A New Visual Language for Incremental Generation of Visual Representations

Hanseung Lee[1]

Abstract— As the user base of visualization solutions expands, it is now even more critical to help end-users visualize their data quickly and effectively. Current automated visualization techniques use only a small set of basic guidelines mostly based on perceptual principles. Yet, visualization design principles and guidelines, as discussed in literature, cover a broad spectrum of factors that guide effective visualization design. In our previous research, we priorly analyzed existing principles and guidelines and examined factors that influence visualization design and found several limitations that the current visual principles include. To overcome these limitations, we propose a semi-automatic visualization approach that uses a declarative language for specifying visual representations and interactions with data. In this approach, visualization design principles and guidelines are expressed as rules that can encode human perception, information organization, data characteristics and semantics, and display characteristics, based on our examination of visualization literature and practice. By applying rules iteratively, a dataset is incrementally transformed into an interactive visualization via the application of data and visual operators. Intermediate representations of the specification, along with applicable rules, can be made shown to users to allow them choose the one that suits them best, which they can further adjust with minimal effort. Rules are easily extensible to customize and conform to domain-specific standards and practices. As a generative and language-based technique our approach offers flexibility not only in terms of design rules applied but also in the variety of visualization produced.

Index Terms— Visual language, automatic visualization system, visualization principles and guidelines

1 Introduction

Data visualization is rapidly becoming a commodity in many applications, media, platforms, and domains, and is consumed by an ever-expanding user base, which includes users of varying degrees of data and visualization expertise, from human-resource staff to marketing experts, to general public consuming news on TV and print media. On the one hand this is good news, as visualization is now becoming mainstream for decision-making and communication, but it poses new challenges. One challenge is to educate public on effective data visualization techniques, which is to some degree being met by the proliferation of books on data visualization, covering principles from human perception to visual design. A complimentary challenge is to design tools to effectively support data visualization design and make this process relatively easy for a broad range of users. The challenge here is to incorporate as much of the guidelines and design know-how into (semi-) automatic techniques such that outcome comes close to professional designs.

Current automated visualization techniques are typically heuristic and use only a small set of basic guidelines, mostly based on perceptual principles with the goal of matching the data to a possible set of visualizations. As such the problem is defined as a matching problem not as an optimization problem that incorporates a large number of factors. There are several questions: What are the factors that influence effective design? How do they interact? What are the trade-offs in terms of supporting user tasks and desired insight?

In our previous research, we have priory conducted a grounded theory study to examine the language and content of data visualization principles and guidelines. In sum, we extracted guidelines from books, papers, and blogs on data visualization and examined factors and relationships among them that influence effective visualization design. From this analysis, we found a variety of factors, including data domain and attribute semantics, user perception, user tasks, insight type, and display characteristics, among many other factors, and also found a variety of interactions between these factors including causal, contextual, and logical relationships. More importantly, by observing these languages on data visualization principles and guidelines, we aim to observe what factors are currently covered and also what limitations exist. One big limitation is that the current visual language cannot express complex set of visualization principles (more details in Appendix A).

To remedy this limitation, we propose a new visual language, VizGo[2], a systematic visual language for incremental generation of visual representations of data. By incrementally generating visual representations, we expect that VizGo overcomes the complexity issues of the current visualization principles. In addition, by adding the concept of codified rules, which allows domain customization, visualizations can be more utilized for data analysis in several domains.

In this paper, we first review the visualization principles / theories and the existing (semi-) automatic visualization systems. Then, we propose a new visualization language, VizGo, and illustrate its algebra and characteristics. The main goal of this research is to develop a (semi-) automatic visualization system based on our new language. Therefore, the proposed system overview for automating visualization using VizGO is described later and internal case studies using our system will be also illustrated.

2 Related Work

2.1. Principles and Guidelines

Many visualization principles and models have been proposed to guide the generation of useful visualizations. One of the most frequently used principles is Shneiderman's Visual Information-Seeking Mantra, "Overview first, zoom and filter, then details-on-demand", describing a recipe for efficient design steps for users to transform the data to useful visualization [1]. While this mantra was validated by several [2][3][4], others advocated incorporating a holistic design methodology [5].

[1] Hanseung Lee is with University of Maryland, College Park Department of Computer Science.
E-mail: hanseung@cs.umd.edu
This document was last revised on 12/1/2014.

[2] VizGo comes from terms "Visualization" and "LEGO." This term is used to express that our visual language is generated incrementally by building and combining smaller pieces of visual objects, which is similar to how LEGO works.

Card suggested a visualization reference model, which enables users to represent abstract data, map data to visualizations, and add interactions [6][7][8]. Chi presented a data state reference model, which includes an operator and user interaction model to guide operational steps of different visualization types [9][10]. Carr analyzed different areas of information visualization based on Shneiderman's abstract user-task taxonomy and then suggested seven general guidelines for designing information seeking applications [1][11]. Beyond generally applicable principles, individual differences, such as user's cognitive abilities, were found to be an important factor in visualization effectiveness [12].

Munzner proposed a nested model for visualization design and validation [13], consisting of four layers, which are 1) to characterize the task and data in the vocabulary of the problem domain, 2) to abstract into operations and data types, 3) to design visual encoding and interaction techniques, and 4) to create algorithms to execute techniques efficiently.

Clearly, many researchers have proposed principles and guidelines to develop effective visualizations. Each focuses on different factors, such as data, interaction, users, tasks, and domain, and has different strengths and weaknesses. However, there is a lack of studies to connect these guidelines together along with the context of use, especially when applied to automatic visualization systems. Dias et al. analyzed visualization rules using grounded theory techniques over five different parameters: data type, task type, scalability, dimensionality, and positioning of attributes and characterized several visualizations based on these parameters [14]. While similar in method our present study goes much more, covering not only broader set of concepts but also deeper in regards to details of concepts and their interactions.

More recently, Kindlmann and Scheidegger [64] proposed an algebraic basis for designing data visualizations. When the mappings from data (D) to data representation (R) to visualization (V) exist, they defined an algebraic denotation α and ω and its relationships. Here, α denotes the changes of data (D) and ω denotes the changes of visualization (V). Finally, by observing and analyzing (α, ω) pairs, they proposed three visualization design principles--invariance, unambiuguity, and correspondence. This work has strengths since it aimed for generality by describing visualization design in algebraic terms. Different from their analysis on (α, ω) pairs, our design principles will focus more on how to define the algebraic terms on transforming data and map data to visualizations (D-R-V) constructively to create a generic flow of automatic visualization.

2.2. Theory and Taxonomy

Many studied how to classify different information visualizations and proposed taxonomies, focusing on different subsets such as data, visualization, tasks, and interactions.

2.2.1. Visualization Theory and Taxonomy

Card and Mackinlay proposed a data-oriented taxonomy to develop a framework for designing effective visualizations for different types of data [6]. Keller identified and classified data types such as scalar, nominal, direction, shape, and position [15]. Keim's classification of data consists of six types: 1-d, 2-d, multi-dimensional, text and hypertext, hierarchies and graphs, and algorithm and software [16].

Task is another important factor in determining effective visualizations and therefore it was included in several taxonomies. Shneiderman proposed data-type-by-task taxonomy, which considered user tasks such as Overview, Zoom, Filter, Details-on-Demand, Related, History, and Extract and data types such as 1-d, 2-d, 3-d, temporal, multidimensional, tree, and network [1]. Keller also classified user tasks to nine categories: identify, locate, distinguish, categorize, cluster, rank, compare, associate, and correlate [15]. Finally, they used their data type classification and task classification together to help users determine effective visualization techniques. Recently, Brehmer and Munzner proposed a more complex way to categorize visualizations to reduce the gap between low-level and high-level tasks in a taxonomy [17]. They categorized user tasks based on three questions: why and how users perform a task, and what inputs/outputs it includes.

Model-based taxonomies generally try to capture the visualization categories not just by individual features but by considering the entire process of visualizations from data to user tasks. Tory and Moeller proposed a high-level taxonomy based on a design model and performed classification by determining whether an attribute is discrete or continuous and how many design attributes (e.g., color, transparency, etc.) were selected [19].

Finally, evaluations are also likely to help in design by allowing designers think about evaluation in advance, considering the questions, variables, tasks, and subjects. Andrews categorized evaluation methods into four based on their purposes: exploratory, predictive, summative, and formative [20]. More recently, Lam et al. analyzed empirical studies in infovis and classified them based on study goals and types of research questions [21].

2.2.2. Interaction Theory and Taxonomy

A. Definition of interaction in information visualization

Let us first find the definition of interaction used in HCI. In Becker et al.'s paper [47], interaction is simply defined as direct manipulation and instantaneous change. In another HCI study, Foley et al. [48] defined that interaction is a technique to perform a generic task via physical input/output device. According to Dix et al. [49], interaction is "the communication between user and the system."

By extending these definitions, several researchers have also defined interaction in the perspective of information visualization. According to Ward and Yang [50], interaction in information visualization can be defined as "a mechanism for modifying what the users see and how they see it." Lam [51] also defined interaction as a user action that causes a visible change in the information visualization system. Another work from Yi et al. [52] stated that interaction in infovis is the features of user abilities to directly or indirectly interpret and manipulate visual representation.

B. Why interaction is important?

As we observed above, interaction is the users' ability to manipulate visual representations of data in infovis. This is one of the important components in most information visualization systems for various reasons. According to Kosara et al. [53], interaction is essential in an infovis system since interaction techniques provide the ability for users to explore, analyze, and represent data. However, when users perform visual tasks, the amount of data flowing from the system-side to user-side is a lot more than the amount of data flowing from the user-side to the system-side [54]. Therefore, this gap should be reduced for a better visual analytic process, and for this reason, more meaningful and appropriate uses of interaction techniques are essential in information visualization systems.

C. Taxonomies and frameworks of interaction in infovis

To better understand interaction techniques in infovis, reviewing previous research on how to classify and categorize interaction is important. In particular, we aim to develop a visual language that contains interaction operators. In this case, the interaction taxonomy and framework can provide the evidence of how to define interaction operators/operands for automatic visualization systems. Chi and Riedl [9] already emphasized the importance of the interaction framework. They argued that the interaction framework can help end-users and system designers reuse operators, understand view/value separation, and reduce gulf of execution [63]. Therefore, in this section, we review different types of taxonomies and frameworks of interactions used in the field of information visualization.

Several previous researches have proposed taxonomies of interaction in information visualization. These taxonomies can be distinguished in three groups based on which features the taxonomy focuses on. We think there are three features that taxonomies can be classified to, which are low-level techniques, system/data, and users.

First, many previous studies proposed taxonomies of interactions by collecting and clustering interaction techniques. We call these classifications are based on low-level interaction techniques. One famous classification of low-level interaction techniques was proposed by Chuah and Roth [55]. According to their basic visualization interaction (BVI) operations, interaction techniques can be classified in three groups: graphical operations, set operations, and data operations. Graphical operations indicate the interaction techniques that encode data, set graphics value, and manipulate objects. Set operations include the techniques to create, delete, and summarize sets. Finally, data operations focused on data manipulations such as add, delete, derived attributes, and so on. Keim categorized user interactions into five classes consisting of dynamic projection, interactive filtering, interactive zooming, interactive distortion, and interactive linking/brushing [16]. Buja et al. [56] also developed the three groups of interactions and classified the interaction techniques into three groups: focusing (projection, zoom, pan), linking (brushing, sectioning, database query), and arranging (scatter plot matrix and conditional plot). According to Keim's classification [57], low-level interaction techniques can be classified as interactive filtering, zooming, distortion, linking, brushing, and dynamic projections. We can also find the categorization of interactions in Wilkinson's studies [26]. In the Grammar of Graphics [26], interaction techniques are classified into seven categories. The seven categories are navigating (zooming, panning, lens), manipulating (reordering, dragging), brushing and linking (shapes, logic), filtering (categorical, continuous, multiple, fast), rotating, transforming (specification, display, assembly, tap), and animating.

With more high-level aspects, several studies proposed taxonomies focused on system/data or users. System/data-centric taxonomies categorize interactions by 1) observing how interaction techniques are systematically different from each other or 2) how interactions are determined from the data or how interactions manipulate the data. For example, Tweedie [58] provided interaction dimensions such as interaction types and directness. Interaction types include manual, mechanized, instructable, steerable, and automatic types and directness indicates direct or indirect manipulation. By combining these two dimensions, Tweedie proposed the classification of interactions. In addition, Spence [59] classified interactions according to its mode such as continuous, passive, composite, or stepped interaction. Ward and Yang's definition [50] proposed a systematic approach of the definition of interaction techniques. They proposed three dimensions of interaction operations: interaction operators, interaction spaces, and interaction parameters. Interaction operators are the low-level techniques such as navigation, selection, and distortion. Interaction spaces include different types of contexts where the interaction techniques can be projected such as screen-space, data value-space, data structure-space, attribute-space, object-space, and visualization structure-space. The final dimension, interaction parameters, features the variation of interaction techniques such as focus, transformation, extents, and blender. Finally, the classification of interactions can be systematically determined by choosing one of the features in these three dimensions.

