Info Vis

Research Paper

Information Visualization
10(4) 271–288
© The Author(s) 2011
Reprints and permissions:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/1473871611415994
ivi.sagepub.com
SAGE

Research directions in data wrangling: Visualizations and transformations for usable and credible data

Sean Kandel[1], Jeffrey Heer[1], Catherine Plaisant[2], Jessie Kennedy[3], Frank van Ham[4], Nathalie Henry Riche[5], Chris Weaver[6], Bongshin Lee[5], Dominique Brodbeck[7] and Paolo Buono[8]

Abstract

In spite of advances in technologies for working with data, analysts still spend an inordinate amount of time diagnosing data quality issues and manipulating data into a usable form. This process of 'data wrangling' often constitutes the most tedious and time-consuming aspect of analysis. Though data cleaning and integration are longstanding issues in the database community, relatively little research has explored how interactive visualization can advance the state of the art. In this article, we review the challenges and opportunities associated with addressing data quality issues. We argue that analysts might more effectively wrangle data through new interactive systems that integrate data verification, transformation, and visualization. We identify a number of outstanding research questions, including how appropriate visual encodings can facilitate apprehension of missing data, discrepant values, and uncertainty; how interactive visualizations might facilitate data transform specification; and how recorded provenance and social interaction might enable wider reuse, verification, and modification of data transformations.

Keywords

data cleaning, data quality, data transformation, uncertainty, visualization

The elephant in the room

Despite continued advances in data management technologies, it remains tedious to examine a newly acquired data set and 'wrangle' it into a form that allows meaningful analysis to begin. First, an analyst must diagnose the data. Are the data responsive to the current analysis questions? What format are they in, and how much effort is required to put them into a format expected by downstream analysis tools? Are there data quality issues, such as missing data, inconsistent values, or unresolved duplicates? Next, the analyst must decide whether to continue working with the data, and, if so, the data must be transformed and cleaned into a usable state.

Our own informal interviews with data analysts have found that this process of assessment and transformation constitutes the most tedious component of their analytic process. Others estimate that data cleaning accounts for up to 80% of the development time and cost in data warehousing projects.[1] Often this process requires writing idiosyncratic scripts in programming languages such as Python, Perl, and R, or engaging in tedious manual editing using tools such as Microsoft Excel. Perhaps more significantly, this hurdle probably discourages a large number of people from working with data in the first place. The end result is that domain experts regularly spend more

[1]Computer Science Department, Stanford University, USA.
[2]Human-Computer Interaction Lab, University of Maryland, USA.
[3]Institute for Informatics & Digital Innovation, Edinburgh Napier University, UK.
[4]Center for Advanced Studies, IBM France.
[5]Microsoft Research, Redmond, USA.
[6]School of Computer Science, University of Oklahoma, USA.
[7]University of Applied Sciences Northwestern Switzerland, CH.
[8]Dipartimento di Informatica, Università degli Studi di Bari Aldo Moro, Italy.

Corresponding author:
Sean Kandel, Stanford University, 559, 2nd St, San Francisco, CA 94107, USA
Email: skandel@stanford.edu

time manipulating data than they do exercising their speciality, while less technical audiences are needlessly excluded.

We define such data wrangling as *a process of iterative data exploration and transformation that enables analysis*. One goal is to make data *usable* – to put them in a form that can be parsed and manipulated by analysis tools. Data usability is determined relative to the tools by which the data will be processed; such tools might include spreadsheets, statistics packages, and visualization tools. We say data are *credible* if, according to an analyst's assessment, they are suitably representative of a phenomenon to enable productive analysis. Ultimately, data are *useful* if they are usable, credible, and responsive to one's inquiry. In other words, data wrangling *is the process of making data useful*. Ideally, the outcome of wrangling is not simply data; it is an editable and auditable transcript of transformations coupled with a nuanced understanding of data organization and data quality issues.

The database community has developed numerous techniques for cleaning and integrating data. Most of this research focuses on specific data quality problems, such as resolving entities to remove duplicates.[2–5] Interactive visual tools have been introduced for tasks such as schema matching,[6] entity resolution,[7] and data cleaning.[8,9] However, most systems for working with data are non-interactive and inaccessible to a general audience, while those that are interactive make only limited use of visualization and direct manipulation techniques.

