viewers have already seen demonstrations of the interactive controls. These and other forms of narrative visualization are wonderful examples of how guided analytics can be used to disseminate data-driven stories to a general audience.

5. CONCLUSION

We hope the above taxonomy and discussion will help advance visual analytics on multiple fronts. For students and newcomers to the field, the taxonomy provides an orienting, high-level introduction to the interactive concerns at the heart of successful visual analysis. We encourage interested readers to consult the systems, books and research papers referenced in this article to develop a deeper understanding of these concerns. For developers, the taxonomy can function as a “check list” of elements to consider when creating new analysis tools. For researchers, the taxonomy helps to highlight critical areas that would benefit from further investigation, including new methods for visualization specification without programming, a

deeper marriage of visualization and statistical algorithms, selection and annotation techniques that leverage data semantics, and effective approaches to guided analytics.

Of course, by attempting to provide a simplified picture of a domain, taxonomies are often incomplete. In some cases, we have separately categorized aspects that are closely related. Dynamic query widgets enabling data specification often serve as a means of view navigation. Selection techniques are also central to effective annotation schemes.

In other instances, we have selectively omitted material. First, we do not go into great depth regarding implementation details. Supporting real-time interactivity often requires careful attention to system design, particularly for large data sets. While popular platforms for large data analysis such as MapReduce \cite{mapreduce} achieve adequate throughput, high latency and lack of online processing limit fluent interaction. The demands of truly interactive analysis pose important research challenges for the designers of analysis platforms, ranging from low-latency architectures to intelligent sampling and aggregation methods (c.f., \cite{keim:vda}).

Our taxonomy is also somewhat sparing in its discussion of the current frontier of visual analytics research. For example, how to best incorporate (semi-)automated statistical methods within a visualization environment is a central challenge. Our discussion of derived data only scratches the surface. A related concern is the task of *data wrangling* \cite{wrangling}: reformatting, cleaning and integrating data sets so that they are amenable to visual analysis. Incorrect or improperly structured data diverts the attention and energy of trained analysts and presents a significant barrier to newcomers. As data cleaning requires nuanced human judgment (“is this outlier an error or a discovery?”), data wrangling is a necessarily interactive process combining statistical methods, visualization, and interaction techniques. This topic deserves a deeper treatment than we can provide within our compact taxonomy.

The above concerns represent active areas of research, and we expect our characterization of the field to evolve in the years to come. Validating and refining this framework is a community project that proceeds by feedback, critique and refinement. We invite the insights and commentary of the visualization, statistics, database and HCI communities, and eagerly anticipate the continued flowering of improved tools for making sense of the wealth of data that surrounds us.

