

# Interactive Dynamics for Visual Analysis

Jeffrey Heer  
Stanford University

Ben Shneiderman  
University of Maryland

December 2011

## Abstract

Visualization is touted for its capacity to leverage human visual performance, enabling users to effectively perceive patterns in data. This capacity helps users to more rapidly conduct exploratory analysis, form hypotheses, make insights, and communicate analysis findings. 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 the results. We propose a taxonomy of interactive dynamics for successful visual analytics systems. This 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 evolving this taxonomy 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 domain-specific 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 [9]. However, a single image typically provides answers to at best a handful of questions. Instead, visual analysis typically

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

Figure 1: A Taxonomy of Interactive Dynamics for Visual Analysis.

progresses in an iterative process of view creation, exploration, and refinement. Meaningful analysis consists of a dialogue: a conversation involving both people and data. Confusing widgets, complex dialog boxes, 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.

The goal of this article is to assist designers, researchers, professional analysts, procurement officers, educators and students in evaluating and creating visual analysis tools. We present a taxonomy of interactive dynamics that contribute to successful analytic dialogues. 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 evolving this taxonomy is a community project that proceeds by feedback, critique and refinement.

Our focus on interactive elements presumes a basic familiarity with visualization design. The merits and frailties of bar charts, scatter plots, 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 articles and books address these topics in great detail [9, 10, 14, 51] and we recommend them to the interested reader.

Within each branch of the taxonomy we describe example systems that exhibit useful interaction techniques. To be clear, these examples do not constitute an exhaustive survey; rather, each is intended to concretely convey the nature and diversity of interactive operations. 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 people tracking various facets of their lives, including their blood pressure, money spent, electricity used, or miles traveled.

## 2 Data & View Specification

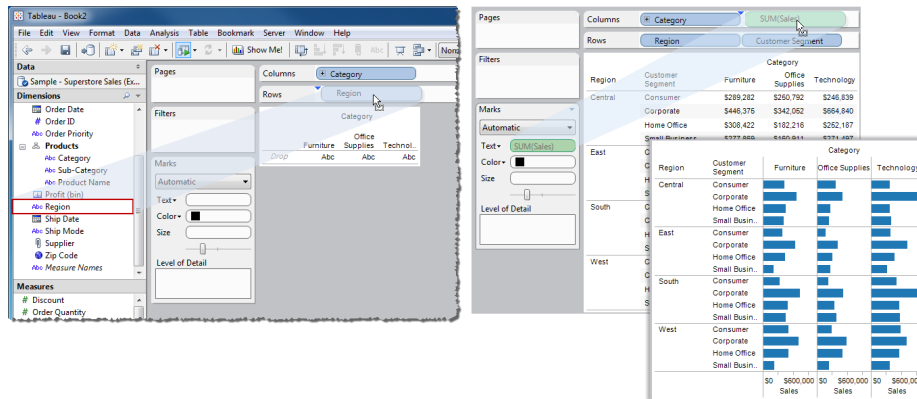
To enable analysts to explore large data sets involving varied data types (e.g., multivariate, geospatial, textual, temporal, networked), flexible visual analysis tools must provide appropriate controls for specifying the data and views of interest. These controls enable analysts to selectively **visualize** the data, to **filter** out unrelated information to focus on relevant items, and to **sort** information to expose patterns. Analysts also need to **derive** new data from the input data, such as normalized values, statistical summaries, and aggregates.

### 2.1 Visualize

Perhaps the most fundamental operation in visual analysis is to specify a visualization of data: analysts 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: users pick a chart type and assign 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 [1] and more recent platforms for artistic expression [27] 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. Analysts 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 require more input effort than chart typologies and may be limited by the set of available operators. In many cases, novel designs require analysts 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, analysts 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 [56], ggplot2 for R [55], and Protovis for HTML5 [8]. However, each of the above requires at least minimal programming ability. Tableau [48] (née Polaris [50]) provides an example of visualization specification by drag-and-drop operations: analysts place data variables on “shelves” corresponding to visual encodings such as spatial position, size, shape, and color (see Figure 2). 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 [37, 36, 43]. While both fluent and expressive, users of systems based on formal grammars need to understand the underlying generative model, imposing a steeper learning curve than the more familiar chart typology.

Fortunately, these methods are not mutually exclusive. Analysts 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

Filtering of data values is intrinsic to the visualization process, as analysts rarely visualize the entirety of a data set at once. Instead, analysts construct a variety of visualizations for selected data dimensions. Given an overview of selected dimensions, analysts then often want to shift their focus among different data subsets, for example to examine different time slices or isolate specific categories of values.

Designers have devised a variety of interaction techniques to limit the number of elements in a display. Analysts might directly select (e.g., lasso) elements in a display and then highlight or exclude them; we discuss these forms of direct view manipulation in section 3.1. Another option is to use a suite of auxiliary controls, or *dynamic query widgets* [46], for controlling element visibility (see Figures 3, 4 and 9). 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). Ordinal, quantitative and temporal data can also be filtered using a standard slider (for a single threshold value) or a range slider (for specifying multiple end 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 4 shows a range slider augmented with a histogram of underlying values.

Expert analysts 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. Filtering also interacts with other operations: filtering widgets may operate over data sorted in a user-specified manner (sec. 2.3), or users might create derived values (sec. 2.4) and filter based on the results.

## 2.3 Sort

Ordering (or sorting) is another fundamental operation within a visualization. A proper ordering can effectively surface trends and clusters of values [4, 59] or organize the data according to familiar unit of analysis (days of the weeks, financial quarters, *etc.*). The most common method of ordering is to sort records according to the value of one or more variables. Sorting controls can be simple choices in a toolbar or clicks on the header of a table to produce ascending or descending sorts for numerical or textual values. Sometimes specialized sort orders, such as weekday or month names are necessary to reveal important patterns.