On the other hand, dirty and ill-formatted data constitute an 'elephant in the room' of visualization research: most visualization research assumes that input data arrive pristine, too often turning a blind eye to concerns of data formatting and quality. This disconnect suggests a research opportunity: data wrangling is a common impediment to analysis that visualization and interaction techniques could do much to alleviate. Data wrangling also constitutes a promising direction for visual analytics research,[10] as it requires combining automated techniques (e.g. discrepancy detection, entity resolution, semantic data type inference) with interactive visual interfaces.

In this article, we survey the problems, established approaches and research opportunities associated with data wrangling. Our hypothesis is that *we can advance the state of the art by enriching data-processing technologies with novel visual interfaces for data diagnostics and transformation*. In particular, we investigate how visualization and interaction techniques might improve analysts' abilities to diagnose and subsequently transform data, and chart a research agenda for both empirical and tools research in data wrangling. The overarching goal is to improve the efficiency and scale at which data analysts can work, while simultaneously lowering the threshold to enable broader audiences to engage with data.

Why we wrangle: Tales of effort and error

Nearly everyone who has taken on a serious data analysis effort has experienced the challenges of assessing data quality and modifying a data set to allow analysis to being in earnest. In this section, we review how the need for data wrangling arises. We begin with a hypothetical usage scenario representative of our experiences, and then enumerate sources of data problems.

A data wrangling scenario

John is tasked with analyzing 30 years of crime data collected by three different authorities. Accordingly, the data arrive in three different formats: one source is a relational database, another is a comma-separated values (CSV) file, and the third file contains data copied from various tables within a portable document format (PDF) report. Knowing the structure required for his visualization tool, John first reviews the different data sets to identify potential problems (step 1 in Figure 1).

The relational database allows him to specify a query and generate a file in an acceptable format. For the comma delimited data, the column headings associated with the data were unclear. Using spreadsheet software he adds a row of header information at the top to fit the format required by the visualization tool. While updating the header, John notices that the location of a given crime is encoded in one column (as 'City, State') in the CSV file and encoded in two columns (one 'City' column and one 'State' column) in the relational database. He decides to split the column in the CSV file into two separate columns. John then opens the text file in the spreadsheet but the spreadsheet does not parse the data as desired. After manually moving data fields to appropriate columns and some other manipulation (step 2), John finally has consistent columns and now combines the three files into one, but then notices that some columns have inconsistently formatted cells. The 'Date' column is formatted as 'dd/mm/yy' in some cells and as 'mm/dd/yyyy' in others. John returns to the original files, transforms all the dates to the same format, and recombines the files.

John loads the merged data file in a visualization tool (step 3). The tool immediately gives the error message 'Empty cells in column 3'; it cannot cope with missing data. John returns to the spreadsheet to fill in missing values using a few spreadsheet formulas (back to step 2). He edits the data by hand; sometimes he transforms the data (e.g. one state reports data only every other year so he uses an average for the missing