Different from the previous two taxonomy sets, other studies focused on users who perform the interactions and classified the taxonomy according to their tasks. Zhou and Feiner [60] classified interaction into two groups using users' tasks, which are relational visual tasks and direct visual organizing and encoding tasks. Relational visual tasks include interaction techniques such as associate, categorize, compare, correlate, cluster, background, distinguish, emphasize, identify, locate, rank, reveal, generalize, and switch. Any other encoding interaction techniques are classified as direct visual organizing and encoding tasks. Interactions also play an important role. Gotz and Zhou observed 20 different user interaction and characterized them into three categorical abstractions based on user intents: exploration actions, insight actions, and meta actions [18].

More recently, Yi et al. [52] developed seven categories of interaction based on user intents. Those seven categories are select, explore, reconfigure, encode, abstract/elaborate, filter, and connect.

As we reviewed, many taxonomies of interaction techniques have been proposed based on different properties. However, to bridge the gap between interaction techniques and the visualization system, it is also important to find the framework on how to model those interactions in the system. However, there has been limited research in on this in the field of information visualization. Chi and Riedl [9] developed an operator framework based on the state model that includes visualization stages from raw data to visualization views. They classified interaction operators into these stages so that interaction can be defined according to the end-user's analysis process. Similar to this approach, Jankun-Kelly et al. [61] proposed the P-set model to describe users' data exploration process with interaction techniques. In addition, another work from Elmqvist et al. [62] suggested the new concept of interaction, which is fluid interaction. They defined a fluid interaction as the best interaction for visualization. By developing an operational definition of fluid interaction, they aimed to develop a framework to find and evaluate the most appropriate interactions in different visualization systems. Lam [51] also proposed a framework to measure the cost of interaction. Here, an interaction cost illustrates when and how users face difficulties during the interaction tasks. He suggested a framework of interaction costs inspired by Norman's Seven Stages of Action [63] and reviewed the previous infovis researches according to his new model.

2.3. Automatic Visualization Techniques

Choosing an appropriate visualization for a dataset is challenging. Generally, previous approaches tackle this automatic visualization problem by constructing and formalizing general rules of visualizations. Bertin’s pioneering work described properties and characteristics of graphical system [22], which were later extended by Mackinlay and formed the foundation of automatic visualization by defining algebraic operators [23]. Roth et al. created SAGE that performs automatic visualization that extends Mackinlay's work to support a wider range of displays [24]. These automatic visualization approaches are based on interpreting data properties and mapping them to visualization recommendations. Similarly, Stolte et al. developed operators using relational algebra [25] and Wills and Wilkinson used statistical analysis to find operators and then, map them to proper visualization [26][27].

While these approaches were more data driven, others focused on automatic visualization techniques using task-analytic approach [28]. It uses the previous visual task as a priori knowledge to generate proper visualizations. Another work presented a new approach to generate visualizations based on user behaviour [29]. By monitoring user's analytic behaviour patterns, the system builds a behaviour model that predicts user's next action. By mapping the predicted action to a visualization type, the system can automatically recommend the next visualization.

More recently, Voigt et al. proposed a way to use semantic information of data for automating visualizations [30]. They first constructed visualization ontology (VISO) to share and formalize semantic knowledge of web data. Based on this VISO concept, they developed a knowledge-assisted and context-aware visualization recommendation algorithm.

Besides such techniques or algorithms for automatic graphic generation, several notable automatic visualization systems for end-users have been also developed in the previous research. Zhou et al. leveraged machine learning techniques and proposed an automated graphics generation system called IMPROVISE*, which models a rich representation of information graphics with both data objects and visual objects [31]. IMPROVISE* use seven visual features and sixteen data features to represent the data structure, visual structure, and data-visual mapping. This work focuses on finding and retrieving design rules on predefined visualization types such as pie chart, bar chart, and scatter plot. Other foundational researches also led to the development of products that uses automatic visualization techniques such as Tableau [32][33] and Spotfire [34].

3 New Visual Language

This paper proposes a new visual language, VizGo, a systematic visual language for incremental generation of visual representations of data. The goal of this new language is overcoming the limitations of current visualization principles. To find the general patterns and limitations of the existing visual languages, we have first conducted a grounded theory study and examined the language and content of data visualization principles and guidelines. From this previous research, our findings suggested that automatic or semi-automatic visualization approaches need to handle not only broader set of factors but also complex interaction between these factors. Furthermore, several factors such as data domain and semantics are likely to gain even more importance, necessitating flexibility in terms the specification of design know-how as input to such systems. More details on our previous study are explained in Appendix A. VizGo is developed by reflecting those suggestions from our previous analysis. In this section, we introduce VizGo and illustrate how our language can remedy the observed limitations.

3.1. VizGo

From the results of grounded theory analysis, we found three big gaps between current visual guidelines and further research directions in visualization (see more details in Appendix A). Our research aims to develop a new visual language to overcome these issues, and thereby the new visual language will be able to automatically create more meaningful visualizations.

```
{
        “type”:
        “props”: {
        }
        “children”: [
              {
                      …
                      “children” : []
              }
        ]
}
```

Figure 1. Formal systematic definition of VizGo language

VizGo aims to automatically create visualizations using a bottom-up approach. In a high-level view, this approach starts from mapping data to basic visual elements, and in each step, chooses a design rule and an associated operator on how to group and aggregate visual data. From this approach, VizGo can support changes in parts of the visualization as well as support changes in the entire visualization itself. In addition, a snapshot of the visualization is available in every step so that users can check their intermediate visualization results while constructing the visual language.

To support a bottom-up concept of our language, we define codified rules in VizGo. Codified rules are a set of regulations on how to update and modify the visual language. In other words, codified rules have the ability to generate visualization specifications, and as a result it can support expressiveness and effectiveness of visualizations. Therefore, in each step, a codified rule is applied to the current visual language, so that it updates to a new visual language. Some examples of codified rules can be 1) map geographic references to longitudes and latitudes, 2) map circular attributes to radial axes, 3) group data with nominal or ordinal attributes, and 4) sort visual groups that are grouped by ordinal attributes.

Note that our visual language is abstracted. It is an intermediate representation of the actual visualization. It is expressed without specifying the low-level details of the visualization. In fact, it acts like a bridge between high-level concepts (e.g., bar chart, line graph, and scatter plot) and low-level details (e.g., specifics of scales). When analysts need to explore the data in a short time, they are interested in looking at visual patterns or trends and therefore details are unnecessary. Details such as specifying scales can be handled as a post-processing stage when actually rendering the visualization.

We define visual elements as the smallest fundamental piece for visualization. A visual element has two properties. One property of the visual element is called visual primitive, which is point, line, polygon, or text. Each of these visual primitives can be seen as they are orthogonal to each other so they have their own properties. The other property is visual cues, which is based on [22][23]: position, length, angle, shape, size, and color. In each visualization, there is at least one basic visual element that has a visual primitive type with its visual cues.

VizGo language expresses a visualization using a systematic language format as shown in Figure 1. As shown in this format, the VizGo language is consisted with three parts; type, props, and children. First, the “type” field defines if it is a complete visualization or it represents an intermediate step during the visualization process. The “props” field indicates the properties of the visual algebra, which are applied to the current visual language. Examples of the argument of “props” can be “sortBy” and “groupBy” to express which data variable it uses to sort or group by. As explained before, the VizGo language is generated incrementally by applying the codified rule in each step. When a rule is applied, the language is updated. The “children” component in the VizGo language is used to express how a new visual component group is formatted by aggregating previous visual components. For example, when the visual components are aggregated, the previous visual language is placed in the children part, and the aggregation rules are stated in the new language. By applying the rules multiple times, the language can potentially express the final visualization with multiple children hierarchically.

Figure 2 illustrates how VizGo language enables a system constructively to generate the visualization from raw data using bottom-up approach.

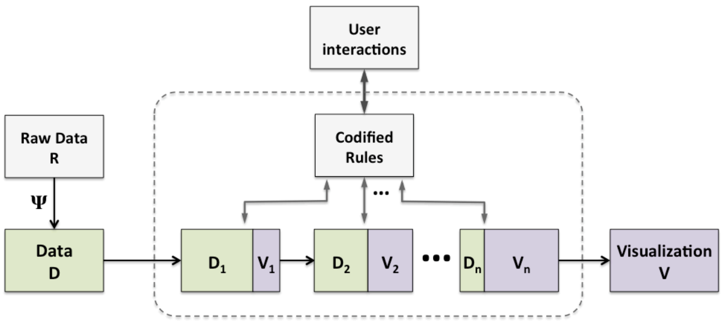


Figure 2. Flow of transforming data to visualization using VizGo: Raw data R is transformed to a dataset D. Through several constructive transforming steps, data is gradually mapped to a visualization V. In each construction step, a codified rule is applied to the data and create/update the visualization. Users can also update the codified rules during the process.

3.2. Operators

Our language is also expressed in an abstract form using our visual language algebra. To develop this algebra, we extended and formalized the algebra in Mackinlay and Wilkinson's earlier work [23][26][27]. VizGo contains operators such as data transformation & manipulation, encode, data organization, visual aggregation, and visual interactions.

- **Data Transformation and Manipulation (ψ):** We can transform certain attributes to create more attributes as additional data. For example, “state”, which is a nominal data, can be transformed into “longitude and latitude” values, which are interval data. In addition, we can manipulate (e.g., filter) data to match the needs.

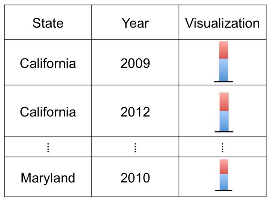
- **Encode or Map (μ):** The first step to start visualizing data is to create a visual primitive such as point, line, polygon, or text. In this step, we can encode any certain type of attribute (e.g., nominal, ordinal, interval, and ratio) to a visual primitive associated with a visual cue. Examples of visual cues are position, length, angle, shape, size, and color.

• **Data Organization (G, ↑, ↓):** After encoding, we need to organize the data. With the grouping operator (i.e., G) we can group the data with a specific attribute. Optionally, to create an ordered list, we can sort (i.e., ↑ for ascending order, ↓ for descending order) the data with a certain attribute.

• **Visual Aggregation (Σ):** After we organize the data, now we can aggregate visual primitives using a bottom-up approach. First, we can create a visual group that contains multiple visual primitives. Next, this process is recursively done to create a visual group that is constructed of other visual groups, hierarchically. We defined three different visual aggregation methods, concatenating, overlaying, and projecting. For example, we can concatenate bars either horizontally or vertically to create a stacked bar charts, overlay several line graphs on top of each other to visualize more information, and project charts on top of each state on a map to visualize geometrically. Finally, we can create the final visualization if all the groups have a visual aggregation method defined.

• **Interactions (i):** All the attributes that are not used in the visualization are open to the user so they could interactively select or filter the data. For example, if the attribute "US States" is not covered in the visualization, possibly because there are many states that will create a cluttered visualization, users can interactively select certain states they have interest in to visualize.

Figure 3 explains how VizGo language generates the visual language expression both in a systematic form and in an algebraic form. Figure 3(a) shows example visualization, where population of each state is represented with a "bar" primitive by encoding gender to the bar color. This visualization can be expressed by systematic visual language as shown in Figure 3(b) and it is also possible to represent with an algebraic form shown in Figure 3(c).



State	Year	Visualization
California	2009	
California	2012	
⋮	⋮	⋮
Maryland	2010	

```
{
        "type": "dataset",
         "props": {
                "groupBy": "YEAR"
                "sortBY": "GENDER"
        },
         "children": [
                "type": "data
                "visual": {
                        "type": "bar",
                        "props": {
                            "color": "GENDER"
                            "length": "POPULATION"
                        }
                }
        ]
}
```

(a) Visualization (b) Visual Language

$$\sum_{\substack{\uparrow gender\\ G(year)}} \mu_{color}(gender)\mu_{length}(population)$$

(c) Visual Language in an algebraic form

Figure 3. Visual language expression via VizGo: (a) example visualization, (b) VizGo expression in systematic language, and (c) VizGo expression in an algebraic form

3.3. Codified Rules

Codified rules are set of regulations on how to update and modify VizGo. Initially, VizGo starts with a set of data attributes and it is gradually constructed to create the final visualization. The automatic visualization system based on VizGo contains a set of pre-defined rules, which is consisted with basic visualization principles. Users can modify rules as well as add new rules and delete unnecessary rules according to their needs. From this domain customization, the codified rules can extend to cover more semantically guided design Figure 4 illustrates the format of codified rules in VizGo language. As shown in here, codified rule consists of two components: match and apply principles.

• **"match" part**
The system searches for matching parts in the data. If any fraction of the VizGo language matches the "match" part in the codified rule, the associated "apply" part is applied and updates the visual language.
Match part can contain one to several arguments, which of each argument is consisted with description, ref, and path. In sum, by finding matches in the VizGo language, it can decide whether the current rule should be applied or not.