Ordering becomes more complicated in the case of multiple view displays,

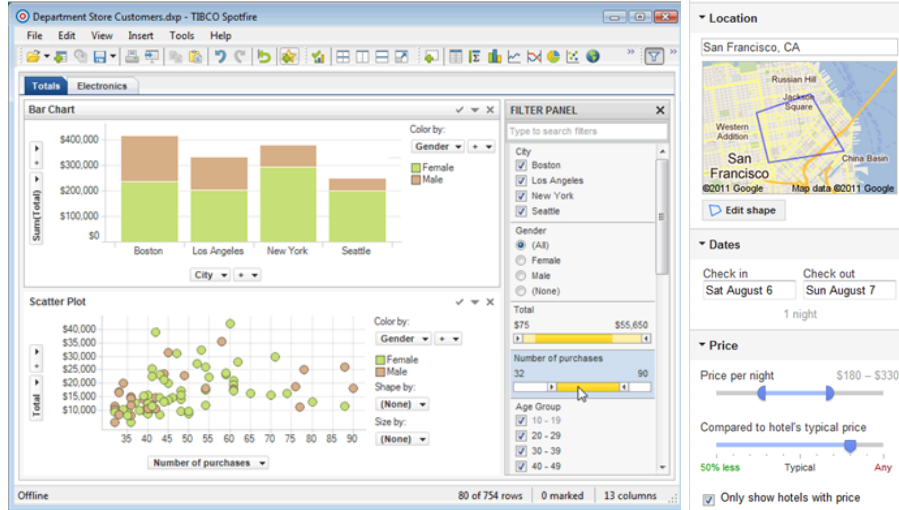


Figure 3: Examples of dynamic query filter widgets. Left: Spotfire provides a variety of controls for filtering visualized data: checkboxes and radio buttons filter categorical variables, while range sliders filter numerical values. Right: Google Hotel Search provides widgets for geographic, date, and price ranges.

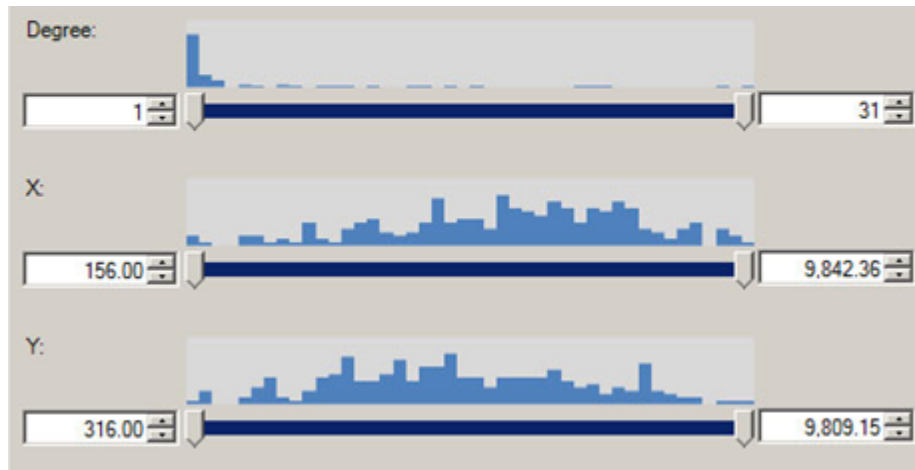


Figure 4: Histogram Sliders in NodeXL. Selection widgets can be augmented with small visualizations showing the distributions of queried data attributes.

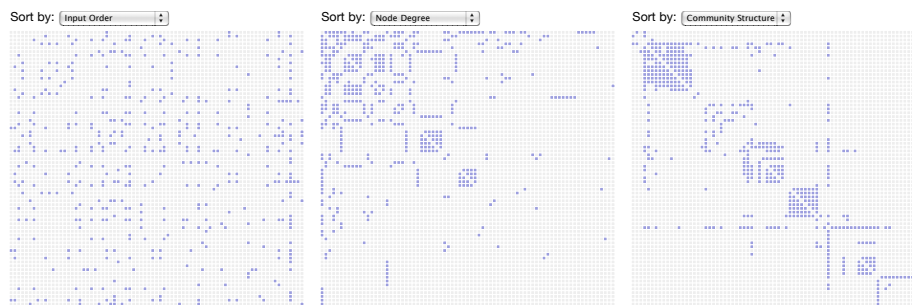


Figure 5: Reorderable Matrices. Left: A matrix plot of a social network conveys little structure when the rows and columns (representing people) are sorted alphabetically. Center: Interactively re-ordering the matrix by node degree reveals more structure. Right: Permuting the matrix by network connectivity reveals community clusters.

in which both entire plots and the values they contain may be sorted to reveal patterns or anomalies. 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 simple sorting by value. Such data may require more sophisticated *seriation* methods [18, 56, 57] that attempt to minimize a distance measure among elements. The goal is to effectively reveal underlying structure (e.g., clustering) within the data. An example is shown in Figure 5, 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.

## 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 or log transforms to enable more effective value comparisons. Derived measures are often used to summarize the input data, ranging from descriptive statistics (mean, median, variance) to model fitting (regression curves) and data transformation (group-by aggregation such as counts or summations). While analysts 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. Often this functionality is provided via a *calculation language*, similar to those found in spreadsheets or database query languages. Beyond these basic functions, hypothesis testing methods (t-tests, ANOVA) can 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 [58]) 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 may require analysts to scroll, pan, zoom and otherwise **navigate** the view to examine both high-level patterns and fine-grained details. Multiple, linked visualizations often provide clearer insights into multi-dimensional data than isolated views. Analysis tools must be able to **coordinate** multiple views so that selection and filtering operations apply to all displays at once and **organize** the resulting dashboards and work spaces.

#### 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) [24]. 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” [3] or dynamic selectors such as the bubble cursor [16], which selects the element currently closest to the mouse pointer). These selections often determine a set of objects to be manipulated, enabling highlighting, annotation, filtering, or details-on-demand. Note that interactive selection is closely related to filtering (sec. 2.2): selections can be used to identify items to remove from the display. The context of interaction must also be taken into account when choosing a selection method. For instance,



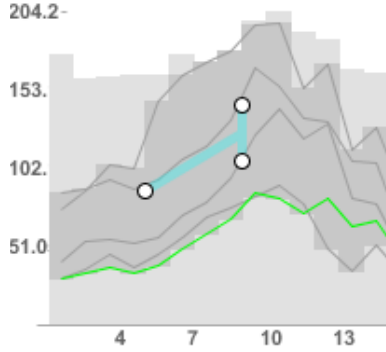


Figure 6: Querying Time-Series by Slope in TimeSearcher [25]. An angular selection tool specifies a target slope (rate of change) and tolerance for a collection of stock prices. All time series with a similar slope over the queried time range are selected; shaded regions show envelopes of minimum and maximum values. The widget operates directly on the visualization: dragging the widget from left to right interactively queries other time windows.

responding to hover events to provide details-on-demand is inappropriate when using touch-based input on a tablet or mobile phone.