REFERENCES

```
@book{card:readings}
author = {Card, Stuart K. and Mackinlay, J. and Shneiderman, B.},
title = {Readings in Information Visualization: Using Vision to Think},
pages = {712},
year = {1999},
publisher = {Morgan Kaufmann}
}
```

```
@article{heer:zoo,  
  author = {Heer, Jeffrey and Bostock, Michael and Ogievetsky, Vadim},  
  title = {A Tour through the Visualization Zoo},  
  journal = {Communications of the ACM},  
  issue_date = {June 2010},  
  volume = {53},  
  issue = {6},  
  month = {June},  
  year = {2010},  
  issn = {0001-0782},  
  pages = {59--67},  
  numpages = {9},  
  url = {http://doi.acm.org/10.1145/1743546.1743567},  
  doi = {http://doi.acm.org/10.1145/1743546.1743567},  
  acmid = {1743567},  
  publisher = {ACM},  
  address = {New York, NY, USA}  
}
```

```
@book{few:now}  
  author = {Few, Stephen},  
  title = {Now You See It: Simple Visualization Techniques for Quantitative Analysis},  
  pages = {329},  
  year = {2009},  
  publisher = {Analytics Press},  
  location = {Berkeley, CA}  
}
```

```
@book{cleveland:elements}  
  author = {Cleveland, William S.},  
  title = {The Elements of Graphing Data},  
  pages = {297},  
  year = {1994},  
  publisher = {Hobart Press}  
}
```

```
@misc{tableau,  
  author = {Tableau Software},  
  month = {March},  
  year = {2009},  
  howpublished = "\url{http://tableausoftware.com}"  
}
```

```
@book{wilkinson:gg,
```



```
author = {Wilkinson, Leland},
title = {The Grammar of Graphics (Statistics and Computing)},
year = {2005},
isbn = {0387245448},
publisher = {Springer-Verlag},
address = {Secaucus, NJ},
}
```

```
@article{wilkinson:heatmap,
author = {Wilkinson, Leland and Friendly, Michael},
title = {The History of the Cluster Heat Map},
year = {2009},
journal = {The American Statistician},
volume = {63},
number = {2},
pages = {179--184}
}
```

```
@article{bostock:pv,
author = {Bostock, Michael and Heer, Jeffrey},
title = {Protovis: A Graphical Toolkit for Visualization},
year = {2009},
journal = {IEEE Trans. Vis. and Comp. Graphics},
volume = {15},
number = {6},
pages = {1121--1128}
}
```

```
@book{wickham:ggplot2,
author = {Wickham, Hadley},
title = {ggplot2: Elegant Graphics for Data Analysis},
year = {2009},
isbn = {0387981403},
publisher = {Springer},
}
```

```
@article{stolte:polaris,
author = {Chris Stolte and Diane Tang and Pat Hanrahan},
title = {Polaris: A System for Query, Analysis, and Visualization of Multidimensional Relational Databases},
journal = {IEEE Trans. Vis. and Comp. Graphics},
year = {2002},
volume = {8},
pages = {52--65}
}
```

```
}
```

```
@inproceedings{weaver:improvise,  
  author = {Weaver,, Christopher E.},  
  title = {Building Highly-Coordinated Visualizations In {I}mprovise},  
  booktitle = {Proc. IEEE InfoVis},  
  year = {2004},  
  pages = {159--166}  
}
```

```
@article{ibm:manyeyes,  
  author = {{Vi}{e}gas, Fernanda B. and Wattenberg, Martin and van Ham, Frank and Kriss,  
Jesse and McKeon, Matt},  
  title = {{M}any {E}yes: a Site for Visualization at Internet Scale},  
  journal = {IEEE Trans. Vis. and Comp. Graphics},  
  volume = {13},  
  number = {6},  
  year = {2007},  
  pages = {1121--1128}  
}
```

```
@article{becker:trellis,  
  author = {R. A. Becker and W. S. Cleveland and M.-J. Shyu},  
  title = {The visual design and control of trellis display},  
  journal = {Journal of Computational and Graphical Statistics},  
  volume = {5},  
  number = {2},  
  year = {1996},  
  pages = {123--155},  
}
```

```
@article{hahsler:seriation,  
  author = {Michael Hahsler and Kurt Hornik and Christian Buchta},  
  title = {Getting Things in Order: An Introduction to the {R} Package seriation},  
  journal = {Journal of Statistical Software},  
  year = {2008},  
  volume = {25},  
  pages = {1--34},  
  number = {3},  
  month = {March},  
  url = {http://www.jstatsoft.org/v25/i03}  
}
```

```
@article{mackinlay:apt,
```

```
author = {Mackinlay,, Jock},
title = {Automating the Design of Graphical Presentations of Relational Information},
journal = {ACM Trans. Graph.},
volume = {5},
number = {2},
year = {1986},
issn = {0730-0301},
pages = {110--141},
doi = {http://doi.acm.org/10.1145/22949.22950},
publisher = {ACM},
address = {New York, NY},
}
```

```
@article{mackinlay:showme,
author = {Mackinlay, J. D. and Hanrahan, P. and Stolte, C.},
journal = {IEEE Trans Visualization & Comp Graphics},
number = {6},
pages = {1137--1144},
title = {Show Me: Automatic Presentation for Visual Analysis},
volume = {13},
year = {2007},
}
```

```
@inproceedings{roth:sage,
author = {Roth, Steven F. and Mattis, Joe},
title = {Automating the presentation of information},
booktitle = {Proc. IEEE Artificial Intelligence Applications},
year = {1991},
pages = {90--97},
numpages = {8}
}
```

```
@article{shneiderman:dq,
author = {Shneiderman, Ben},