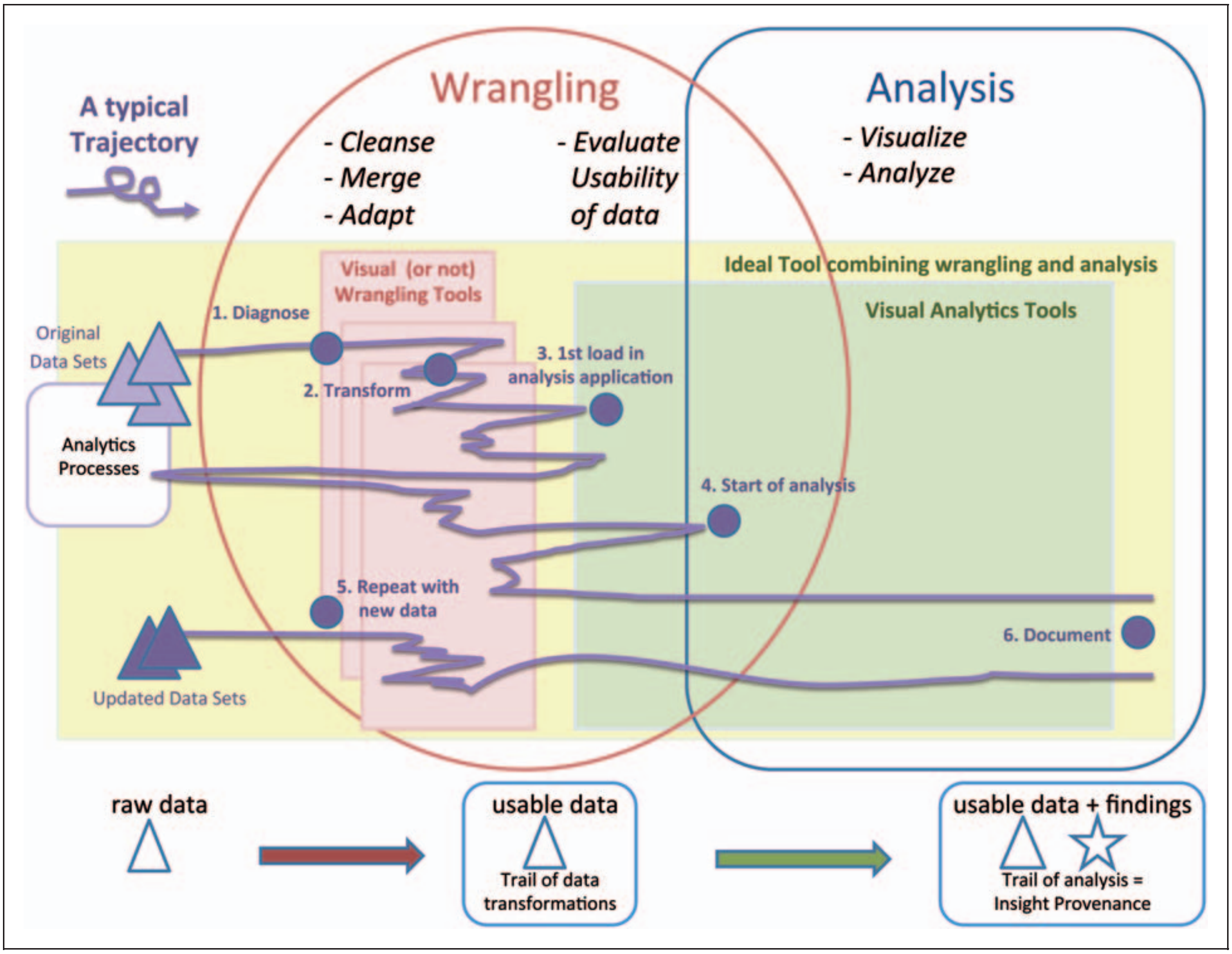


Figure 1. The iterative process of wrangling and analysis. One or more initial data sets may be used and new versions may come later. The wrangling and analysis phases overlap. While wrangling tools tend to be separated from the visual analysis tools, the ideal system would provide integrated tools (light yellow). The purple line illustrates a typical iterative process with multiple back and forth steps. Much wrangling may need to take place before the data can be loaded within visualization and analysis tools, which typically immediately reveals new problems with the data. Wrangling might take place at all the stages of analysis as users sort out interesting insights from dirty data, or new data become available or needed. At the bottom we illustrate how the data evolves from raw data to usable data that leads to new insights.

years). At other times there is nothing he can do after diagnosing a new problem (i.e. return to step 1). For example, he finds out that survey question 24 did not exist before 2000, and the most recent year of data from Ohio has not been delivered yet, so he tries to pick the best possible value (e.g. −1) to indicate missing values. John detects other, more nuanced, problems; for example, some cells have a blank space instead of being empty. It took hours to notice that difference.

John tries to follow a systematic approach when evaluating the data, but it is difficult to keep track of what he has inspected and how he has modified the data, especially because he discovers different issues across different files. Even after all of this work, he is not sure if he has examined all of the variables or overlooked any outliers. After a while, the data file seems good enough and he decides to move on.

It took a few days so it is with a great sense of accomplishment that John finally loads the data for the second time into the visualization tool he wants to use (step 3 again). He constructs several views of the data, including a geospatial representation of the crimes and a scatterplot of age against crime.