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 analysts to specify higher-level criteria. A more powerful, albeit more complex, approach is to support selections as queries over the data [20]. 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 [25] (Figure 6) and attribute ranges in parallel coordinates displays [28] (Figure 7).

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 [34] (see Figure 8) or to query complex patterns of temporal variation [26]. Initial selections can also be used as a starting point for more complex selections, as analysts might click a representative object and then formulate a broader selection based on the objects properties (e.g., “select all items blue like this one”) [20]. 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.

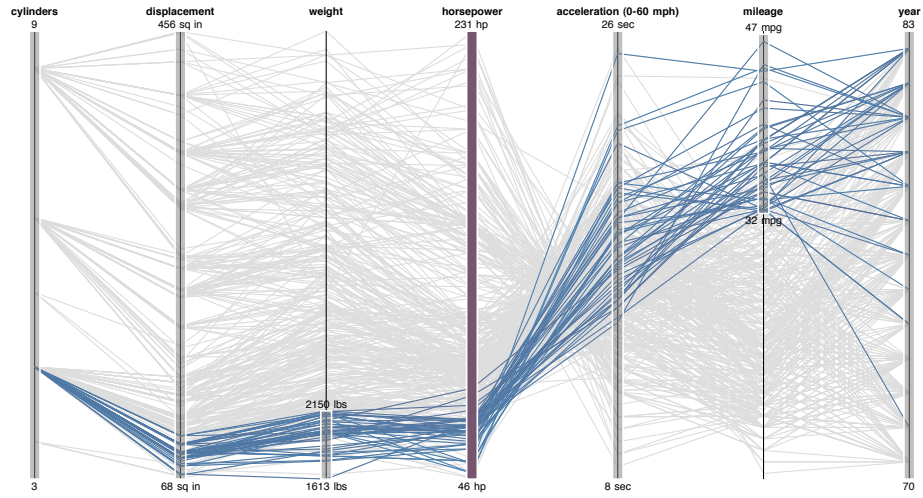


Figure 7: Selection Queries in Parallel Coordinates. Parallel coordinates plot multidimensional data as line segments among parallel axes. Here, an analyst has dragged along the axes to create interactive selections that highlight automobiles with low weight and high mileage.

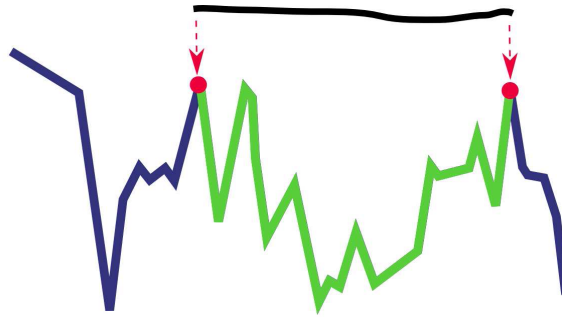


Figure 8: Perceptual Interpretation of Ink Annotations [34]. Matching input strokes (black) to a time-series segmentation model enables rapid annotation of perceptually-salient regions such as spans, peaks, and slopes.

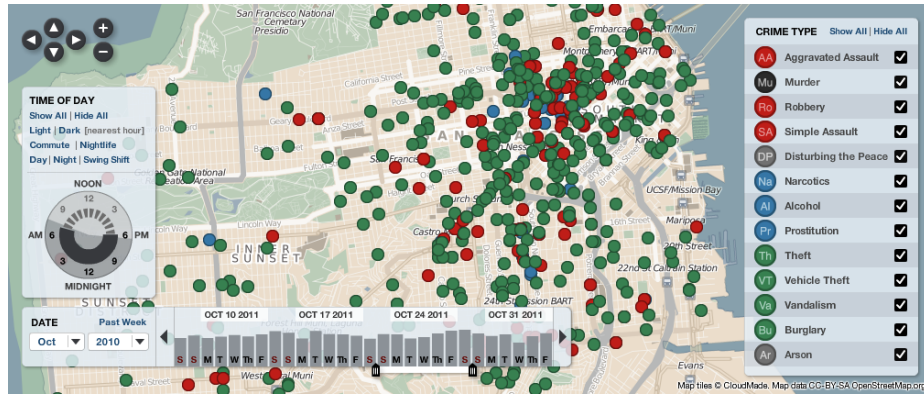


Figure 9: Zoomable Map from CrimeSpotting.org. This map depicts criminal activity by time and region, and currently shows all crimes committed after dark during the last week of October 2011. Dynamic query widgets enable filtering by time-of-day (left), date span (bottom), and type of crime (right). Pan (drag) and zoom (buttons and scroll wheel) controls enable view navigation. As an analyst zooms in on the map, the circular crime markers gain detailed labels – a form of semantic zooming.

### 3.2 Navigate

How analysts navigate a visualization is in part determined by where they start. One common pattern of navigation adheres to the widely-cited visual information seeking mantra: “Overview first, zoom and filter, then details-on-demand” [47]. 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. A common example is geographic maps: an overview might show an overall territory, followed by zooming into regions of interest (Figure 9).

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” [52].

In either case, visualizations often function as “viewports” onto an information space. Analysts need to manipulate these viewports 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 (Figure 9). Zooming need not follow a strict geometric metaphor: *semantic zooming* [6] methods can modify both the amount of information shown

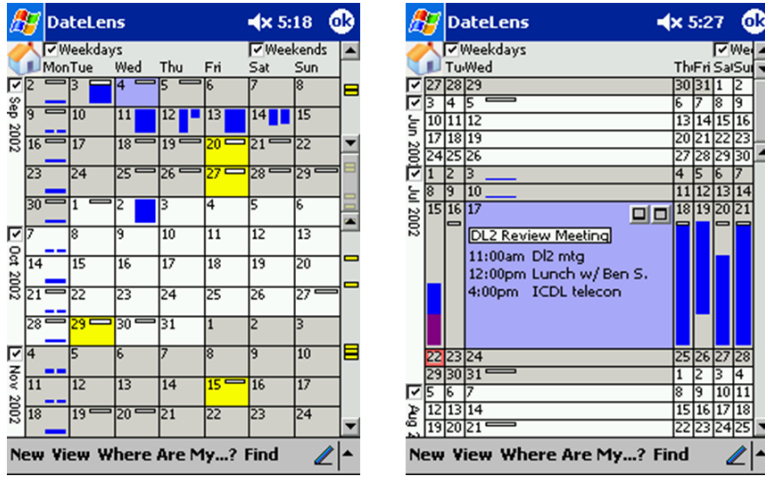


Figure 10: DateLens Focus+Context Calendar [5]. As analysts navigate from months to days to hours, the display magnifies selected regions. Semantic zooming reveals more details within focal regions.

and how it is displayed as analysts move among levels of detail (Figure 10). 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.