title = {Dynamic queries for visual information seeking},
journal = {IEEE Software},
volume = {11},
number = {6},
pages = {70--77},
year = {1994}
}
```

```
@article{heer:voyagers,
author = {Heer, Jeffrey and Vil'egas, Fernanda B. and Wattenberg, Martin},
```

```
title = {Voyager and Voyeurs: Supporting Asynchronous Collaborative Information
Visualization},
journal = {Communications of the ACM},
volume = {52},
issue = {1},
month = {January},
year = {2009},
pages = {87--97},
numpages = {11},
url = {http://doi.acm.org/10.1145/1435417.1435439},
doi = {http://doi.acm.org/10.1145/1435417.1435439}
}
```

```
@book{tufte:visual,
author = {Edward Tufte},
title = {The Visual Display of Quantitative Information},
year = {1983},
publisher = {Graphics Press},
address = {Cheshire, CT},
}
```

```
@inproceedings{xiao:kr,
author = {Xiao, Ling and Gerth, John and Hanrahan, Pat},
title = {Enhancing Visual Analysis of Network Traffic Using a Knowledge Representation},
booktitle = {Proc. IEEE VAST},
year = {2006},
pages = {107--114}
}
```

```
@inproceedings{grossman:bubble,
author = {Grossman, Tovi and Balakrishnan, Ravin},
title = {The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's
activation area},
booktitle = {Proc. ACM CHI},
year = {2005},
pages = {281--290},
numpages = {10},
url = {http://doi.acm.org/10.1145/1054972.1055012},
doi = {http://doi.acm.org/10.1145/1054972.1055012}
}
```

```
@inproceedings{hill:deixis,
author = {Hill, William C. and Hollan, James D.},
title = {Deixis and the future of visualization excellence},
```

```
booktitle = {Proc. IEEE Visualization},
year = {1991},
pages = {314--320},
numpages = {7},
url = {http://portal.acm.org/citation.cfm?id=949607.949659}
}
```

```
@article{hochheiser:timesearcher,
author = {Hochheiser, Harry and Shneiderman, Ben},
title = {Dynamic query tools for time series data sets: Timebox widgets for interactive exploration},
journal = {Information Visualization},
volume = {3},
number = {1},
pages = {1--18},
year = {2004}
}
```

```
@inproceedings{inselberg:parallel,
author = {Inselberg, Alfred and Dimsdale, Bernard},
title = {Parallel coordinates: a tool for visualizing multi-dimensional geometry},
booktitle = {IEEE Visualization},
year = {1990},
pages = {361--378}
}
```

```
@inproceedings{kong:ink,
author = {Kong, Nicholas and Agrawala, Maneesh},
title = {Perceptual interpretation of ink annotations on line charts},
booktitle = {Proc. ACM UIST},
year = {2009},
pages = {233--236},
numpages = {4},
url = {http://doi.acm.org/10.1145/1622176.1622219},
doi = {http://doi.acm.org/10.1145/1622176.1622219}
}
```

```
@inproceedings{holz:relax,
author = {Holz, Christian and Feiner, Steven},
title = {Relaxed selection techniques for querying time-series graphs},
booktitle = {Proc. ACM UIST},
year = {2009},
pages = {213--222},
numpages = {10},
```

```
url = {http://doi.acm.org/10.1145/1622176.1622217},
doi = {http://doi.acm.org/10.1145/1622176.1622217}
}
```

```
@inproceedings{shneiderman:eyes,
author = {Shneiderman, Ben},
title = {The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations},
booktitle = {Proc. IEEE Symposium on Visual Languages},
year = {1996},
url = {http://portal.acm.org/citation.cfm?id=832277.834354}
}
```

```
@inproceedings{bederson:padpp,
author = {Bederson, Benjamin B. and Hollan, James D.},
title = {Pad++: a zooming graphical interface for exploring alternate interface physics},
booktitle = {Proc. ACM UIST},
year = {1994},
pages = {17--26},
numpages = {10},
url = {http://doi.acm.org/10.1145/192426.192435},
doi = {http://doi.acm.org/10.1145/192426.192435}
}
```

```
@inproceedings{nekrasovski:nav,
author = {Nekrasovski, Dmitry and Bodnar, Adam and McGrenere, Joanna and Guimbreti\`{e}re, Fran\c{c}ois and Munzner, Tamara},
title = {An evaluation of pan \& zoom and rubber sheet navigation with and without an overview},
booktitle = {Proc. ACM CHI},
year = {2006},
pages = {11--20},
numpages = {10},
url = {http://doi.acm.org/10.1145/1124772.1124775},
doi = {http://doi.acm.org/10.1145/1124772.1124775}
}
```

```
@article{willett:scented,
author = {Willett, Wesley and Heer, Jeffrey and Agrawala, Maneesh},
title = {Scented Widgets: Improving Navigation Cues with Embedded Visualizations},
journal = {IEEE Transactions on Visualization and Computer Graphics},
volume = {13},
issue = {6},
month = {November},
year = {2007},
```

```
pages = {1129--1136},
numpages = {8},
url = {http://dx.doi.org/10.1109/TVCG.2007.70589},
doi = {http://dx.doi.org/10.1109/TVCG.2007.70589}
}
```

```
@inproceedings{heer:doi,
author = {Heer, Jeffrey and Card, Stuart K.},
title = {DOITrees revisited: scalable, space-constrained visualization of hierarchical data},
booktitle = {Proc. Advanced Visual Interfaces},
year = {2004},
pages = {421--424},
numpages = {4},
url = {http://doi.acm.org/10.1145/989863.989941},
doi = {http://doi.acm.org/10.1145/989863.989941}
}
```

```