• **"apply" part**
If a fraction of the VizGo language matches the "match" part in the codified rule, the associated rules, which are defined in "apply" part, is used to update the visual language.

```
{
        "description":
        "match": [
                {
                        "description": $1,
                        "ref": ,
                        "path":
                },
                {
                        "description": $2,
                        "ref": ,
                        "path":
                },
        ],
         "apply": [
                "description":
                "mixin":{
                },
                "add": {
                        ...
                         "replace":{
                         }
                },
                "scope": "{$visual}"
        ]
}
```

Figure 4. Formal systematic definition of codified rule in VizGo

In a high-level view, "apply" part can have "replace," "add," and "mixin" components. The "replace" is used when we want to change the values of any components in the visual language. The "add" is used when we want to add/concatenate new values with preserving the current value. The "mixin" is used to perform the previous two functions together. In the "mixin" part, if any arguments already have a value, it change the value to the new one such as in "replace", but for the arguments that do not exist, those are newly added to the rule such as in "add".
Figure 5 shows an example of codified rule and Figure 6 illustrates how this rule is applied to VizGo language.

```
{
        "type": "dataset",
        "props": {
                "groupBy": "STATE"
        },
        "children": [
                {
                        "type": "dataset",
                        "props": {
                                "groupBy": "YEAR"
                                "sortBy": "GENDER"
                        },
                        "children": [
                                {
                                        "type": "data",
                                        "omitted data information due to space"
                                        "visual": {
                                                "type": "line",
                                                "props": {
                                                     "color": "GENDER"
                                                     "length": "POPULATION"
                                                }
                                        }
                                }
                        ]
                 }
        ]
}
```

(a)

```
{
        "type": "dataset",
        "props": {
                "groupBy": "STATE"
                "sortBy": "YEAR"
        },
        "children": [
                {
                        "type": "dataset",
                        "props": {
                                "groupBy": "YEAR"
                                "sortBy": "GENDER"
                        },
                        "children": [
                                {
                                        "type": "data",
                                        (omitted data information due to space)
                                        "visual": {
                                                "type": "line",
                                                "props": {
                                                     "color": "GENDER"
                                                     "length": "POPULATION"
                                                }
                                        }
                                }
                        ],
                        "visual": {
                                "props": {
                                        "label": "YEAR"
                                }
                        }
                }
        ]
}
```

(b)

Figure 6. An example on how rules are applied to the VizGo language; (a) VizGo language prior to applying the rule in Figure 5. (b) VizGo language after applying the rule in Figure 5. The text in the boxes are the parts that are added by applying the codified rule in Figure 5.

```
{
        "description": "sort visual groups that are grouped by ordinal"
        "match": {
                "$$1" : {
                        "type": "dataset",
                        "props" : {
                                "groupBy": "$$id"
                        },
                        "visual": "undefined"
                },
                "$$2": {
                        "type": "attribute",
                        "id": "$$id",
                        "props": {
                                "ordinal": true
                        }
                }
        },
        "apply": [
                {
                        "mixin": {
                                "props": {
                                        "sortBy": "$$id"
                                }
                        },
                        "target": "../$$1"
                 },
               {
                        "mixin": {
                                "visual": {
                                        "props": {
                                                "label": "$$id"
                                        }
                                }
                        },
                        "target": "$$1"
                }
        ]
}
```

Figure 5. An example of a codified rule, which sorts visual groups that are grouped by ordinal attributes.

One of our focused considerations is that codified rules should be designed to incorporate semantic information to achieve a more appropriate visualization. For example, if there is gender data type, "male" is usually represented by a blue color and the "female" type is colored by red/pink color. By understanding and incorporating this semantic information, the system can generate a more user-friendly and understandable visualizations automatically.

With an example, we will go through how the rule is applied to VizGo. Figure 6(a) shows the VizGo language prior to applying the rule in Figure 5. The VizGo language in Figure 6(a) is currently grouped by "year" but not yet sorted by "year." Since "year" is an ordinal attribute and is already grouped, it is a good design choice to also sort it by "year." We can see that the match component contains two matches, $$1 and $$2. Interpreting these two match statements, it is trying to find a dataset that is grouped by an ordinal attribute. The validation component is validating that a visual component is defined in the children of the dataset that is grouped. Since both the match and validate components are satisfied, the system now updates the VizGo language using the apply component. Figure 6(b) shows the VizGo language after applying the rule in Figure 3. As a result, we can confirm that the new VizGo langauge is sorted by "year" and a visual component is also added.

4 Automatic Visualization System with VizGo

The eventual goal of our visual language is to develop a novel automatic visualization system, which can generate different visualizations by interpreting data structure and semantics automatically. Based on our visualization language and grammar, VizGo, we can define a generic flow for automatically creating visualizations.

Figure 7 illustrates the system overview of the automatic visualization system. In a high-level view, the automatic visualization engine takes data, semantics, and metadata as inputs. Then, it applies codified rules to create a VizGo visual language. From these visual languages, the system renders visualizations. Finally, it ranks and displays the top ranked visualizations.

More specifically, Figure 8 shows the generic flow for automatically generating visualization with an example. In the automatic visualization algorithm, it first receives the raw data itself. As shown in Figure 8(a), the system receives the raw data as a table. Then, we analyze the data semantics and metadata associated with the raw data. From this information, we decide to either transform or manipulate the data so that it makes more sense to the user. For example, if it contains geo-related data (e.g., states and cities) then users might prefer it on a map rather than a bar chart. Next, we select visual primitives and map the attributes to an appropriate visual cue. As shown in Figure 8(b), we mapped the population information to a length and the gender information to a color. Then, we organize the data so that it contains groups hierarchically and the elements in each group can be sorted in ascending or descending order. For each group, we aggregate visual elements. For example, as in Figure 8(c), the gender visual group is created by stacking visual elements vertically while the year and state visual group is concatenated horizontally. After performing aggregation on all the visual groups we created, we end up with the final visualization. In the example, two different final visualization products are created automatically. The left visualization of Figure 8(d) is generated by sorting and aggregating visual groups with "state." Since there are many states, this visualization it is not that effective. To solve this scalability issue, the right visualization of Figure 8(d) selected "state" as an interactive attribute. Here, the interface provides an interaction method for users to click on the checkboxes to visualize the state visual groups. The system will automatically create many different types of visualizations. Finally, the system ranks the visualizations and shows the top few to the user for further interaction.

As the system follows this flow, we expect it to automatically generate potentially useful visualizations from raw data. Here, users can intervene in the flow process and make visual decisions at important junctions.

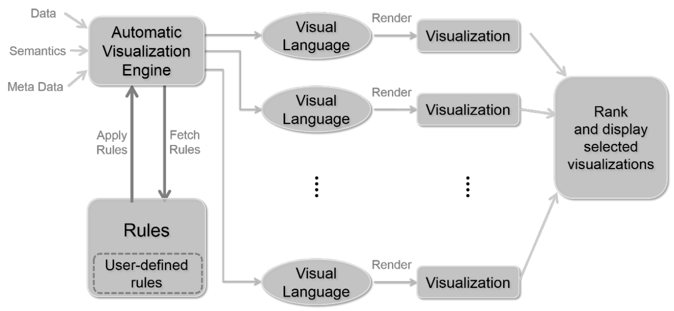
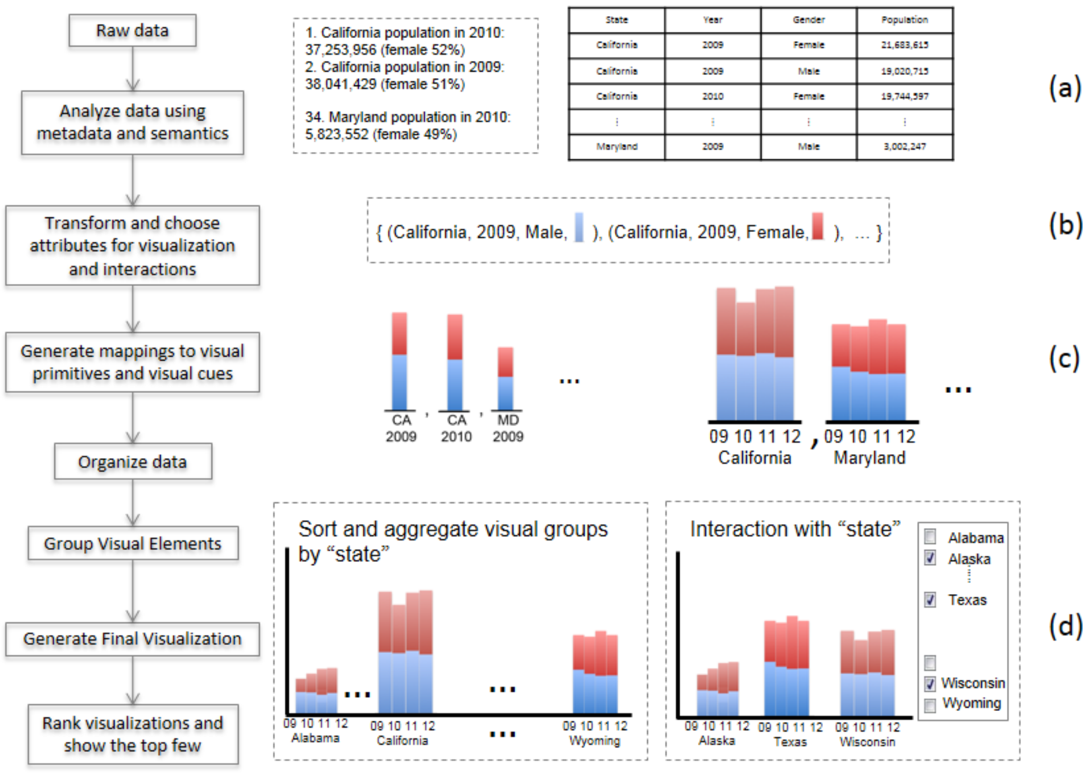


Figure 7. The system overview of the automatic visualization system using VizGo.



1. California population in 2010: 37,253,956 (female 52%)
2. California population in 2009: 38,041,429 (female 51%)

34. Maryland population in 2010: 5,823,552 (female 49%)

State	Year	Gender	Population
California	2009	Female	21,683,615
California	2009	Male	19,020,715
California	2010	Female	19,744,597
⋮	⋮	⋮	⋮
Maryland	2009	Male	3,002,247

{ (California, 2009, Male, 🟦), (California, 2009, Female, 🟥), ... }

Figure 8. The generic flow of our automatic visualization system using VizGo language: (Left) a flow for creating visualizations automatically, (Right) a step-by-step example of constructively created automatic visualization with the US population data.

5 Internal Case Study

In this section, we explain how VizGo generates visualizations incrementally via example. In this example, we use the data, which contains the population information in US by state, year, and gender. Here, we illustrate seven steps to generate the final visualization using VizGo. In each step, specific visual operators and rules are applied to the visual grammar, and so this grammar gradually constructs the final visualization expression. In other words, by showing individual steps in this example, we will show how the final grammar is incrementally generated from the data.

5.1. Example 1

Step 1. First we have the raw data table as the following.

State	Year	Gender	Population
California	2009	Female	21,683,615
California	2009	Male	19,020,715
California	2010	Female	19,744,597
⋮	⋮	⋮	⋮
Maryland	2010	Female	3,002,247

As the table shown above, our example data contains the population data in US by state, year, and gender. Then, for each column in the data table, the metadata and semantics should be analyzed. The metadata can be determined as the following: State (Nominal, Geo), Year (Interval, Time), Gender (Male maps to blue color, Female maps to red color), and Population (Ratio).

Step 2. Create visual primitive and encode “population” to bar length

$$\mu_{length}(population)$$

Here, “bar” is chosen as a visual primitive, and “population” is encoded to length of each bar.

State	Year	Gender	Visualization
California	2009	Female	▮
California	2009	Male	▮
California	2010	Female	▮
⋮	⋮	⋮	⋮
Maryland	2010	Female	▮

3. Encode “gender” to bar color

$$\mu_{color}(gender)\mu_{length}(population)$$

Next, gender is encoded by color. As we interpreted in step 1, gender’s metadata is determined by red and blue colors. Therefore, visual primitives with female gender are encoded to red and male to blue in this step.

State	Year	Visualization
California	2009	🟥
California	2009	🟦
California	2010	🟥
⋮	⋮	⋮
Maryland	2010	🟥

Step 4. Group by “year”, sort and aggregate visual primitives by “gender”

$$\sum_{\substack{\uparrow gender\\ G(year)}} \mu_{color}(gender)\mu_{length}(population)$$

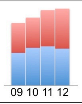
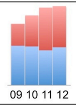
Visual primitives are grouped by “year.” Since two primitives are integrated, we need to determine how to aggregate visual primitives. Here, the two visual primitives are stacked vertically. The vertical sorting order is determined by “gender.”

State	Year	Visualization
California	2009	🟥🟦
California	2012	🟥🟦
⋮	⋮	⋮
Maryland	2010	🟥🟦

Step 5. Group by “state”, sort and aggregate visual groups by “year”

$$\sum_{\substack{\uparrow year\\ G(state)}} \sum_{\substack{\uparrow gender\\ G(year)}} \mu_{color}(gender)\mu_{length}(population)$$

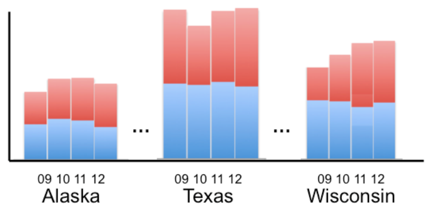
The visualizations are integrated by “state” horizontally and each visual primitive is sorted by “year.”