To further aid navigation, researchers have developed a variety of *focus plus context* methods. These “bifocal” views [49] 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 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 [41]. When larger zoom factors are needed, intermediate overviews may also 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 no significant performance improvement over simpler methods such as zooming [38].

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

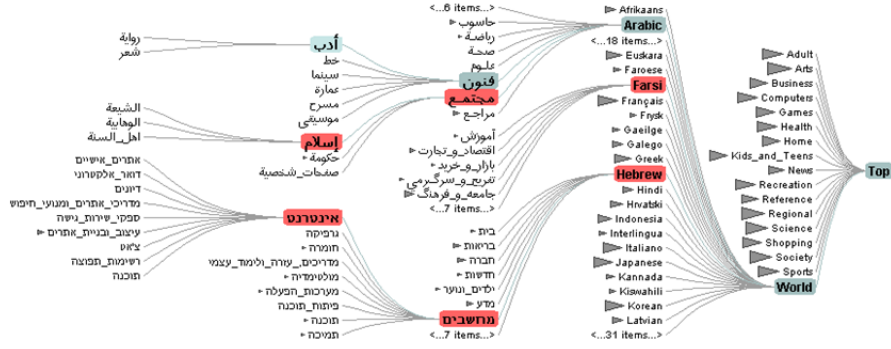


Figure 11: Degree-of-Interest Tree of a taxonomy with 600k elements [21]. A model of the analyst’s current interest filters the display to the most relevant elements. Low-interest elements are elided but still accessible through aggregate representations. The interest estimates update as an analyst explores the taxonomy, initiating animated transitions between different views of the data.

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 [15, 21, 52] 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 analysts 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 analysts’ 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 analysts, 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 [60] or the periphery of the display [2, 17].

### 3.3 Coordinate

Many analysis problems require *coordinated multiple views* that enable analysts 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

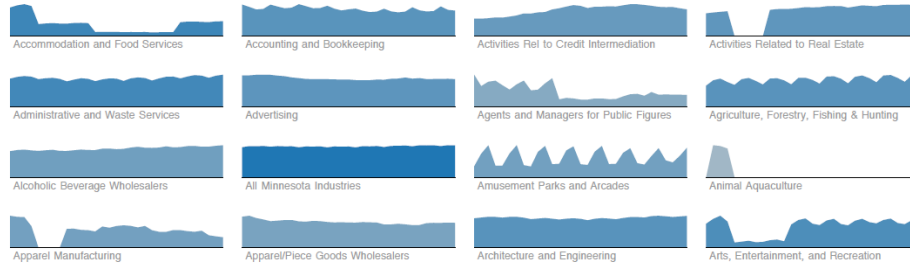


Figure 12: Small Multiples Display. The visualization shows employment figures by economic sector in the state of Minnesota. The repetition of the chart form supports comparison among sectors. Plotting all the data in one chart would otherwise clutter and obscure individual trends. Selecting a point in time in one view highlights the corresponding point in all other views.

vs. education. By selecting a single item or a group in one view, analysts might see related details or highlighted items in the other views. This powerful approach to exploring multivariate data also enables drilling down into subgroups, marking sets, and exporting selections.

Multi-view displays can facilitate comparison. For example, Tufte [51] 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 12, these small multiples, also called *trellis plots*, enable rapid comparison of different data dimensions or time-slices. Alternatively, multiple 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 (Figures 13 and 14). Though comparing multiple visualizations requires viewers to orchestrate their attention and mentally integrate patterns among views, this process is often more effective than cluttering a single visualization with too many dimensions.

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 [42].

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 [3]. Each visualization can thus serve as an input channel for revealing patterns across a data set (Figure 14). 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 [39, 54]: selecting el-



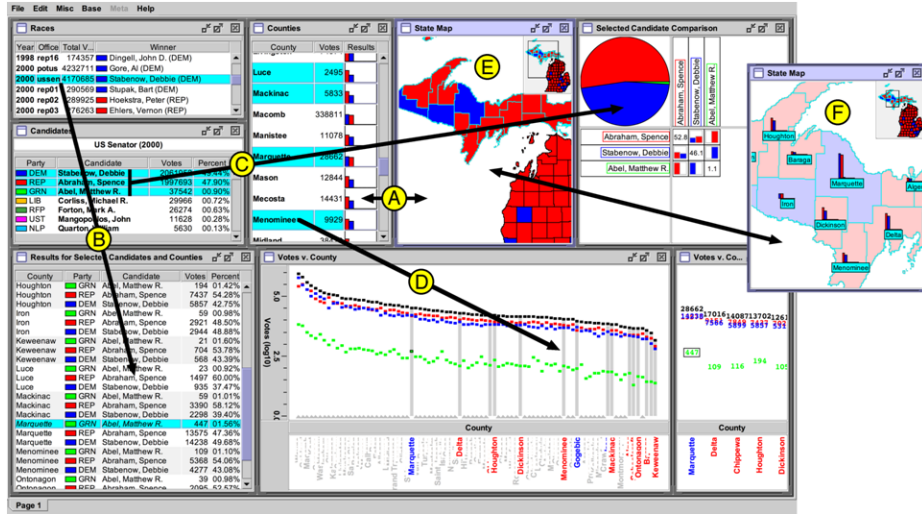


Figure 13: Multiple coordinated views in Improvise [54]. An analyst constructs a complex patchwork of interlinked tables, plots and maps to analyze the outcomes of elections in Michigan. Annotations indicate how selected data elements correspond between visualization views.



Figure 14: 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 that put-outs (middle-left plot) reveals a strong dependence on position.

ements 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.

Future studies of how analysts construct multi-view displays and specify coordination behaviors (e.g., highlighting, filtering) could 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 can 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, well-designed visual analytics tools simplify the organization of visualization views, legends, and interactive controls. For example, a human resources dataset may show a scatter plot of salary by years of experience, plus a bar chart showing 10 age groups, and a treemap with 7 corporate sites, each with 10-30 job titles. These three visualizations might give a large area for the scatter plot, with the bar chart and treemap to the right side stacked one above the other. A control panel with sliders, check boxes, radio buttons, and a search box could be on the far right, with a details-on-demand window and annotation box across the bottom. This tiled approach allows analysts with sufficiently large displays to see all the information and selectors at once, minimizing distracting scrolling or window operations, while enabling them to concentrate on extracting and reporting insights. The coordination across windows means that slider movements or check box selections will cause all views to update, allowing rapid exploration of just the employees at certain sites or specific job titles.