@inproceedings{furnas:followup,
author = {Furnas, George W.},
title = {A fisheye follow-up: further reflections on focus + context},
booktitle = {Proc. ACM CHI},
year = {2006},
pages = {999--1008},
numpages = {10},
url = {http://doi.acm.org/10.1145/1124772.1124921},
doi = {http://doi.acm.org/10.1145/1124772.1124921}
}
```

```
@article{vanham:doi,
author = {van Ham, Frank and Perer, Adam},
title = {"Search, Show Context, Expand on Demand": Supporting Large Graph Exploration with Degree-of-Interest},
journal = {IEEE Transactions on Visualization and Computer Graphics},
volume = {15},
issue = {6},
month = {November},
year = {2009},
pages = {953--960},
numpages = {8},
url = {http://dx.doi.org/10.1109/TVCG.2009.108},
doi = {http://dx.doi.org/10.1109/TVCG.2009.108}
}
```

```
@inproceedings{ahlberg:film,
```

```
author = {Ahlberg, Chris and Shneiderman, Ben},
title = {Visual Information Seeking: Tight coupling of dynamic query filters with starfield
displays},
booktitle = {Proc. ACM CHI},
pages = {313--317},
year = {1994}
}
```

```
@article{spence:bifocal,
author = {Spence, Robert and Apperley, Mark},
title = {Data Base Navigation: An Office Environment for the Professional},
journal = {Behaviour and Information Technology},
volume = {1},
number = {1},
pages = {43--54},
year = {1982}
}
```

```
@article{riche:legends,
author = {Nathalie Henry Riche and Bongshin Lee and Catherine Plaisant},
title = {Understanding Interactive Legends: A Comparative Evaluation with Standard Widgets},
journal = {Computer Graphics Forum},
volume = {29},
number = {3},
pages = {1193--1202},
year = {2010}
}
```

```
@inproceedings{zellweger:lights,
author = {Zellweger, Polle T. and Mackinlay, Jock D. and Good, Lance and Stefik, Mark and
Baudisch, Patrick},
title = {City lights: contextual views in minimal space},
booktitle = {Extended Abstracts of ACM CHI},
year = {2003},
pages = {838--839},
numpages = {2},
url = {http://doi.acm.org/10.1145/765891.766022},
doi = {http://doi.acm.org/10.1145/765891.766022}
}
```

```
@inproceedings{baudisch:halo,
author = {Baudisch, Patrick and Rosenholtz, Ruth},
title = {Halo: a technique for visualizing off-screen objects},
booktitle = {Proc. ACM CHI},
```



```
year = {2003},
pages = {481--488},
numpages = {8},
url = {http://doi.acm.org/10.1145/642611.642695},
doi = {http://doi.acm.org/10.1145/642611.642695}
}
```

```
@inproceedings{gustafson:wedge,
author = {Gustafson, Sean and Baudisch, Patrick and Gutwin, Carl and Irani, Pourang},
title = {Wedge: clutter-free visualization of off-screen locations},
booktitle = {Proc. ACM CHI},
year = {2008},
pages = {787--796},
numpages = {10},
url = {http://doi.acm.org/10.1145/1357054.1357179},
doi = {http://doi.acm.org/10.1145/1357054.1357179}
}
```

```
@article{becker:brushing,
author = {Becker, R. A. and Cleveland, W. S.},
title = {Brushing Scatterplots},
journal = {Technometrics},
volume = {29},
issue = {2},
year = {1987},
pages = {127--142}
}
```

```
@inproceedings{north:snap,
author = {North, Chris and Shneiderman, Ben},
title = {Snap-together visualization: a user interface for coordinating visualizations via relational schemata},
booktitle = {Proc. Advanced Visual Interfaces},
year = {2000},
pages = {128--135},
numpages = {8},
url = {http://doi.acm.org/10.1145/345513.345282},
doi = {http://doi.acm.org/10.1145/345513.345282}
}
```

```
@inproceedings{bell:space,
author = {Bell, Blaine A. and Feiner, Steven K.},
title = {Dynamic space management for user interfaces},
booktitle = {Proc. ACM UIST},
```

```
year = {2000},
pages = {239--248},
numpages = {10},
url = {http://doi.acm.org/10.1145/354401.354790},
doi = {http://doi.acm.org/10.1145/354401.354790}
}
```

```
@article{heer:history,
author = {Heer, Jeffrey and Mackinlay, Jock and Stolte, Chris and Agrawala, Maneesh},
title = {Graphical Histories for Visualization: Supporting Analysis, Communication, and Evaluation},
journal = {IEEE Transactions on Visualization and Computer Graphics},
volume = {14},
issue = {6},
month = {November},
year = {2008},
pages = {1189--1196},
numpages = {8},
url = {http://portal.acm.org/citation.cfm?id=1477066.1477414}
}
```

```
@article{derthick:history,
author = {Derthick, Mark and Roth, Steven F.},
title = {Enhancing Data Exploration with a Branching History of User Operations},
journal = {Knowledge Based Systems},
volume = {14},
issue = {1-2},
month = {March},
year = {2001},
pages = {65--74}
}
```

```
@article{scheidegger:prov,
author = {Scheidegger, Carlos and Koop, David and Santos, Emanuele and Vo, Huy and Callahan, Steven and Freire, Juliana and Silva, Cláudio},
title = {Tackling the Provenance Challenge one layer at a time},
journal = {Concurrency and Computation: Practice & Experience},
volume = {20},
issue = {5},
month = {April},
year = {2008},
pages = {473--483},
numpages = {11},
url = {http://portal.acm.org/citation.cfm?id=1350745.1350757}
}
```