State	Visualization
California	
⋮	⋮
Maryland	

Step 6. Sort and aggregate visual groups by “state”

$$\sum_{\uparrow state} \sum_{\substack{\uparrow year\\ G(state)}} \sum_{\substack{\uparrow gender\\ G(year)}} \mu_{color}(gender)\mu_{length}(population)$$

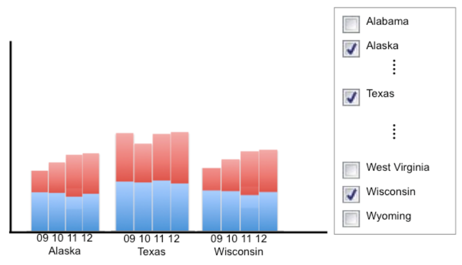
Finally, by integrating all visualizations via “state,” we can get the final visualization.



Step 7. Interaction with “state”

$$i_{state} \sum_{\uparrow state} \sum_{\substack{\uparrow year\\ G(state)}} \sum_{\substack{\uparrow gender\\ G(year)}} \mu_{color}(gender)\mu_{length}(population)$$

Since there are so many states, we can add a visual interaction with “state”. As shown in the figure below, the final visualization can have “state” as an interactive element where users can interact via mouse click interaction.



5.2. More Complex Visualization Examples

Similar to the visual language construction of VizGo explained above, more complex visualizations are also possible.

<u>*Geomap Visualization 1*</u>

Step 1. First we have the data table from the raw data set.

State	Year	Gender	Population
California	1990	Female	17,343,412
California	1990	Male	18,077,508
California	2000	Female	22,767,344
⋮	⋮	⋮	⋮
Maryland	2010	Female	3,002,247

Again, as the table shown above, our example data contains the population data in US by state, year, and gender. The data contains population information for each state in year 1990, 2000, and 2010. The use of metadata is same as the previous example.

Step 2. Create visual primitive polygon-circle and encode "population" to circle size

$$\mu_{size}(population)$$

Here, a "circle" is chosen as the visual primitive, and "population" is encoded to the size of each circle.

State	Year	Gender	Visualization
California	1990	Female	●
California	1990	Male	●
California	2000	Female	●
⋮	⋮	⋮	⋮
Maryland	2010	Female	●

Step 3. Encode "gender" to circle color

$$\mu_{color}(gender)\mu_{size}(population)$$

Next, gender is encoded by color. As we interpreted in step 1, gender's metadata is determined by red and blue colors. Therefore, visual primitives with female gender are encoded to red and male to blue in this step.

State	Year	Visualization
California	1990	🔴
California	1990	🔵
California	2000	🔴
⋮	⋮	⋮
Maryland	2010	🔴

Step 4. Visual elements are projected using "state," which is transformable to "longitude" and "latitude."

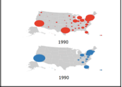
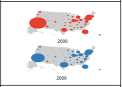
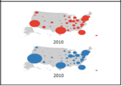
$$\prod_{\Psi(state)} \mu_{color}(gender)\mu_{size}(population)$$

State	Year	Visualization
California	1990	
California	1990	
California	2000	
⋮	⋮	⋮
Maryland	2010	

Visual primitives are projected to a geomap using the "longitude" and "latitude" information.

Step 5. Group by "year", sort and aggregate visual primitives by "gender"

$$\sum_{\substack{\uparrow gender\\ G(year)}} \prod_{\Psi(state)} \mu_{color}(gender)\mu_{size}(population)$$

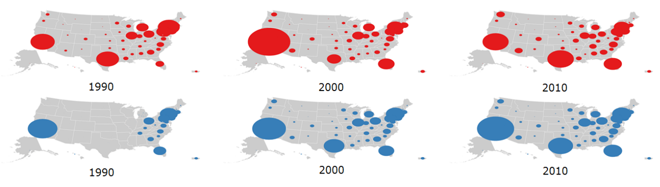
Year	Visualization
1990	
2000	
2010	

Visual primitives are grouped by "year." Since two primitives are integrated, we need to determine how to aggregate visual primitives. Here, the two visual primitives are overlaid on each other. The vertical sorting order in each visualization is determined by "gender."

Step 6. Sort and aggregate visual groups by "year"

$$\sum_{\uparrow year} \sum_{\substack{\uparrow gender\\ G(year)}} \prod_{\Psi(state)} \mu_{color}(gender)\mu_{size}(population)$$

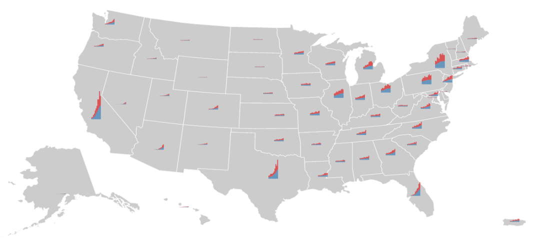
Finally, by integrating and sorting all visualizations via "year," we can get the final visualization.



<u>*Geomap Visualization 2*</u>

Stacked vertical bars in example 1 (Section 5.1) can also be projected to a geomap using our VizGo expression. The automatic visualization system can follow the same steps in Section 5.1 until step 5, and in step 6, the visual groups are projected using "state," which is transformable to "longitude" and "latitude." Finally, this visualization can be expressed with the VizGo grammar such as the following.

$$\prod_{\Psi(state)} \sum_{\substack{\uparrow year\\ G(state)}} \sum_{\substack{\uparrow gender\\ G(year)}} \mu_{color}(gender)\mu_{length}(population)$$



6 Discussions

From the results of our grounded theory analysis, we figured out three main limitations of the current visualization principles and guidelines. The limitations were determined since the current visual principles cannot support a more complex and necessary visualization systems in three aspects. In other words, the current principles have gaps to potential directions for further work on visualization systems.

In sum, we particularly concluded three limitations as follows: 1) *complexity of visualization guidelines* in form and content, particularly important as it relates to big data and broader use of analytics, 2) *importance of qualitative aspects* of visualization design, particularly relevant as visualization becoming more a commodity for data analysis in several domains, and 3) *relatively low utilization of some concepts*, such as interaction and collaboration, as potential research directions.

To overcome these limitations, we developed a new visual language, VizGo. VizGo has its main contributions by designing the language based on the concept of *constructiveness*. As we stated above, VizGo generates the visual language incrementally by applying different rules in each step. The example in Section 5 illustrated this incremental approach in detail.

Due to this incremental generation approach of visual language, VizGo can cover more extensive visualization principles. As we pointed in Section Appendix A, a systematic approach is required to overcome the complexity issues of current visualization guidelines. In other words, the visual guidelines should not be a simple matching problem from data to visualization, but rather it should be an optimization problem by generating visualizations incrementally. In addition, VizGo enables users to solve the visualization problem as an optimization problem by establishing visual analytics algebra and language to define data, visualization, and interaction altogether.

As a second limitation, we pointed the issues related to the visual guidelines' commodity in different domains. One of our main design goals for codified rules is to allow domain customization, which extends the basic set of design principles to cover more semantically guided design principles. For example, users might want to define the following design rules for their visualization: 1) apply logarithmic binning on attributes with power-law distributions; 2) map continuous attributes to a solid line; and 3) map attributes with a dollar currency metric to a dollar shape (i.e., $) with a green color. As a result, automatic visualization is achieved by reflecting users' customized rules in the system.

To overcome the low utilization of interaction and collaboration in the current visualization principles, we also added the *interactions (i)* operator to a set of VizGo's operators. By having this operator, VizGo allows adding visual interaction to the data and enables users to interact with the visualization.

7 Conclusion

As data grow exponentially, the sense-making process for information became increasingly important. However, among various types of visualizations, selecting the most appropriate visualization is still a challenging task due to the complexity of data or users' limited knowledge of graphic design. To remedy this issue, several visualization guidelines and principles have been proposed. In addition, various automatic visualization systems have been developed to reduce users' cognitive loads during the visualization process. However, until now these guidelines and automatic systems still face many limitations when interpreting the data and mapping the data to visualizations.

The first goal of our research is to find these limitations of current visualization principles. After finding these limitations, we mainly aim to develop a new visual language that can remedy those limitations. By having a new visual language, we eventually aim to develop a novel automatic visualization system that can cover various data and visualizations at the same time.

For this purpose, we first analyzed a large set of current visualization principles and guidelines via grounded theory analysis. Our analysis finding suggests that a variety of factors are considered for effective visualization design, including aspects of data, visualization, user, and insight and a variety of relationships between these factors including causal, contextual, and logical relationships. More importantly, we reported gaps and potential directions that we found in the literature requiring further work on principles and guidelines, for example, regarding effective support of collaboration and presentation. In sum, our study found the limitations of visualization principles, as visualization becomes a commodity used in several different fields by a variety of users of different backgrounds. From this analysis, we could conclude that a new visual language should be able to cover the following aspects: 1) It should be able to express a complex set of guidelines, 2) It should be flexible enough to customize for different domains, and 3) It should allow optimization of the complete pipeline, from data processing to visual representation and interaction.

Based on these three aspects, we developed a new visual language, VizGo. By defining a systematic and constructive visual language, we argued that VizGo can cover more complex and massive data visualizations. In addition, VizGo's codified rules allow users to customize the languages based on the domains and semantics. Finally, even though it's not yet developed completely, VizGo also contains simple interaction operators, so we can also say that it supports the complete pipeline, from data processing to visual representation and interaction.

However, our interaction operator is in its preliminary stage, and we need to develop a better interaction operator to cover visual guidelines. As a future work, we also propose to figure out the key concepts of interaction (e.g., are there such things as interaction primitives? If so, can we combine multiple interactions to create another interaction?). In addition, there has to be a natural way to blend the language of interactions to the current visual language, which only describes the pipeline from data processing to a visual representation.

In addition, we also proposed the architecture of the automatic visualization system based on our visual language, VizGo. By implementing this system, we expect to develop a more effective and flexible automatic visualization system that can reduce users' cognitive load during the visualization process.

Acknowledgement

The work described in this document was performed by collaborating with Dr. Eser Kandogan in IBM Research, Almaden. The author wishes to gratefully acknowledge Dr. Ben Shneiderman for his generous support and assistance. This work would not have been possible without their thoughtful comments.

References

[1] Shneiderman, B., The eyes have it: A task by data type taxonomy of information visualizations, Proc. IEEE Visual Languages '96, pp. 336-343, 1996.

[2] Tory, M., Potts, S., Möller, T., A Parallel Coordinates Style Interface for Exploratory Volume Visualization, IEEE Transactions on Visualization and Computer Graphics, 2004.

[3] Maletic, J.I., Leigh, J., Marcus, A., Dunlap, G., Visualizing Object Oriented Software in Virtual Reality, in Proceedings of the 9th IEEE International Workshop on Program Comprehension (IWPC'01), Toronto, Canada, May 12-13, 2001, pp. 26-35.

[4] Wiss, U., and Carr, D., A Cognitive Classification Framework for 3-Dimensional Information Visualization, Research report LTU-TR—1998/4—SE, Luleå University of Technology, 1998.

[5] Craft, B. and Cairns, P., Beyond Guidelines: What Can We Learn from the Visual Information Seeking Mantra?. In Proceedings of the Ninth International Conference on Information Visualization (IV '05). IEEE Computer Society, Washington, DC, USA, 110-118, 2005.

[6] Card, S. K. and Mackinlay, J., The structure of the information visualization design space. In Proceedings of the 1997 IEEE Symposium on Information Visualization (InfoVis '97), pages 92–99, 1997.

[7] Card, S. K., Mackinlay, J., and Shneiderman, B. Readings in Information Visualization: Using Vision to Think. Morgan Kaufmann, 1999.

[8] Card, S. K. Information visualization. In Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications. pp. 544 - 582, Lawrence Erlbaum Associates, Inc., Mahwah, NJ, 2002.

[9] Chi, E. H. and J. T. Riedl, J. T., An operator interaction framework for visualization systems. In Proc. IEEE Symposium on Information Visualization (InfoVis '98), pages 63–70, 1998.

[10] Chi. E. H., A Taxonomy of Visualization Techniques Using the Data State Reference Model. In Proceedings of the IEEE Symposium on Information Visualization 2000 (INFOVIS '00). IEEE Computer Society, Washington, DC, USA, 69-75, 2000.

[11] Carr, D., Guidelines for designing information visualization applications, Proceedings of the 1999 Ericsson Conference on Usability Engineering, Stockholm, 1999.

[12] Conati, C. and Maclare, H., Exploring the role of individual differences in information visualization. In Proceedings of the working conference on Advanced visual interfaces (AVI '08). ACM, New York, NY, USA, 199-206, 2008.

[13] Munzner, T., A Nested Model for Visualization Design and Validation. IEEE Transactions on Visualization and Computer Graphics 15, 6 (November 2009), 921-928, 2009.

[14] Dias, M. M., Yamaguchi, J. K., Rabelo, E., and Franco, C., Visualization Techniques: Which is the Most Appropriate in the Process of Knowledge Discovery in Data Base?, In Advances in Data Mining Knowledge Discovery and Applications, Associate Prof. Adem Karahoca (Ed.).