As soon as he sees the visualized data he realizes that, unfortunately, data quality issues still persist. Extreme outliers appear in the visualization. Some outliers seem to be valid data (e.g. data from the District of Columbia are very different from data from every other state). Others seem suspicious (criminals may vary in age from teenagers to older adults, but apparently babies are also committing crimes in certain states). John iteratively removes those outliers he believes to be dirty data (e.g. criminals under 7 and over 120 years old). Time-series visualizations indicate that, in 1995, some causes of death disappear abruptly while new ones appear. Two days later, an email exchange with colleagues reveals that the classification of causes of death was changed that year. John writes a transformation script to merge the data so he can analyze distinct terms referring to the same (or at least similar) cause of death.

Although the 'real' analysis is just about to start (step 4), John has made dozens of transformations, repeated the process several times, made important discoveries relating to the quality of the data, and made many decisions impacting the quality of the final 'clean' data. He also used visualization repeatedly while walking through the process, but still does not have results to show to his boss. Finally, he is able to work with the usable data, and useful insights come to the surface, but updated data sets arrive (step 5). Without proper documentation (step 6) of his transformations, John might be forced to repeat many of the tedious tasks.

The many sources of data problems

Many sources of error contribute to the types of problems described in the scenario above. Human error during manual data entry often includes entering incorrect or misleading default values. For instance, certain states may require data clerks to enter a criminal's age as an integer, even if the age is unknown. Clerks resort to entering arbitrary but legal values that have impossible interpretations, such as 0 or 999 for ages, resulting in erroneous ages.

Data from different sources often follow different conventions, formats, or data models. Integrating these sources into a common data model often requires not only manipulation of data formats, but also making other judgements to resolve incompatible schemas. Even within one data set, schemas may evolve over time or be misinterpreted by new users entering data. Classification systems may change, making it hard to compare categories over time. Finally, although automated data collection systems such as sensors can reduce errors in data entry, they introduce new types of uncertainty, such as inconsistencies in calibration or interference from outside sources.

The database, statistics and scientific workflow literature each offer several categorizations of the types and sources of errors. Li et al.[11] outline 41 different types of dirty data, and examine the costs of fixing these errors within different contexts. Kim et al.[12] propose a taxonomy of 33 dirty data types. These types fall into three broad categories: missing data (e.g. a state fails to report crime data for one year), incorrect data (e.g. incorrect criminal ages), and inconsistent representations of the same data (e.g. different encodings of crime location). They conclude that existing technologies address less than half of the dirty data types in their taxonomy and that 25 of the 33 types require some kind of human intervention. Müller and Freytag[13] roughly classify data anomalies into syntactical, semantic, and coverage anomalies. Syntactical anomalies are errors in data format and values. Semantic anomalies include inconsistencies within or across data sets (e.g. integrity constraint violations, contradictions, duplicates, and invalid tuples). Coverage anomalies refer to missing or incomplete data.

After identifying the source or type of error, analysts most likely need to transform their data. Data transforms generally fall within three categories: syntactic, structural and semantic transformations.[14] Syntactic transformations refer to parsing or reformatting data to ensure they can be read. Structural transformations refer to schema modifications. Semantic transformations refer to the meaning of the concepts and constraints in the schema, such as mapping causes of death across classifications in the data above. In many cases, these semantic transforms are not expressible in database languages or in the terms of low-level data models.

Identifying and correcting these different forms of dirty data may benefit from interactive visualizations; however, some types of dirty data prevent the direct application of traditional visualization tools. Novel visual interfaces for data transformation that are more robust to common data quality issues could help analysts identify and correct these types of errors.

We hypothesize that a tight coupling of data verification, transformation, and visualization can accelerate analysis and lead to more effective results. The analysis process often involves many iterations, as analysts generate hypotheses or develop insights that call for new data requirements. For instance, an insight may reveal the need to transform a data source to better suit an ensuing analysis task, or require assimilating additional data sets. The iterative nature of data wrangling suggests that the process might be facilitated by visual interfaces that intimately integrate both data diagnostics and a variety of data transformations.

However, how to best couple visualization, interaction, and algorithmic techniques remains unclear. Additional research questions arise around the effectiveness of different visual encodings for data wrangling and how to handle increased data sizes. We would also like any resulting data transformations to be amenable to reuse and refinement. To avoid reinventing the wheel, both cleaned data and wrangling transformations might be shared and evolved via the social web.