}

```
@article{jankun:model,  
author = {Jankun-Kelly, T. J. and Ma, Kwan-Liu and Gertz, Michael},  
title = {A Model and Framework for Visualization Exploration},  
journal = {IEEE Transactions on Visualization and Computer Graphics},  
volume = {13},  
issue = {2},  
month = {March},  
year = {2007},  
pages = {357--369},  
numpages = {13},  
url = {http://dx.doi.org/10.1109/TVCG.2007.28},  
doi = {http://dx.doi.org/10.1109/TVCG.2007.28}  
}
```

```
@inproceedings{kurlander:history,  
author = {Kurlander, Davis and Feiner, Steven},  
title = {Editable Graphical Histories},  
booktitle = {Proc. IEEE Workshop on Visual Language},  
year = {1988},  
pages = {127--134}  
}
```

```
@inproceedings{danis:place,  
author = {Danis, Catalina M. and Viegas, Fernanda B. and Wattenberg, Martin and Kriss,  
Jesse},  
title = {Your place or mine?: visualization as a community component},  
booktitle = {Proc. ACM CHI},  
year = {2008},  
pages = {275--284},  
numpages = {10},  
url = {http://doi.acm.org/10.1145/1357054.1357102},  
doi = {http://doi.acm.org/10.1145/1357054.1357102}  
}
```

```
@inproceedings{isenberg:esv,  
author = {Isenberg, Petra and Tang, Anthony and Carpendale, Sheelagh},  
title = {An exploratory study of visual information analysis},  
booktitle = {Proc. ACM CHI},  
year = {2008},  
pages = {1217--1226},  
numpages = {10},  
url = {http://doi.acm.org/10.1145/1357054.1357245},
```

```
doi = {http://doi.acm.org/10.1145/1357054.1357245}
}
```

```
@inproceedings{perer:syf,
author = {Perer, Adam and Shneiderman, Ben},
title = {Systematic yet flexible discovery: guiding domain experts through exploratory data
analysis},
booktitle = {Proc. Intelligent User Interfaces},
year = {2008},
pages = {109--118},
numpages = {10},
url = {http://doi.acm.org/10.1145/1378773.1378788},
doi = {http://doi.acm.org/10.1145/1378773.1378788}
}
```

```
@article{segel:narrative,
title = {Narrative Visualization: Telling Stories with Data},
author = {Edward Segel AND Jeffrey Heer},
journal = {IEEE Trans. Visualization & Comp. Graphics},
year = {2010},
volume = {16},
number = {6},
pages = {1139--1148}
}
```

```
@inproceedings{mapreduce,
title = {MapReduce: Simplified Data Processing on Large Clusters},
author = {J. Dean and S. Ghemawat},
booktitle = {Operating Systems Design & Implementation (OSDI)},
pages = {137--150},
year = {2004}
}
```

```
@article{wrangling,
title = {Research Directions for Data Wrangling: Visualizations and Transformations for Usable
and Credible Data},
author = {S. Kandel and J. Heer and C. Plaisant and J. Kennedy and F. van Ham and N.
Henry-Riche and C. Weaver and B. Lee and D. Brodbeck and P. Buono},
journal = {Information Visualization},
year = {2011}
}
```

```
@inproceedings{keim:vda,
author = {Daniel A. Keim and Florian Mansmann and Jorn Schneidewind and Hartmut Ziegler},
```

```
title = {Challenges in Visual Data Analysis},  
booktitle = {Information Visualization},  
pages = {9--16},  
year = {2006}  
}
```