[15] Keller, P. R., and Keller, M. M., Visual Cues: Practical Data Visualization, Alamitos, CA: IEEE Computer Society Press, 1994.

[16] Keim, D., Designing pixel-oriented visualization techniques: theory and applications, IEEE Transactions on visualization and computer graphics 6:1 (2000), 59-78, 2000.

[17] Brehmer, M. Munzner, T., A Multi-Level Typology of Abstract Visualization Tasks," IEEE Transactions on Visualization and Computer Graphics, vol.19, no.12, pp.2376-2385, Dec. 2013.

[18] Gotz, D., Zhou, M.X., Characterizing users' visual analytic activity for insight provenance, IEEE Symposium on Visual Analytics Science and Technology, 2008. VAST '08, pp.123,130, 19-24 Oct. 2008.

[19] Tory, M., Moller, T., Rethinking Visualization: A High-Level Taxonomy, IEEE Symposium on Information Visualization, 2004. INFOVIS 2004, vol., no., pp.151-158, 2004.

[20] Andrews, K., Evaluation comes in many guises. AVI Workshop on BEyond time and errors (BELIV) Position Paper, pages 7-8, 2008.

[21] Lam, H., Bertini, E., Isenberg, P., Plaisant, C., Carpendale, S., Empirical Studies in Information Visualization: Seven Scenarios, IEEE Transactions on Visualization and Computer Graphics, vol.18, no.9, pp.1520-1536, Sept. 2012.

[22] Bertin, J., Semiology of Graphics. The University of Wisconsin Press. 1983.

[23] Mackinlay, J., Automating the design of graphical presentations of relational information. ACM Trans. On Graphics, 5(2):110-141. Apr. 1986.

[24] Roth, S., Kolojejchick, J., Mattis, J., and Goldstein, J., Interactive graphic design using automatic presentation knowledge. In Proceedings of the SIGCHI, 112-117. 1994.

[25] Stolte, C., Tang, D., and Hanrahan, P., Multiscale Visualization Using Data Cubes. Proceedings of the Eighth IEEE Symposium on Information Visualization, October 2002.

[26] Wilkinson, L., The grammar of graphics, 2nd edition. New York: Springer-Verlag.

[27] Wills, G., Wilkinson, L., AutoVis: Automatic Visualization. Information Visualization, 9(1): 47-69. 2010.

[28] Casner, S. M., Task-analytic approach to the automated design of graphic presentations. ACM Trans. Graph. 10, 2 (April 1991), 111-151, 1991.

[29] Gotz, D. and Wen, Z., Behavior-driven visualization recommendation. In Proceedings of the 14th international conference on Intelligent user interfaces (IUI '09). ACM, New York, NY, USA, 315-324, 2009.

[30] Voigt, M., Pietschmann, S., and Grammel, L., Context-aware recommendation of visualization components. eKNOW 2012, The Fourth International Conference on Information, Process, and Knowledge Management, 2012.

[31] Zhou, M. X., Ma, S., and Fend, Y., Applying machine learning to automated information graphics generation. IBM System Journal, 41(3): 504-523, 2002.

[32] Mackinlay, J., Hanrahan, P., Stolte, C., Show Me: Automatic Presentation for Visual Analytics. IEEE Transactions on Visualization and Computer Graphics, 13(6). November/December 2007.

[33] Tableau Software. http://www.tableausoftware.com/

[34] TIBCO Spotfire. http://spotfire.tibco.com/

[35] Steele, J., and Iliinsky, N., Beautiful Visualization: Looking at Data Through the Eyes of Experts (1st ed.). O'Reilly Media, Inc, 2010.

[36] Yau, N., Data Points: Visualization that Means Something (1st ed.). Wiley Publishing, 2013.

[37] Wong, D., The Wall Street Journal Guide to Information Graphics: The Dos and Don'ts of Presenting Data, Facts, and Figures (1st ed.) W. W. Norton & Company, 2013.

[38] Ward, M., Grinstein, G., and Keim, D., Interactive Data Visualization: Foundations, Techniques, and Applications. A. K. Peters, Ltd., Natick, MA, USA, 2010.

[39] Battiti, R., Mauro B., Reactive Business Intelligence. From Data to Models to Insight, Trento, Italy: Reactive Search Srl, 2011.

[40] Bertini, E., and Lalanne, D., Investigating and reflecting on the integration of automatic data analysis and visualization in knowledge discovery. SIGKDD Explor. Newsl. 11, 2 (May 2010), 9-18, 2010.

[41] Polowinski, J., and Voigt, M., VISO: a shared, formal knowledge base as a foundation for semi-automatic infovis systems. In CHI '13 Extended Abstracts on Human Factors in Computing Systems (CHI EA '13). ACM, New York, NY, USA, 1791-1796, 2013.

[42] Cleveland, W.S. and McGill, R. Graphical Perception: Theory Experimentation and Application to the Development of Graphical Models, Journal of the American Statistical Association, Vol. 79, No. 387. (Sep., 1984), pp. 531-554, 1984.

[43] Kelleher, C., and Wagener, T., Short communication: Ten guidelines for effective data visualization in scientific publications. Environ. Model. Softw. 26, 6 (June 2011), 822-827, 2011.

[44] Sokal, R. R. and Rohlf, F. J., Biometry: the principles and practice of statistics in biological research. 3rd edition. W. H. Freeman and Co., 1995.

[45] Hoey, J., The Two-Way Likelihood Ratio (G) Test and Comparison to Two-Way Chi Squared Test, 2012. eprint arXiv:1206.4881.

[46] Harremoes, P. and Tusnady, G., Information divergence is more $\chi 2$-distributed than the $\chi 2$-statistics, 2012 IEEE International Symposium on Information Theory Proceedings (ISIT), vol., no., pp.533-537, 1-6 July 2012.

[47] Becker, R. A., Cleveland, W. S., and Wilks, A. R., Dynamic graphics for data analysis, Statistical Science, vol .2, pp. 355-383, 1987.

[48] Foley, J. D., van Dam, A., Feiner, S. K., and Hughes, J. F., Computer Graphics: Principles and Practice (2nd Ed.). Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1990.

[49] Dix, A., Finlay, J., Abowd, G., and Beale, R., Human-Computer Interaction. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1997.

[50] Ward, M. and Yang, J., Interaction spaces in data and information visualization. In Proceedings of the Sixth Joint Eurographics - IEEE TCVG conference on Visualization (VISSYM'04), 2004.

[51] Lam, H., "A Framework of Interaction Costs in Information Visualization," IEEE Transactions on Visualization and Computer Graphics, vol. 14, no. 6, pp. 1149-1156, November/December, 2008

[52] Yi, J., Kang, Y., Stasko, J., and Jacko, J., Toward a Deeper Understanding of the Role of Interaction in Information Visualization. IEEE Transactions on Visualization and Computer Graphics 13, 6 (November 2007), 1224-1231, 2007.

[53] Kosara, R., Hauser, H., and Gresh, D., An interaction view on information visualization, In State-of-the-Art Proceedings of EUROGRAPHICS 2003 (EG 2003), pages 123–137, 2003.

[54] Ware, C., Information Visualization: Perception for Design. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

[55] Chuah, M.C., Roth, S.F., On the semantics of interactive visualizations, Information Visualization '96, Proceedings IEEE Symposium on , vol., no., pp.29,36, 28-29 Oct 1996.

[56] Buja, A., Cook, D., and Swayne, D. F., Interactive High-Dimensional Data Visualization, Journal of Computational and Graphical Statistics, Vol. 5, No. 1 (Mar., 1996), pp. 78-99, 1996.

[57] Keim, D. A., Information Visualization and Visual Data Mining. IEEE Transactions on Visualization and Computer Graphics 8, 1 (January 2002), 1-8, 2002.

[58] Tweedie, L., Characterizing interactive externalizations. In Proceedings of the ACM SIGCHI Conference on Human factors in computing systems (CHI '97). ACM, New York, NY, USA, 375-382, 1997.

[59] Spence R., Information Visualization: Design for Interaction (2nd Edition). Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2007.

[60] Zhou, M. X. and Feiner, S. K., Visual task characterization for automated visual discourse synthesis. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '98), ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 392-399, 1998.

[61] Jankun-Kelly, T. J., Ma, K., and Gertz, M., A Model and Framework for Visualization Exploration. IEEE Transactions on Visualization and Computer Graphics 13, 2 (March 2007), 357-369, 2007.

[62] Elmqvist, N., Moere, A. V., Jetter, H., Cernea, D., Reiterer, H., and Jankun-Kelly, T. J., Fluid interaction for information visualization. Information Visualization 10, 4 (October 2011), 327-340, 2011.

[63] Norman, D. A., The Design of Everyday Things. Basic Books, Inc., New York, NY, USA, 2002.

[64] Kindlmann, G., Scheidegger, C., An Algebraic Process for Visualization Design, IEEE Transactions on Visualization and Computer Graphics, vol.20, no.12, pp.2181-2190, Dec. 31 2014.

[65] Kandogan, E. and Lee, H., A Study on the Language of Data Visualization Principles and Guidelines, working paper.

APPENDIX A. ANALYSIS OF VISUALIZATION PRINCIPLES [65]

To find the general patterns and limitations of the existing visual languages we conducted a grounded theory study and examined the language and content of data visualization principles and guidelines.

Method

We employed grounded theory as the primary method of analyzing our data. Grounded theory is commonly used in social sciences to develop a theory based solely on data. It is based on iterative multi-level coding of data, where concepts (categories) emerge from analysis of the data. In our case, our purpose was not to develop a theory, but the iterative coding and the data-driven concept development process of grounded theory was what we needed.

Resources

We used 5 books and 18 papers as resources for our analysis. Books chosen were more intended for general public, specifically on effective visualization design, included Beautiful Visualization by Steele and Illinsky [35], Data Points by Yau [36], and Wall Street Journal Guide to Information Graphics by Wong [37], and [38][39]. Papers included work on automatic data visualization, visualization in knowledge discovery, visualization recommendations for science, and visual analytics, such as [23][41][42][43]. In addition, we also examined a visualization blog (School of Data, schoolofdata.org) and a documentation of a visualization product. Overall, we extracted about 550 guidelines, ranging in length from 5 to 140 words. The full list of resources and exemplar guidelines are in Appendix E.

Analysis

In grounded theory there are three stages of coding: *Open coding* is the first level of coding, in which concepts, properties, and dimensions are identified at the desired level of granularity. In the second level, *axial coding*, the goal is to relate concepts, and identify context of these relationships. In the last level, *selective coding*, the goal is to integrate all these to form a larger theoretical scheme. Since our goal was not to develop a theory we only performed open coding and axial coding, and did not proceed to selective coding.
We used iterative coding in which we selected a small subset of the guidelines and coded it independently. We then gathered together and resolved issues in our coding scheme, and coded a different set of guidelines. Codes used in our iterations emerged from the data, as it should in grounded theory analysis. After a couple of iterations we (two coders) each felt comfortable with the scheme and coded the rest of the guidelines independently.
Our coding scheme was hierarchical. We identified about 5 high-level codes, relating to general concepts such as Data, Visualization, and User. Each level then described more detailed sub-concepts. For example, Data/Attribute was a 2nd-level concept, Data/Attribute/ Measurement was identified as a 3rd-level concept. At certain levels we identified varying instances of the concept, and coded them in parenthesis. For example, nominal measurement was coded as, Data/Attribute/Measurement(nominal).
In total, we derived 515 concepts at several levels, including all the varying instances. On the average, each guideline was coded with 6 (hierarchical) concepts, and the longest guideline contained 17 concepts. In guidelines we examined 3% contained 1st-level, 56% 2nd-level, 30% 3rd-level, and 10% higher level concepts. See Appendix B, C, and D for a detailed description of the coding scheme.

Summary of Findings

Below, we present our findings from analyzing the 550 guidelines we identified. First, we will describe the high-level concepts we identified in the description of guidelines and relationships that exist among these concepts. Then, we will examine key patterns we observed, particularly in conditional, ranking, and explanation type guidelines.

Table 1. Top twenty pairs of concepts with strong associations ($p < 0.0002$) (V=Visualization, D=Data)

Code 1	Code 2
User	Expression/Supportive
V/Attribute	V/Operation/Mapping
Data/Attribute	V/Operation/Mapping
User/Task/Integrate	Insight/Composition
Expression/Supportive	Insight
V/Component/Axis(h)	V/Component/Axis(v)
V/Attribute/*/Position	V/Class(xy)
D/Attribute	Expression/Conditional
V/Element	V/Class(xy)
V/Element/Point	V/Attribute/*/Position
D/Operation	V
Expression/Existential(1)	Expression/Conditional
Expression/Conditional	Expression/Supportive
User/Ability	Expression/Supportive
V/Element/Point	V/Class(xy)
Qualifier/Logical(not)	Expression/Adversative
V/Element	V/Attribute/*
Qualifier/Existential(only)	Expression/Existential(none)
V/Element/Bar	V/Attribute/*/Length
V/Attribute/*/Color	V/Operation/Mapping

Observed Concepts from the open coding analysis

Finding 1. High-level concepts

During our iterative coding, five high-level concepts emerged from our data: **Data**, **Visualization**, **User**, **Insight**, and **Device**, as explained below:

• **Data**: Data covers all aspects related to information, including its attributes, schema (e.g. hierarchical), operations (e.g. aggregation), domain (e.g. business), and issues related to domain such as trust, privacy, validity, etc. Most of these aspects constituted second level concepts, each having further sub-concepts. For example, Data/Attribute concept was further refined regarding size, semantics (e.g. time), role (e.g. measure), measurement (e.g. nominal, ordinal), distribution (e.g. sparse, normal), etc.