Typical systems allow analysts to add views, such as a second scatter plot, in ways that make modest changes to the existing window organization. An alternative approach is to add a new tab which contains the second scatter plot, so analysts can switch between the first and second set of windows. A common feature is to add trellised views, so multiple visualizations can be created at once, for example separate bar charts showing age distributions for each of the 7 corporate sites.

More advanced systems might aid this process through automated support [7, 32] that enables multiple windows to be opened/closed as a group and lays them out in orderly ways. Useful methods include standard scatter plot matrices (showing all pairs of scatter plots) or custom generation of related views of interest (e.g., of data variables correlated to the visualized attributes). Desirable



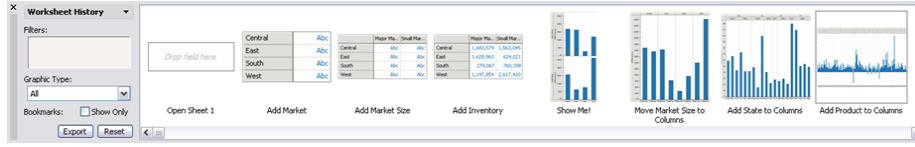


Figure 15: Visual Analysis History [22]. A “comic strip” display retraces the steps taken in a visual analysis of business operations data.

features are automatic (re)sizing as views are added or removed and layout routines to place related views in spatial proximity.

As larger and multiple displays become more common, layout organization tools will become decisive factors in creating effective user experiences. Similarly, the demand for tablet and smartphone visualizations will promote innovation in layout organizations that are compact and reconfigurable by simple gestures. Zooming, panning, flipping, and sequencing strategies will also improve analyst experiences and facilitate effective presentations to others.

## 4 Process & Provenance

Visual analytics is not limited to the generation and manipulation of visualizations – it involves a process of iterative data exploration and interpretation. As a result, visual analytics tools that provide facilities for scaffolding the analysis process will be more widely adopted. Tools should **record** analysts’ actions and insights to preserve analytic provenance and let analysis histories 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 a networked environment, analysts should be 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, provide progress indicators for experts, or lead viewers through an analysis story.

### 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 and answer. 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.

To support iterative analysis, visual analysis tools can record and visualize analysts’ *interaction histories*. At a minimum, applications should provide basic undo and redo support. While low-level input such as mouse and key-

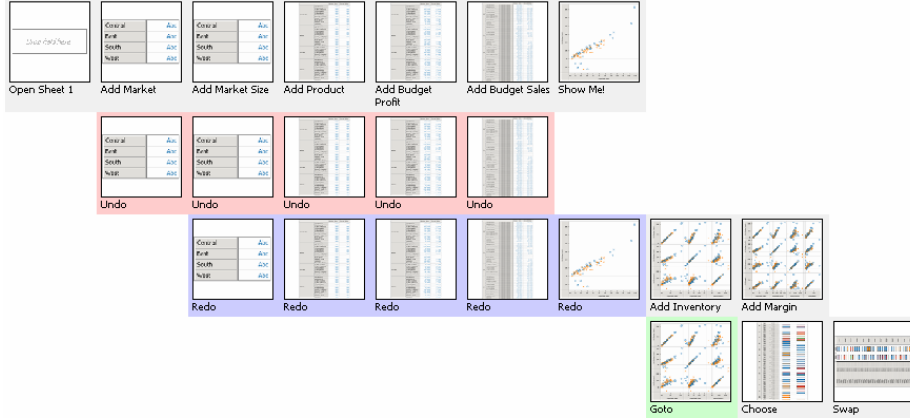


Figure 16: Behavior Graph of Branching Analysis [22]. Reading the graph in a snake-like fashion (first left-to-right, then right-to-left) reveals patterns of iterative exploration, branching, and backtracking in an analysis.

board 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 [13, 22, 30, 44]. Common visual representations of analytic actions include both chronological (“timeline”) and sequential (“comic strip”) views (Figure 15). As shown in Figure 16, visual histories also reveal the hierarchical patterns of branching histories. Techniques for “chunking” related actions together can further reduce clutter [35, 22].

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 many similar datasets to automate their efforts. 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 analysts and model analytic processes [30, 44].

## 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 is to allow textual annotation of states within a visual history. More expressive annotations are possible through direct interaction with the view, using the selection techniques discussed in sec. 3.1.

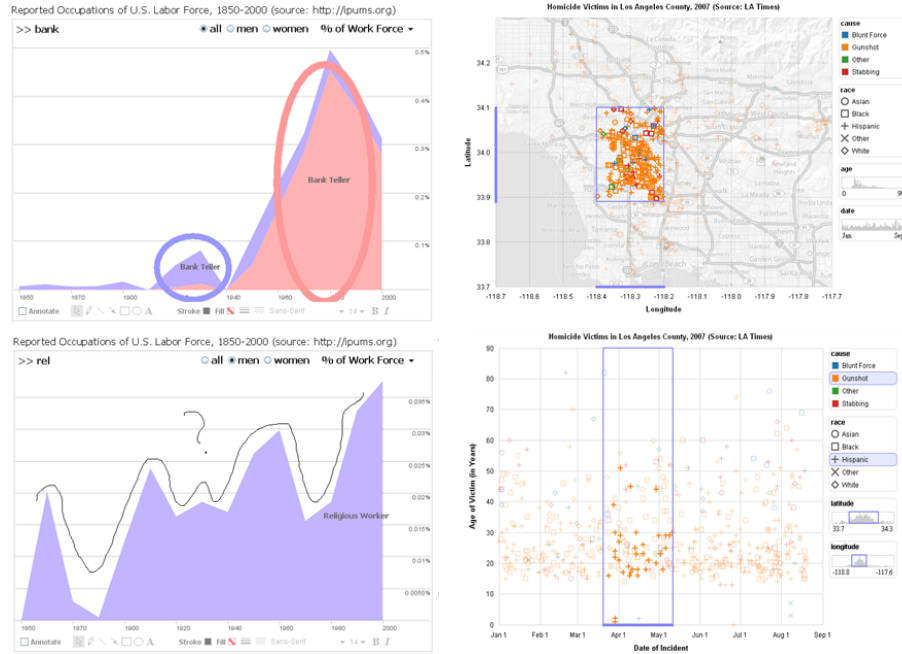


Figure 17: Annotated Visualizations. Left: Freeform graphical annotations of occupational data. The top annotation highlights a gender reversal among bank tellers using color-coded ellipses; the bottom annotation expresses confusion regarding the erratic percentage of religious workers. Right: Data-aware 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.