• **Visualization**: Visualization addresses all aspects related to representation and interaction of information, such as elements (e.g. bar), attributes (e.g. length), components (e.g. legend, axis), class (e.g. scatter plot), operations (e.g. group), and interaction (e.g. zoom). As in Data, each of these had further sub-concepts. For example, Visualization/Attribute further contained various visual attributes such as size, position, color, etc.

• **User**: User concept entails all aspects related to human viewing and interacting with visual representation of data, including tasks (e.g. compare, communicate), (dis)abilities (e.g. read, recognize, recall), and social aspects (e.g. conventions).

• **Insight**: Insight represents all intuition, understanding, and knowledge to be gained from data visualization. While there might be overlap with user tasks, the difference is in functionality. Insight is the outcome, task is the process, and interaction is lower-level actions.

• **Device**: Device covers the physical aspects of the visualization medium, particularly related to display and interaction capabilities.

Table 2. Patterns used in the instruction guidelines (V=Visualization, D=Data, A=Attribute, C=Component, E=Element, O=Operation)

Pattern with example (coverage)	*Pattern with example (coverage)*
V/O/* "...colored by...", (26%)	**V/O/Aggregation** "...bars should be stacked….", (3.5%)
V/Class/* "stacked bar chart", (22.3%)	**V/A/Color** "...warm color...", (3.3%)
(V/[A|E|C|Class])+ "...size and position of the stars...", (17.9%)	**V/C/Axis** "... x-axis...", (3.3%)
D/A/* "...category dimensions in the data...", (17.0%)	**V/Coordinate/*** "...cartesian coordinates..", (3.1%)
V/A/* "...height of the...", (13.5%)	**V/O/Order** "...order bars by...", (2.9%)
(V/[A|E|C|Class])+ -- V/O "...lines on a line chart should be colored according to...", (12.6%)	**D/A/Semantics(temporal)** "...year...", (2.7%)
Qualifier/Qualitative(*) "...unfamiliar with..", (11.3%)	**V/Multiple** "....multiple views...", (2.7%)
V/O/Mapping(*)"map a particular field to ...", (10.1%)	**V/Class(scatter)** "...scatter plot...", (2.7%)
V/E/* "...shape of the...", (9.7%)	**V/Size/*** "...large visualization...", (2.4%)
V/C/* "...legend should...", (9.3%)	**V/* -- D/* -- D/Domain** "...pie charts are common in business domain for data...", (2.4%)
(D/*)+ -- (V/[A|E|C|Class])+ -- V/O "...categorical data represented as bars should also have colors...", (8.2%)	**D/Schema(*)**"...network data...", (2.2%)
D/A/Measurement(*)"..one category dimension...", (8.0%)	**V/O/Projection/*** "... mercator projection..", (2.2%)
D/O/* "...average...", (5.9%)	**V/Class(xy)** " ...two-dimensional plot...", (2.0%)
(D/*)+ -- D/O -- (V/[A|E|C|Class])+ "...numeric metric aggregated as average on a bar...", (4.8%)	**D/* -- D/O/*** "..numeric data can be summarized by...", (2.0%)
D/Domain(*)"...in business...", (4.4%)	**D/A/Role(measure)** " ..two or more measures in data...", (1.8%)
D/A/Measurement(nominal) "...one or more categorical dimensions...", (3.8%)	**V/Class(line)** " ...a line chart..", (1.8%)
V/Class(bar) "...bar chart...", (3.8%)	**V/C/ColorPalette(*)**"...using a light to dark color palette...", (1.8%)
D/A/Semantics "...location data...", (3.5%)	**V/E/Bar** ".. vertical bars on the..", (1.8%)

Finding 2. Contextualizing concept

We also identified a subsidiary concept, which is a refining/contextualizing concept **Qualifier**. Qualifiers serve to refine and contextualize quality and quantity of concepts. For example, it may refer to a specific quantity, such as "two nominal attributes", or refer to a less well-defined quality such as "large data" or "dirty data". We identified over 35 such qualities in our data. Qualifiers may also refer to compute aspects of concepts such as "length of the labels".

Finding 3. Relationships

In order to express the relationships among (groups of) concepts, we created an **Expression** concept. It entails logical, (in)equality, similarity, existential, rank, and prepositional relationships. Also, it includes structural representation such as conditional, copulative, causal, adversative, and supportive relationships.

<u>Observed Patterns from the axial coding analysis</u>

From the analysis, we found that frequency of these high-level concepts is ranked as follows: Expression (25.9%) - Visualization (25.7%) – Data (15.1%) – Qualifier (14.9%) – User (9.98%) – Insight (7.74%) – Device (0.71%). The co-occurrence of concepts is also important to analyze, since it can reflect a strong association between codes in guidelines. The results are shown in Table 1, which lists the top twenty pairs of concepts with strong associations. Here, the strength of association is calculated using the G-test [44], which is asymptotically equivalent to the chi-squared test for goodness of fit, but it is more accurate for small sample sizes [45][46].
In our data we found three forms of guidelines: (1) Imperatives, (2) Declaratives, and (3) Conditionals.

• ***Imperatives*** are basically guidelines that provide an essential direction to follow, typically the do's and don'ts in visualization design, for example, "Don't create shadows behind bars", "Always extend bar charts to zero baseline", or "Reveal data at several levels of detail". Overall we found that 11% of the guidelines were imperatives. Imperatives tend to be shorter in description and frequently have a negative form (e.g. "Do not...", "Never..."). In our data, imperatives were about 52.5% shorter than declarative forms and about 41.0% of the imperatives were expressed in negative form.

• ***Declaratives*** are statements regarding a design rule, principle, or opinion often with an explanation. They were by far the most common form of guidelines, constituting about 73% of the guidelines we examined.

• ***Conditionals*** are guidelines that state a condition and a consequence, which holds true only if the condition is satisfied. We observed several pattern both in the condition and consequence parts. Conditionals made up about 16% of the guidelines.

Whether specified in conditional, imperative, or declarative forms, guidelines provides 1) instructions, describing what should be done and how, 2) conditions, illustrating a condition and a consequence, 3) rankings (or comparisons), stating preferences of one concept over others, and 4) explanations, providing a reasoning beyond the suggested action or statement. Below, we provide our analysis of the patterns identified in the form and content of the guidelines.

1. Instructions

Instructions make up the core part of the guideline, they express what actions to take and how. Instructions constitute the bulk of the content in imperative and declarative forms. In conditionals, instructions only exist in the consequence part (i.e. not in the condition part).
Instructions typically involve a *segment*. A data segment refers to a sequence of data attribute, schema, size, etc., optionally with data operations and qualifiers. And, a visual segment refers to a sequence of visualization attribute, element, component, or type, optionally with one or more visual operations and qualifiers. Simple instructions

include either a data or a visualization segment, e.g. “group data into bins”, coded as D/Operation/ Aggregate(bin), or “use a bar chart”, coded as V/Class (bar), respectively.

More complex instructions involve several data and visual segments, often combined with a logical expression (e.g. “two measures and a lot of data values”, coded as Q/Existential(2), D/Attribute/Role (measure), X/Logical (and), Q/Qualitative(large), D/Size). Other expressions such as (in)equalities (e.g. “median is a more robust measure than average”, coded as D/Operation/Aggregation(median), X/Inequality(more), Q/Qualitative(robust), D/Operation/ Aggregation(average)), similarities (e.g. “keep axis ranges similar”, coded as V/Component/Axis/Range, X/Similar, V/Component/Axis/Range), prepositional or possessive expressions such as in, with, and of (e.g. “labels in graphs”, coded as V/Element/Label, X/Preposition(in), V/Class(graph)), and other types of expressions that requires a computation such as distances (e.g. “legend separated from the line”, coded as V/Component/Legend, X/Spatial/Distance(far), V/Element/Line).

In our data we found 70.4% of the guidelines to be in simple form. Even so, 42.8% of these simple forms included one or more visual or data operations. On the other hand, 22.3% of the guidelines contained one, and 7.3% two or more expressions. Logical expressions are used in 9.9% of the guidelines, while in(equalities) and similarities were included in 4.2% and 1.1% of the guidelines, respectively. Prepositional expressions were used in 9.0%. Other expressions (e.g. Spatial) were used rarely, i.e. 0.2%. Qualifiers existed in about 47.2% of the guidelines, of these 68.2% only one, 20.5% two, and 11.2% three or more.

In Table 2, we list the common patterns in data and visual parts of the instructions. Note that the patterns here can be combined with others to make up the guideline through.

2. Conditions

In our data, we found that data and visualization concepts are heavily utilized in the condition part of the guidelines with 69% and 25% coverage respectively, while on the other hand user, insight, and device concepts are rather underutilized, with 9%, 6%, and 3% coverage respectively

Overall patterns in the condition part are of the form: (1) Concept/*, which states that a certain concept is valid, for example, “hierarchical schema”, coded as D/Schema(hierarchical), (2) Concept/* (-- Q/*), which states a specific concept, with a certain qualifier, for example, “large number of bars”, coded as V/Element/Bar-- Q/Qualitative(many), (3) Concept/* -- X/* -- Concept/*, which states an expression or operation applied on concepts, for example, “data contains time”, coded as Data -- X/Existential -- D/Attribute/Semantics (temporal), and (4) Operand -- X/Logical(*) -- Operand, a logical expression composed of above primitive forms. We found that in our data only 12.6% is in the first primitive form, while 19.4% in the second form, and 37.9% and 27.6% in third and fourth form, indicating the complexity of the condition part of the guideline.

In Table 3, we list frequently observed patterns (ordered from most frequent to least) in the condition part, along with examples and coverage, indicating % of conditions in which pattern occurred. At the top of the table are mostly data concept related conditions, e.g. whether a data attribute of certain measurement, semantics, range, distribution quality is satisfied. On the other hand, interaction (Visualization/Interaction/*) and data domain (Data/Domain/*) concepts (not listed on table) came at the bottom, with 1.2% each.

3. Rankings

Rankings are statements that indicate a preference between two or more concepts in a guideline. Rankings can be relative in which two or more concepts are compared, and given a relative ordering. They can be absolute, suggesting one or more concepts are given a fixed

Table 3. Patterns in the conditional guideline

Patterns with example (coverage)
D/*--Expression/Existential/*--D/A/Measurement(*) "there is a category dimension", (14.9%)
D/A/(Value|Range|Distribution)/*--(Qualifier/Qualitative(*)) "long names", "values are in the negative", "outliers are present"(%14.9, total)
D/*--EZ/Existential/*--D/A/Semantics(*) “data contains time”, (11.5%)
V/E/*(--Qualifier/*) "most important line", "at least ten bars", (9.2%)
User/Task*(--Qualifier/Qualitative(*)) "communicating data to audience", "when contrasting...", (8.0%)
D/Size--Qualifier/Qualitative(*) “large datasets", (6.9%)
D/*-- D/O/*--D/A/* "aggregated as average", (%6.9)
D/Schema/*--(Qualifier/Qualitative(*)) "hierarchies in data", (%6.9)
D/*--Expression/Existential/*--D/A/Role(*) "there are (two) measures:, (6.9,%)
Insight/*(--Qualifier/Qualitative(*)) "When detailed values are important", (5.7%)
V/A/*(--Qualifier/Qualitative(*)) "very long labels", "large block of text", (4.6%)
--V/O/Mapping/*-- "when using direction", "coloring", (4.6%)
V/Class/*(--Qualifier/*) "On choropleth maps", (4.6%)
Device/*(--Qualifier/*) "color is available", (3.4%)

Table 4. Patterns used in the ranking relationship guidelines

Patterns with example (coverage)
V/Class/* vs. **V/Class/*** “...preferred a grouped dot chart to a grouped bar chart”, (40.7%)
V/A/* vs. **V/A/*** “...position has a higher ranking than area…”, (22.2%)
V/A/X/* vs. **V/A/X/*** “...warm colors appear larger than cold colors…”, (11.1%)
V/O/* vs. **V/O/*** “...log scale is .. less than … linear scale”, (11.1%)
Insight/* vs. **Qualifier/Qualitative/*** “...density of points… more informative… than … points overlap”, (7.4%)
V/E/* vs. **V/E/*** “...continuous line…(easier).. bars”, (7.4%)
V/Coordinates/* vs. **V/Component/*** “...labels …instead of… legend...”, (3.7%)
Qualifier/Qualitative vs. **Qualifier/Qualitative/*** “...informative and efficient, than …aesthetics...”, (3.7%)

Table 5. Patterns used in the explanation guidelines

Patterns with example (coverage)
(Q/Qualitative(*))* -- Insight/* --* "...significant outlier...", (40.4%)
(Q/Qualitative(*))* -- User/Ability/* --* "….hard to read...", (29.4%)
(Q/Qualitative(*))* -- User/Task/*--* "….easily rank....", (19.3%)
(V/*)* "...scatter plot...", (12.1%)
(D/*)* "...categorical data....", (12.1%)
(Q/Qualitative(*))* -- Insight/* -- (V/* | D/*)+ "... of the clusters in scatter plot", (7.7%)
*** -- D/Domain(*) -- *** "...represent residential areas...", (6.5%)
(Q/Qualitative(*))* -- User/Ability/* -- (V/* | D/*)+ "...length and position easily quantitatively perceived...", (5.2%)
(V/Operation/*) -- * "...scale the data points...", (3.6%)
(Q/Qualitative(*))* -- User/Task/* -- (V/* | D/*)+ "...better compare all lines...", (3.6%)
(D/Operation/*) -- * "...sort by...", (3.6%)
*** -- D/Schema(*) -- *** "...multi-dimensional data...", (2.8%)
*** -- Device/* -- *** "...limitation on display size and resolution..", (2.8%)
(V/Interaction/*) -- * "...link to more explanation...", (1.6%)
***-- D/Size(*) -- *** "...a lot of data at once...", (1.2%)
(D/New|Privacy) -- * "...change in data...", (0.8%)

rank independently. Guidelines can also talk about changes in ranking in response to a specific choice in the guideline. In our data about 5% of the guidelines contained a ranking. (Table 4).