Analysts may wish to “point” to specific elements or regions within a visualization and associate these annotations with explanatory text or links to other views [23].

Freeform *graphical annotations* provide one expressive form of pointing [23], shown on the left side of Figure 17. 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 an explicit tie to the underlying data. Freeform annotations implemented as vector graphics 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 aggregation.

Annotations can be made data-aware when realized as selections (see the right side of Figure 17). These selections can be represented by a set of selected elements, a declarative query, or both [20]. 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 analysts 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 [23, 53]. 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 18 shows one example of a collaborative visual analysis tool incorporating view sharing, annotation and discussion.

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 [23, 53]. Unlike a static screenshot, bookmarking enables analysts to take up an exploration where their collaborators left off. 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 IBMs Many-Eyes [53] can be used to embed visualization applets in external web sites. Publishing is particularly important for reaching larger audiences. However, while publishing is a necessary condition for broad sharing, it may not be sufficient by itself for engaging viewers [19]. Visualizations embedded within a blog or discussion forum can reach an established audience and may foster discussion more effectively than a centralized site [11].

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 [19, 23, 29].

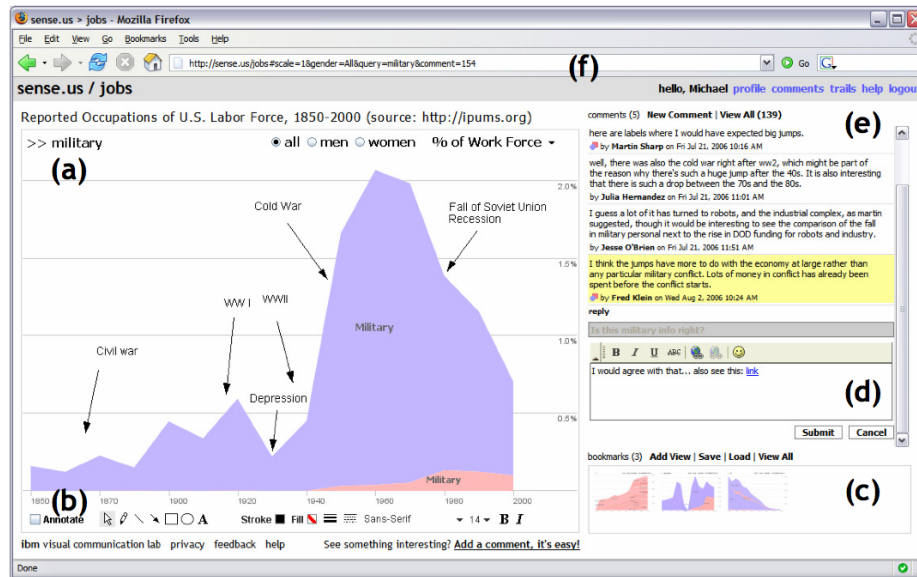


Figure 18: Collaborative Visual Analysis in Sense.us [23]. Counter-clockwise: (a) An interactive visualization, with a graphical annotation for the currently selected comment. The visualization is a stacked time-series visualization of the U.S. labor force, broken down by gender. Here the percentage 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.

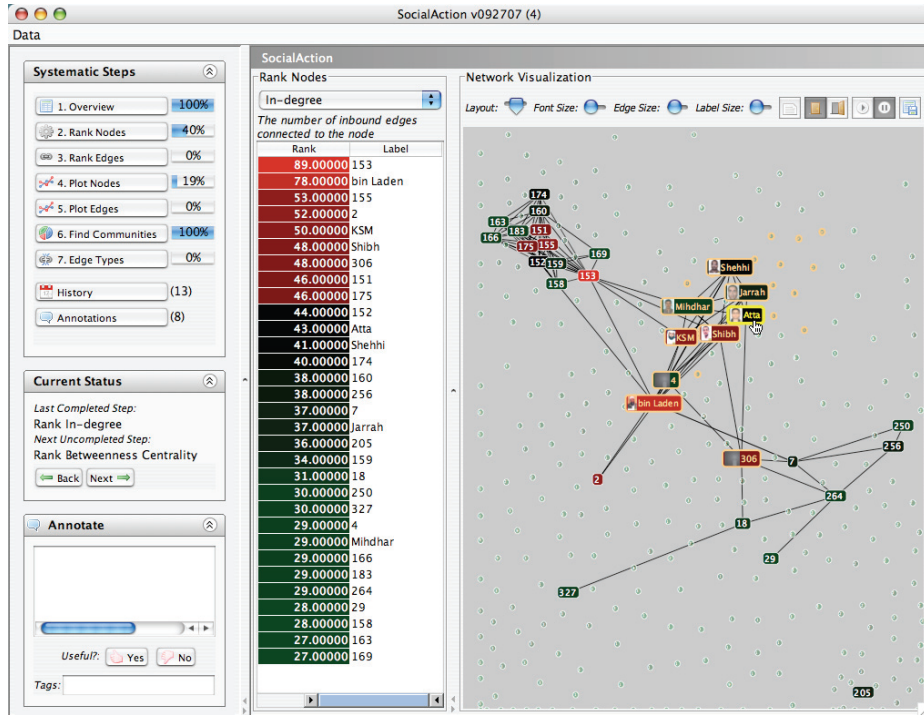


Figure 19: Systematic yet flexible analysis in SocialAction [40]. The panel on the left suggests common steps to structure social network analysis and provides progress indicators.

## 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 analysts through workflows for common tasks.

Some processes are clearly linear, but many visual analytics tasks require richer *systematic yet flexible* processes that allow analysts to take excursions while keeping track of what they have done. For example, SocialAction [40] 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 19). 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

Published: February 2, 2010

## 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.

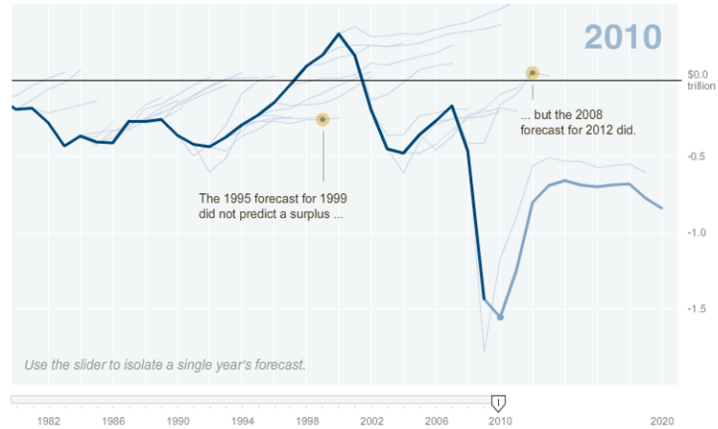


Figure 20: Data storytelling by the New York Times. Through staging and annotations, this interactive graphic guides the reader through decades of budget predictions while also permitting interactive exploration.

expert analysts to author reusable workflows.

In recent years, journalists have been experimenting with different forms of *narrative visualization* [45] 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 20). 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 provide 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 an abstracted picture of a domain, taxonomies may be 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, especially for large data sets. While popular platforms for large data analysis such as MapReduce [12] 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., [33]).

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* [31]: 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 based on domain knowledge (“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 evolving this framework is a community project that can proceed through feedback, critique and refinement by visual analytics researchers and practitioners. 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.

## 6 Acknowledgements

We thank our colleagues and students for providing valuable comments on drafts: Maneesh Agrawala, Jason Chuang, Cody Dunne, John Guerra-Gomez,



Pat Hanrahan, Sean Kandel, Diana MacLean, and Kostas Pantazos. This work was partially supported by National Science Foundation grants IIS-0968521, IIS-1017745, CCF-0964173 and SBE-0915645, NIH-National Cancer Institute grant RC1-CA147489, and an ONC-SHARP grant on Cognitive Information Design and Visualization.

## 7 Bios

**Jeffrey Heer** (<http://vis.stanford.edu/jheer>) is an Assistant Professor of Computer Science at Stanford University. The visualization tools developed by his lab are used by researchers, corporations, and thousands of data enthusiasts around the world. In 2009, Jeff was named to MIT Technology Review's TR35. He holds B.S., M.S., and Ph.D. degrees in Computer Science from the University of California, Berkeley.

**Ben Shneiderman** (<http://www.cs.umd.edu/~ben>) is a Professor in the Department of Computer Science and Founding Director (1983-2000) of the Human-Computer Interaction Laboratory at the University of Maryland. He is a Fellow of the AAAS, ACM, and IEEE, and a Member of the National Academy of Engineering. Ben is the co-author with Catherine Plaisant of *Designing the User Interface: Strategies for Effective Human-Computer Interaction* (5th ed., 2010).

## References

- [1] Greg Abram and Lloyd Treinish. An extended data-flow architecture for data analysis and visualization. In *Proc. IEEE Visualization*, pages 263–270, 1995.
- [2] Patrick Baudisch and Ruth Rosenholtz. Halo: a technique for visualizing off-screen objects. In *Proc. ACM CHI*, pages 481–488, 2003.
- [3] R. A. Becker and W. S. Cleveland. Brushing scatterplots. *Technometrics*, 29:127–142, 1987.
- [4] R. A. Becker, W. S. Cleveland, and M.-J. Shyu. The visual design and control of trellis display. *Journal of Computational and Graphical Statistics*, 5(2):123–155, 1996.
- [5] Benjamin B. Bederson, Aaron Clamage, Mary P. Czerwinski, and George G. Robertson. Datelens: A fisheye calendar interface for pdas. *ACM Transactions on Computer-Human Interaction*, 11:90–119, March 2004.
- [6] Benjamin B. Bederson and James D. Hollan. Pad++: a zooming graphical interface for exploring alternate interface physics. In *Proc. ACM UIST*, pages 17–26, 1994.

- [7] Blaine A. Bell and Steven K. Feiner. Dynamic space management for user interfaces. In *Proc. ACM UIST*, pages 239–248, 2000.
- [8] Michael Bostock and Jeffrey Heer. Protovis: A graphical toolkit for visualization. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1121–1128, 2009.
- [9] Stuart K. Card, J. Mackinlay, and B. Shneiderman. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann, 1999.
- [10] William S. Cleveland. *The Elements of Graphing Data*. Hobart Press, 1994.
- [11] Catalina M. Danis, Fernanda B. Viegas, Martin Wattenberg, and Jesse Kriss. Your place or mine?: visualization as a community component. In *Proc. ACM CHI*, pages 275–284, 2008.
- [12] J. Dean and S. Ghemawat. MapReduce: Simplified data processing on large clusters. In *Proc. Operating Systems Design & Implementation (OSDI)*, pages 137–150, 2004.
- [13] Mark Derthick and Steven F. Roth. Enhancing data exploration with a branching history of user operations. *Knowledge Based Systems*, 14:65–74, March 2001.
- [14] Stephen Few. *Now You See It: Simple Visualization Techniques for Quantitative Analysis*. Analytics Press, 2009.
- [15] George W. Furnas. A fisheye follow-up: further reflections on focus + context. In *Proc. ACM CHI*, pages 999–1008, 2006.
- [16] Tovi Grossman and Ravin Balakrishnan. The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor’s activation area. In *Proc. ACM CHI*, pages 281–290, 2005.
- [17] Sean Gustafson, Patrick Baudisch, Carl Gutwin, and Pourang Irani. Wedge: clutter-free visualization of off-screen locations. In *Proc. ACM CHI*, pages 787–796, 2008.
- [18] Michael Hahsler, Kurt Hornik, and Christian Buchta. Getting things in order: An introduction to the R package seriation. *Journal of Statistical Software*, 25(3):1–34, March 2008.
- [19] Jeffrey Heer and Maneesh Agrawala. Design considerations for collaborative visual analytics. *Information Visualization*, 7:49–62, 2008.
- [20] Jeffrey Heer, Maneesh Agrawala, and Wesley Willett. Generalized selection via interactive query relaxation. In *Proc. ACM CHI*, pages 959–968, 2008.
- [21] Jeffrey Heer and Stuart K. Card. Doitrees revisited: scalable, space-constrained visualization of hierarchical data. In *Proc. Advanced Visual Interfaces*, pages 421–424, 2004.