We also analyzed rankings to see whether a cause or supportive argument is made or not and examined the arguments. About 74% of the rankings contained a cause or supportive argument. Insight and user ability/task related concepts (> 80%) are identified as top concepts as part of the arguments. It is also interesting to note that 16.7% of the supportive argument is made in negative form.

4. Explanations

Explanations are basically parts of the statements that provide reasoning in support of the guidance. In our data, about 45.3% of the guidelines contained either a cause (E/Cause) or supportive (E/Supportive) argument.

Table 5 shows a breakdown of the content of the explanations. At the top are insight, user ability, and user tasks, optionally qualified by qualitative attributes. Next come explanations that indicate support for a particular visualization aspect, followed by data aspects, in general. Then, we saw insight, user ability, and user's tasks with additional context provided in reference to a specific visualization or data aspect. We saw little use of visualization interaction as a cause. Device characteristics, data size, privacy, etc. were very rarely specified as argument for the explanation.

Discussions of Current Visualization Principles

From the analysis, we also observed gaps between current visualization principles and potential directions in further visualization researches. In sum, we particularly concluded three limitations from our analysis: 1) *complexity of visualization guidelines* in form and content, particularly important as it relates to big data and broader use of analytics, 2) *importance of qualitative aspects* of visualization design, particularly relevant as visualization becoming more a commodity for data analysis in several domains, and 3) *relatively low utilization of some concepts*, such as interaction and collaboration, as potential research directions. In the next few sections, we discuss these themes in more detail, along gaps and potentials, and their implications.

Limitation 1. Complexity, Scalability, Automation

As big data becomes a major topic in enterprise analytics and consequently as the user base of analytics broadens, automating the design of effective visualizations (along with the underlying data and analytics pipeline) is now more critical than ever. This has several implications.

First, as discussed earlier, visualization design is a complex task, requiring consideration of several factors at once. Simple matching between data attribute characteristics and a set of visualizations may not do justice to represent the complexity of visual analytics. For big data, the situation is worse, as data must be carefully transformed and organized before even visualization is considered. Interactive visualization of big data is even harder. In our analysis, we found relatively lower use of data operations, particularly as they related to visualization and interaction with data. We need to systematically identify patterns of visual interaction with big data that exhibit high utility. Secondly, we argue that visualization and analytics should be considered, and optimized, in an integrated manner, to increase the overall effectiveness of the whole process. This requires a systematic approach in which data and visual operators are represented on equal terms, potentially requiring visual analytics algebra and language to define data, visualization, and interaction altogether, that can be optimized (as in relational algebra), as such making automatic visualization not a matching problem but rather an optimization problem.

Limitation 2. Domain, Semantics

Another implication of the broadening user base is that visualization is becoming more a commodity and used in several domains. In our data, we have see relatively low use of data domain and semantics (beyond time) in guiding effective data visualization. However, we argue that domain and semantics will be more important as visualizations will be used in different domains, as part of everyday tools. There are several implications of this.

First, we need to develop more guidelines that leverage domain and semantics. For example, temperature in physics and medicine are very different concepts. Though some general principles apply, based on the measurement type, much is left to design, particularly as it relates to qualitative issues. In different domains, possible value ranges are different, more importantly the meaning associated with values are different. For example, in medicine, the normal body temperature ranges should be considered, while in physics, for example, it could be the melting point. These impact visual design, choices of colors, axis ranges, emphasis on critical points and ranges, etc.

Secondly, in our data we observed only very little consideration of multiple views and datasets. In almost any domain, decision-making involves consideration of multiple data and how they relate to each other. This is clearly another area where guidelines need to be developed for effective visualizations.

Lastly, given the wide range of possibilities to incorporate domain and semantics into the equation of visualization design, again we may need to consider a systematic approach to design, one that involves a language that facilitates specification of different considerations. The language should be flexible to express different semantics and design know-how. A language based approach to automation would support customization quite effectively by building different repositories of design know-how for different domains and incorporating desired repository into the system based on domain of the data analytics problem.

Limitation3. Interaction, Collaboration, Presentation

Driven by broader use of visualization in decision-making in business and public, we need to support different phases of the process, including the analysis phase, which includes a lot of interaction with data but also the collaborative aspect of it. A critical and often overlooked phase is that of presentation of analytics work directly in support of decision makers. Unfortunately, we saw few guidelines that incorporated aspects of interaction, and almost none that considered collaboration and presentation. These are clearly areas of further research, to establish clear guidelines on how to best support interaction, collaboration, and presentation.

Appendix B. Concept-related codes: 5 high-level concepts and their 2nd and 3rd level concepts with necessary arguments

1st level concept: Data (D)	
2nd level concept	3rd level concept (argument if needed)
Data/Schema/*	2d, 3d, multi-dimensional, multivariate, mixed, sequential, hierarchical, graph
Data/Attribute/*	value (min, max, average, median), size (large, small), semantics (temporal, spatial, geographical, temperature, percentage, count), unit, role (measure, control), measurement (nominal, ordinal, interval, ratio, numeric, continuous, discrete, cyclical), distribution (sparse, dense, uniform), range (wide, narrow)
Data/Operation/*	aggregation (spatial, average-mean, average-median, sum, min, max), outlier, sort (increase, decrease, alphabetically), union, annotate, explain, slice, filter, simplify, group (small, range, type, region, time), transform (text-numeric, text-abbreviation), fit (line, curve), ratio, round, reduce
Data/Domain/*	geography, business, politics, it, survey, science, technology, media, sports, meteorology
Data/Misc/*	size (large, small), missing, incorrect, new, privacy, trust, precision, related (time), context, source, selection, organization

1st level concept: Visualization (V)	
2nd level concept	3rd level concept (argument if needed)
Visualization/Element/*	point, line, shape (circle, rectangle, cube, sphere, 2d, 3d, fill), bar, pie, text, node, edge, glyph, error bar
Visualization/Attribute/*	size, radius, position, color (hue-bright/dark/red/gray, saturation, brightness-light/dark), length, area, angle, direction, orientation, pattern, shape, label, weight, texture, closure, connection, volume, transparency, style, typography
Visualization/Operation/*	label, mapping, projection (3d, row, column), aggregation (stack, cluster, overlay), nesting, order (top-bottom, clockwise, time-incremental, cyclical), rearrange, distortion, highlight (color, multi), group (bottom-N, 3, 5), scale, position (x, y, start, end, middle, below, aligned, left-aligned, right-aligned, center-aligned, baseline-aligned, point-aligned)
Visualization/Component/*	background, title, axis (negative, h, v, tick, range), grid, label, legend, color palette (redgreen, lightdark, warm, cold, alternating), shape palette, description, source, thumbnail, table (row, column, cell)
Visualization/Coordinate/*	cartesian-scale (double, range-small/baseline, numeric-linear/log, categorical), polar, geo (projection-mercator/albers)
Visualization/Class/*	xy plane, scatter, bar, list, table, graph, 2d, 3d, map, pie, donut, radial, calendar, histogram, hierarchy, abstract, boxwhisker, symbols, line, area, cycle, treemap, mosaic, star, contour, choropleth map, cartogram, heatmap, parallel coordinates, dot, density, surface, pictogram, graphical
Visualization/Interaction/*	change metaphor, filter, pan, rotation, scroll, select, zoom (multiple), link to source, annotate, explain, overview, detail, sort, split, animation, highlight, style (direct manipulation)
Visualization/Misc/*	standards, organization, size (small, large), aspect, layer (map), multiple, aesthetics

1st level concept: User (U)	
2nd level concept	3rd level concept (argument if needed)
User/Task/*	process, context, goal, reflect, emotion, perspective, comparison, integrate, search, describe, communicate, find, browse, explore, analyze, present, explain, monitor, decision making, classify
User/Ability/*	Attention, perceive (differentiate, order, measure, change), understand, recognize, read, learn, retain, recall, locate
User/Mics/*	disability (colorblind), social (conventions), action (click, move)

1st level concept: Insight (I)	
2nd level concept	3rd level concept (argument if needed)
Insight/Trend/*	change, past, steady, strength, ragged, absolute, time, linear, cyclical
Insight/Misc/*	relationship (multi, part to whole), correlation (pairwise, multi), composition, variance, extrema, value, comparison, distribution, progress, rank, quality, summarization, structure (hierarchy), cluster, causality, message, guide, meaning, outlier, gaps, percentage, similarity, pattern, detail, overview, focus

1st level concept: Device	
2nd level concept	3rd level concept (argument if needed)
Device/Display/*	size, resolution, aspect, density
Device/Input/*	mouse, keyboard

Appendix C. Concept-related codes: Contextualizing concept and its 2nd and 3rd level concepts with necessary arguments

1st level concept: Qualifier (Q)	
2nd level concept	3rd level concept (argument if needed)
Qualifier/Qualitative/*	good, bad, relevant, easy, accurate, consistent, important, powerful, useful, precise, noisy, simple, fancy, informative, novel, efficient, beautiful, successful, appropriate, quick, familiar, redundant, usable, distorted, dense, cluttered, lossy, clear, explicit, implicit, flexible, compact, narrow, engaging, different, continuously, discretely
Qualifier/Misc/*	numeric (1, 2, 3, 4, 5, … , N, many, few), cardinality (N, many, few), rank (N, top, bottom), inequality (more, less, N), existential (many, some, none, only, all, except, emphasis), logical (not), temporal (recent), spatial (width)

Appendix D. Relationship-related codes and their usage

1st level concept: Expression (E)	
2nd level concept	Usage examples
Expression/Logical	and, or
Expression/Inequality	more, less
Expression/Existential	1,2, many, some, none, only, all, at least,…
Expression/Rank	If X is ranked * than Y
Expression/Conditional	If X then Y
Expression/Copulative	Connecting X and Y
Expression/Optional	Optionally, ... Y
Expression/Alternative	X ... alternatively... Y
Expression/Exemplar	such as …, for example, ...
Expression/Causal	So that ...,because of X, … Y, X leads to Y
Expression/Adversative	... but/however, X but Y
Expression/Supportive	X supports Y, X enables Y, X allows Y
Expression/Possessive	X of Y, X with Y
Expression/Prepositional	X in Y, X below Y, X with Y
Expression/Declarative	X is Y
Expression/Imperative	do X
Expression/Misc	Equal, Similar, Spatial

Appendix E. Exemplar Guidelines

Guideline (*reference*) ***[coding result w/ concept-related and relation-related codes]***
"All color rankings are based on social convention: blue ribbon, red ribbon, white ribbon yellow alert, orange alert, red alert." (*a*) ***[E/Declarative V/Operation/Order V/Attribute/*/Color E/Supportive U/Social/Conventions]***
"Use pie charts with care, and only to show part of whole relationships." (*b*) ***[E/Imperative V/Element/Pie E/Supportive Q/Existential(only) I/Relationship(partToWhole)]***
"Two is the ideal number of slices, but never show more than five." (*b*) ***[Qualifier/Numeric(2) E/Declarative Q/Cardinality(N) V/Element/Pie E/Adversative Qualifier/Logical(not) E/Inequality(more) Q/Numeric(5)]***
"Use line charts to show time series data. That's simply the best way to show how a variable changes over time." (*b*) ***[E/Imperative V/Element/Line V/Class/xy E/Supportive D/Attribute/Semantics(temporal) E/Copulative E/Supportive I/Trend(time)]***
"To compare values based on length, you must see both ends of the lines or bars. Otherwise you end up with a skewed view of maximums, minimums, and everything in between." (*c*) ***[U/Task/Comparison E/Prepositional V/Attribute/*/Length E/Supportive V/Coordinate/Cartesian/Scale/Numeric E/Possessive V/Element/Line V/Element/Bar E/Adversative Qualifier/Qualitative(distorted)]***
"Angles are commonly used to represent parts of a whole, using the fan favorite, but often maligned, pie chart shown in X." (*c*) ***[E/Declarative V/Attribute/*/Angle E/Supportive I/Relationship(partToWhole) E/Adversative Q/Logical(not) Q/Qualitative(accurate)]***
"A logarithmic scale could suggest a focus on percentage changes, and reduce focus on absolute values." (*c*) ***[V/Coordinate/Cartesian/Scale/Numeric(log) E/Supportive U/Ability/Perceive(change) I/Trend(time) E/Adversative U/Ability/Perceive(measure)]***
"Structural properties include the domain sets and their functional relationships." (*d*) ***[D/Schema E/Declarative D/Domain E/Copulative I/Relationship]***
"Single-axis composition aligns two sentences that have identical horizontal or vertical axes." (*d*) ***[Q/Numeric(1) V/Component/Axis I/Composition V/Operation/Position(aligned) Q/Numeric(2) D E/Prepositional(in) E/Equal V/Component/Axis(h) E/Copulative V/Component/Axis(v)]***
"If a system detects that the user has a low need for cognition, it can offer help in interpreting the visualizations during this task." (*e*) ***[U/Ability E/Inequality(less) E/Causal Insight/Guide E/Possessive V U/Task/Process]***
"The output devices also influence the decision about which visualization to use. Traditional 2D displays are not the only representation means available, as recently 3D displays and VR equipment are being used in more and more applications." (*f*) ***[Device/Display E/Causal U/Task/DecisionMaking E/Possessive Q/Qualitative(useful) V]***