- [22] Jeffrey Heer, Jock Mackinlay, Chris Stolte, and Maneesh Agrawala. Graphical histories for visualization: Supporting analysis, communication, and evaluation. *IEEE Transactions on Visualization and Computer Graphics*, 14:1189–1196, November 2008.
- [23] Jeffrey Heer, Fernanda B. Viégas, and Martin Wattenberg. Voyager and voyeurs: Supporting asynchronous collaborative information visualization. *Communications of the ACM*, 52:87–97, January 2009.
- [24] William C. Hill and James D. Hollan. Deixis and the future of visualization excellence. In *Proc. IEEE Visualization*, pages 314–320, 1991.
- [25] Harry Hochheiser and Ben Shneiderman. Dynamic query tools for time series data sets: Timebox widgets for interactive exploration. *Information Visualization*, 3(1):1–18, 2004.
- [26] Christian Holz and Steven Feiner. Relaxed selection techniques for querying time-series graphs. In *Proc. ACM UIST*, pages 213–222, 2009.
- [27] Bestiario Impure. <http://www.impure.com/>.
- [28] Alfred Inselberg and Bernard Dimsdale. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Proc. IEEE Visualization*, pages 361–378, 1990.
- [29] Petra Isenberg, Anthony Tang, and Sheelagh Carpendale. An exploratory study of visual information analysis. In *Proc. ACM CHI*, pages 1217–1226, 2008.
- [30] T. J. Jankun-Kelly, Kwan-Liu Ma, and Michael Gertz. A model and framework for visualization exploration. *IEEE Transactions on Visualization and Computer Graphics*, 13:357–369, March 2007.
- [31] S. Kandel, J. Heer, C. Plaisant, J. Kennedy, F. van Ham, N. Henry-Riche, C. Weaver, B. Lee, D. Brodbeck, and P. Buono. Research directions for data wrangling: Visualizations and transformations for usable and credible data. *Information Visualization*, 10(4):271–288, 2011.
- [32] Eser Kandogan and Ben Shneiderman. Elastic windows: evaluation of multi-window operations. In *Proc. ACM CHI*, pages 250–257, 1997.
- [33] Daniel A. Keim, Florian Mansmann, JornSchneidewind, and Hartmut Ziegler. Challenges in visual data analysis. In *Information Visualization*, pages 9–16, 2006.
- [34] Nicholas Kong and Maneesh Agrawala. Perceptual interpretation of ink annotations on line charts. In *Proc. ACM UIST*, pages 233–236, 2009.
- [35] Davis Kurlander and Steven Feiner. Editable graphical histories. In *Proc. IEEE Workshop on Visual Language*, pages 127–134, 1988.

- [36] J. D. Mackinlay, P. Hanrahan, and C. Stolte. Show me: Automatic presentation for visual analysis. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1137–1144, 2007.
- [37] Jock Mackinlay. Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics*, 5(2):110–141, 1986.
- [38] Dmitry Nekrasovski, Adam Bodnar, Joanna McGrenere, François Guimbretière, and Tamara Munzner. An evaluation of pan & zoom and rubber sheet navigation with and without an overview. In *Proc. ACM CHI*, pages 11–20, 2006.
- [39] Chris North and Ben Shneiderman. Snap-together visualization: a user interface for coordinating visualizations via relational schemata. In *Proc. Advanced Visual Interfaces*, pages 128–135, 2000.
- [40] Adam Perer and Ben Shneiderman. Systematic yet flexible discovery: guiding domain experts through exploratory data analysis. In *Proc. Intelligent User Interfaces*, pages 109–118, 2008.
- [41] C. Plaisant, C. Carr, and B. Shneiderman. Image-browser taxonomy and guidelines for designers. *IEEE Software*, 12(2):21–32, March 1995.
- [42] Nathalie Henry Riche, Bongshin Lee, and Catherine Plaisant. Understanding interactive legends: A comparative evaluation with standard widgets. *Computer Graphics Forum*, 29(3):1193–1202, 2010.
- [43] Steven F. Roth and Joe Mattis. Automating the presentation of information. In *Proc. IEEE Artificial Intelligence Applications*, pages 90–97, 1991.
- [44] Carlos Scheidegger, David Koop, Emanuele Santos, Huy Vo, Steven Callahan, Juliana Freire, and Cláudio Silva. Tackling the provenance challenge one layer at a time. *Concurrency and Computation: Practice & Experience*, 20:473–483, April 2008.
- [45] Edward Segel and Jeffrey Heer. Narrative visualization: Telling stories with data. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1139–1148, 2010.
- [46] Ben Shneiderman. Dynamic queries for visual information seeking. *IEEE Software*, 11(6):70–77, 1994.
- [47] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proc. IEEE Symposium on Visual Languages*, 1996.
- [48] Tableau Software. <http://tableausoftware.com>.

- [49] Robert Spence and Mark Apperley. Data base navigation: An office environment for the professional. *Behaviour and Information Technology*, 1(1):43–54, 1982.
- [50] Chris Stolte, Diane Tang, and Pat Hanrahan. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. *IEEE Transactions on Visualization and Computer Graphics*, 8:52–65, 2002.
- [51] Edward Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, 1983.
- [52] Frank van Ham and Adam Perer. “Search, show context, expand on demand”: Supporting large graph exploration with degree-of-interest. *IEEE Transactions on Visualization and Computer Graphics*, 15:953–960, 2009.
- [53] Fernanda B. Viégas, Martin Wattenberg, Frank van Ham, Jesse Kriss, and Matt McKeon. Many Eyes: a site for visualization at internet scale. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1121–1128, 2007.
- [54] Christopher E. Weaver. Building highly-coordinated visualizations in Improve. In *Proc. IEEE InfoVis*, pages 159–166, 2004.
- [55] Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer, 2009.
- [56] Leland Wilkinson. *The Grammar of Graphics (Statistics and Computing)*. Springer-Verlag, Secaucus, NJ, 2005.
- [57] Leland Wilkinson and Michael Friendly. The history of the cluster heat map. *The American Statistician*, 63(2):179–184, 2009.
- [58] Ling Xiao, John Gerth, and Pat Hanrahan. Enhancing visual analysis of network traffic using a knowledge representation. In *Proc. IEEE VAST*, pages 107–114, 2006.
- [59] Ji Soo Yi, Youn ah Kang, John Stasko, and Julie Jacko. Toward a deeper understanding of the role of interaction in information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13:1224–1231, 2007.
- [60] Polle T. Zellweger, Jock D. Mackinlay, Lance Good, Mark Stefik, and Patrick Baudisch. City lights: contextual views in minimal space. In *Extended Abstracts of ACM CHI*, pages 838–839, 2003.