"The topology of a mesh consists of discrete points and connections and can be visualized as a node-link diagram." (*g*) ***[D/Attribute/Measurement(discrete) V/Element/Point E/Copulative V/Attribute/*/Connection V/Operation/Mapping V/Class/graph V/Element/Node V/Element/Edge]***
"Classification of continuous data could be viewed as a decision tree however, by doing this, the data is segregated into discrete categories." (*g*) ***[U/Task/Classify D/Attribute/Measurement(continuous) D/Operation/Slice D/Attribute/Measurement(discrete) D/Attribute/Measurement(nominal)]***
"If we think of the data as a list of cities or locations and their populations, then the data is discrete and can be visualized with methods such as a bar chart or a map with glyphs indicating population." (*g*) ***[D/Attribute/Measurement(discrete) V/Operation/Mapping V/Class/bar V/Class/map E/Preopositional(with) V/Element/Glyph]***
"To simplify visualizations, remove redundancy in properties, while ensuring that the reader can discriminate between the different visualization properties, such as shape, color, and thickness." (*h*) ***[Q/Qualitative(simple) V E/Causal Q/Logical(not) Q/Qualitative(redundant) V/Attribute E/Copulative U/Ability/Perceive(differentiate) V/Element/Shape V/Attribute/*/Color V/Attribute/*/Weight]***
"When necessary, multi-dimensional data can be visualized in 2D space by changing colors, shapes, and sizes to represent other data dimensions (e.g. contour plots) or by slicing the dataset, though too much variation can overcomplicate the plot." (*h*) ***[D/Schema(multid) V/Operation/Projection V/Class/2d E/Supportive V/Operation/Mapping V/Attribute/*/Color V/Element/Shape V/Attribute/*/Size E/Alternative Data/Operation/Slice E/Adversative D/Attribute/Distribution E/Causal Q/Logical(not) Q/Qualitative(easy) V]***
"Other alternatives for displaying multi-dimensional data are coplots (Cleveland, 1994), which visualize three variables in 2D space, and scatter plot matrices, which display a matrix of 2D scatter plots for any number of variables." (*h*) ***[E/Alternative D/Schema(multid) E/Conditional Q/Numeric(3) Data/Attribute E/Prepositional(in) D/Schema(2d) V/Organization E/Copulative V/Element/Point V/Attribute/*/Position V/Class/xy V/Organization]***
"Selecting attributes to use within a plot is especially important, because humans can quantify certain graph attributes better than others." (*h*) ***[D/Attribute V/Operation/Mapping E/Declarative Q/Qualitative(important) E/Causal Q/Existential(some) V/Attribute Q/Inequality(more) U/Ability/Perceive(measure)]***
"An alternative to heatmaps is horizon graphs (e.g. www.panopt icon.com), which display multiple time-series in parallel. Horizon graphs are similar to a time-series plot, but use color to highlight differences and extreme values within and across time-series." (*h*) ***[V/Attribute/*/Position V/Attribute/*/Color V/Class/xy E/Alternative V/Attribute/*/Length V/Attribute/*/Color V/Class/xy V/Organization E/Supportive Q/Numeric(N) D/VZ/Semantics(temporal) E/Supportive I/Extrema]***
"Using a color scheme that matches the type of data will further support the purpose of a plot." (*h*) ***[V/Component/ColorPalette Q/Qualitative(familiar) D/Domain Expression/Supportive I/Message]***
"Overview first, zoom and filter, then details-on-demand." (*i*) ***[D/Size(large) U/Task/Process V/Interaction/Overview V/Interaction/Zoom V/Interaction/Filter V/Interaction/Detail]***
"Bar graph is one of the most common ways to show categorical data." (*j*) ***[V/Class/bar E/Declarative Q/Qualitative(familiar) D/Attribute/Measurement(nominal)]***
"Differences among categories doesn't look as dramatic in the symbols plots as they do in the bar graphs." (*j*) ***[U/Ability/Perceive(differentiate) D/Attribute/Measurement(nominal) E/Supportive V/Class/symbols E/Rank(less) V/Class/bar]***
"Lines can make it easier to see trends in time series." (*j*) ***[V/Class/line E/Supportive I/Trend(time) E/Prepositional(in) D/Schema(sequential)]***
"Some views show only summary statistics, such as median, whereas other views, such as histogram, show distribution in greater detail." (*j*) ***[Q/Existential(some) Visualization D/Operation/Aggregation E/Adversative V/Class/histogram E/Supportive D/Attribute/Distribution]***
"Cartesian graph of paired values of two variables, x and y. The values of x can be visually extracted by perceiving position along a scale, in this case the horizontal axis. The y values can be perceived in a similar manner." (*k*) ***[V/Coordinate/Cartesian V/Class/xy E/Declarative V/Operation/Position(x) E/Prepositional(in) V/Component/Axis(h) E/Copulative V/Operation/Position(y) E/Prepositional(in) V/Component/Axis(v)]***
"The intention of a user that Zooms in is to change his/her existing view of a set of information to display a larger or more detailed view of a particular region." (*l*) ***[V/Interaction/Zoom Q/Qualitative(useful) E/Inequality(more) I/Detail]***
"The Change-Metaphor action is primarily intended by a user to change the view of currently presented information (e.g., a map-based display) to an alternative view (e.g., a timeline display)." (*l*) ***[V/Interaction/Change-Metaphor E/Causal U/Ability/Perceive(change) V]***
"A bar-chart representation contains vertical and horizontal axes, a set of labels on each, and a set of rectangular bars or lines. Constraints include the fact that the axes are orthogonal, the order of elements along a nominal axis is free to vary but default to lexicographic, the bars are aligned relative to one axis and extend parallel to the other, the color of bars is free to vary, but their shape must be constant and their size is constrained by the need to refer to the axis." (*m*) ***[V/Class/bar E/Declarative V/Component/Axis(h) V/Component/Axis(v) V/Component/Label V/Element/Bar V/Element/Line V/Operation/Mapping V/Attribute/*/Color V/Attribute/*/Shape V/Attribute/*/Size]***
"The bars of charts express correspondence between elements of two axes." (*m*) ***[V/Class/bar V/Operation/Mapping]***
"Make numerical adjustments to the raw data to enhance your point, e.g. absolute values, vs percentage change." (*n*) ***[D/Operation/Transformation E/Examplar D/Operation/Ratio E/Supportive I/Message]***
"Label the chart, e.g. title, description, legends and source line." (*n*) ***[V/Component/Label E/Prepositional(in) V/Component/Title V/Component/Description V/Component/Source]***
"Use color and typography to accentuate the key message." (*n*) ***[V/Attribute/*/Color V/Attribute/*/Typography E/Supportive Q/Qualitative(accurate) I/Message]***
"Given a one-dimensional sequence of univariate data (only one value per data item), we can map the spatial data to one of the screen dimensions and the data value itself to either the other screen dimension (to from a line graph, see Figure 5.1) or to color of a mark or region along the spatial axis (to form a color bar)." (*o*) ***[D/Attribute/Semantics(spatial) E/Existential(1) D/Attribute/Measurement E/Conditional V/Operation/Mapping V/Class/xy V/Element/Line V/Attribute/*/Position E/Logical(or) V/Element/Bar V/Attribute/*/Color V/Attribute/*/Length]***

Full List of Guideline References

a. Noah's White Paper (https://www14.software.ibm.com/webapp/iwm/web/signup.do?source=swg-BA_WebOrganic&S_PKG=ov12766)
b. School of Data Blog (http://schoolofdata.org/2013/04/26/data-visualization-guidelines-by-gregor-aisch-international-journalism-festival/)

c. Yau, N., Data Points: Visualization that Means Something (1st ed.). Wiley Publishing, 2013.
d. Mackinlay, J., Automating the design of graphical presentations of relational information. ACM Trans. On Graphics, 5(2):110-141. Apr. 1986.
e. Conati, C. and Maclare, H., Exploring the role of individual differences in information visualization. AVI '08, 199-206, 2008.
f. Golemati, M.; Halatsis, C.; Vassilakis, C.; Katifori, A.; Lepouras, G., "A Context-Based Adaptive Visualization Environment," Information Visualization, 2006. IV 2006. Tenth International Conference on , vol., no., pp.62,67, 5-7 July 2006.
g. Tory, M., Moller, T., Rethinking Visualization: A High-Level Taxonomy, IEEE Symposium on Information Visualization, 2004, pp.151-158, 2004.
h. Kelleher, C., and Wagener, T., Short communication: Ten guidelines for effective data visualization in scientific publications. Environ. Model. Softw. 26, 6 (June 2011), 822-827, 2011.
i. Shneiderman, B., The eyes have it: A task by data type taxonomy of information visualizations, Proc. IEEE Visual Languages '96, pp. 336-343, 1996.
j. Yau, N., Data Points: Visualization that Means Something (1st ed.). Wiley Publishing, 2013.
k. Cleveland, W.S. and McGill, R., Graphical Perception: Theory Experimentation and Application to the Development of Graphical Models, Journal of the American Statistical Association, Vol. 79, No. 387. (Sep., 1984), pp. 531-554, 1984.
l. Gotz, D., Zhou, M.X., Characterizing users' visual analytic activity for insight provenance, IEEE VAST '08, pp.123,130, 19-24 Oct. 2008.
m. Roth, S., Kolojejchick, J., Mattis, J., and Goldstein, J., Interactive graphic design using automatic presentation knowledge. In Proceedings of the SIGCHI, 112-117. 1994.
n. Wong, D., The Wall Street Journal Guide to Information Graphics: The Dos and Don'ts of Presenting Data, Facts, and Figures (1st ed.) W. W. Norton & Company, 2013.
o. Ward, M., Grinstein, G., and Keim, D., Interactive Data Visualization: Foundations, Techniques, and Applications. A. K. Peters, Ltd., 2010.
p. http://moz.com/blog/data-visualization-principles-lessons-from-tufte
q. Steele, J., and Iliinsky, N., Beautiful Visualization: Looking at Data Through the Eyes of Experts (1st ed.). O'Reilly Media, Inc, 2010.
r. Elzer, S., Carberry, S., and Zukerman, I.. 2011. The automated understanding of simple bar charts. Artif. Intell. 175, 2 (February 2011), 526-555.
s. Nazemi K., Stab, C., and Kuijper, A., A Reference Model for Adaptive Visualization Systems, Human-Computer Interaction: Design and Development Approaches, volume 6761 of Lecture Notes in Computer Science, page 480-489. Springer, (2011).
t. Stolte, C., Tang, D., and Hanrahan, P., Multiscale Visualization Using Data Cubes. IEEE Symposium on Information Visualization, October 2002
u. Battiti, R. and Brunato, M., Reactive Business Intelligence. From Data to Models to Insight. Reactive Search Srl, February 2011.
v. Bertini, E., and Lalanne, D., Investigating and reflecting on the integration of automatic data analysis and visualization in knowledge discovery. SIGKDD Explor. Newsl. 11, 2 (May 2010), 9-18, 2010.
w. Polowinski, J., and Voigt, M., VISO: a shared, formal knowledge base as a foundation for semi-automatic infovis systems. In CHI '13 Extended Abstracts on Human Factors in Computing Systems (CHI EA '13). ACM, New York, NY, USA, 1791-1796, 2013.
x. Voigt, M. et al., Capturing and Reusing Empirical Visualization Knowledge. In 1st International Workshop on User-Adaptive Visualization, 2013.
y. Gotz, D. and Wen, Z., Behavior-driven visualization recommendation. IUI '09. ACM, New York, NY, USA, 315-324, 2009.
z. Casner, S. M., Task-analytic approach to the automated design of graphic presentations. ACM Trans. Graph. 10, 2 (April 1991), 111-151, 1991.