In the following sections, we outline the research challenges and opportunities that lie in applying visualization and interaction techniques to the problem of data wrangling, consider past related work, and identify areas for future research.

Diagnosing data problems

To make data useful for analysis, analysts must first identify any problems in their data. As stated above, there are dozens of possible 'errors' that can arise in data. We believe that tightly integrating visualization into the iterative process of wrangling will help unearth data quality issues. Visualizations can appropriately convey the 'raw' data and present the results of automated routines such as outlier detection. Different visual representations highlight different types of issues in the data; currently, this requires an analyst to select an appropriate progression of visualizations to view.

Visualizing 'raw' data

Some of the most common insights people gain using visualization are about data errors and outliers. Outliers often stand out in a plot, sometimes reducing the visibility of other data points owing to an extreme scale. Similarly, missing data may surface as a prominent gap or zero value in the data. Duplicate or misspelled values may appear adjacent to one another in a sorted list. Other errors may be more subtle, becoming apparent only when an appropriate transform is performed; for example, calculating an aggregate over individual demographics may not match a provided total because of a privacy-preserving redaction of some lower-level values.

A central concern in visualizing raw data is the choice of representation. Consider Figure 2, in which social network diagrams show data extracted from the Facebook web application programming interface (API). Figure 2(a) visualizes the data as a node-link diagram with a force-directed layout, revealing multiple clusters. Figure 2(b) shows the same data as a matrix diagram: rows and columns represent people, and filled cells represent a connection between them. Following best practices, automatic permutation of rows and columns has been applied to highlight patterns of connectivity. One sees clusters along the diagonal, including more substructure than can be seen in the node-link diagram.

However, for the purposes of data cleaning, the 'raw' visualization in Figure 2(c) is more revealing. The rows and columns are instead sorted in the order provided by the Facebook API. We now see a striking pattern: the bottom-right corner of the matrix is completely empty. Indeed, this is a missing data problem, as Facebook enforces a 5000-item result limit for a query. In this case, the maximum was reached, the query failed silently, and the mistake went unnoticed until visualized. As this example indicates, choices of representation (e.g. matrix diagram) and parameterization (e.g. default sort order) are critical to unearthing data quality issues.

Analysts need to be aware of the potentially misleading factors induced by visualizations. A natural starting point is a simple textual or tabular view of data: inspecting the raw values (or at least a subset) provides insight into data formatting and potential errors. In many cases, no other visualizations are applicable until the data are suitably transformed. However, as one restructures the data, additional visualizations can shed further light. For instance, the data cleaning tool Google Refine[8] uses histograms to aid inspection and outlier detection. However, the visualization chosen must also fit the semantics of the data. For example, an error in the encoding of geographic locations may not become apparent until plotted on a map. Once the data have been assessed in a 'raw' fashion, an analyst may move on to assess more abstract or transformed (e.g. aggregate) views of the data.

What forms of summary visualization best assist analysts as they profile their data? More research is needed to characterize the effectiveness of available visualization techniques for surfacing data quality issues across various data types. The results of this research might then be applied to suggest protocols for visual data diagnosis.

Scaling to large data sets

Another important concern is the issue of scale. As data set sizes become large, it becomes exceedingly difficult to visualize the 'raw' data. In response, researchers have invented techniques such as pixel-oriented visualizations[15] to increase data density while still showing individual values. However, this approach reaches an obvious breaking point when there are more data elements than pixels. Furthermore, in many cases perception breaks down much earlier; for example, with only a few hundred data points, overplotting can quickly render a scatter plot ineffective.

A common recourse is to apply aggregation, but doing so risks obscuring low-level details in the data. Histograms are a common form of one-dimensional (1D) aggregation, both for categorical data and for binned quantitative data. Binning is also applicable in scatter plots, for example to form a heat map visualizing data density (Figure 3). Statisticians have suggested numerous techniques for plotting data at scale,[16,17] including using hexagonal (as opposed to rectangular) two-dimensional (2D) bins in order to improve density estimates and de-emphasize horizontal and vertical striping.[16] A related issue is the judicious use of color: a naïve color ramp visualizing counts of data elements results in bins with few elements being practically invisible. Instead, a color ramp with a perceptible discontinuity between 0 and

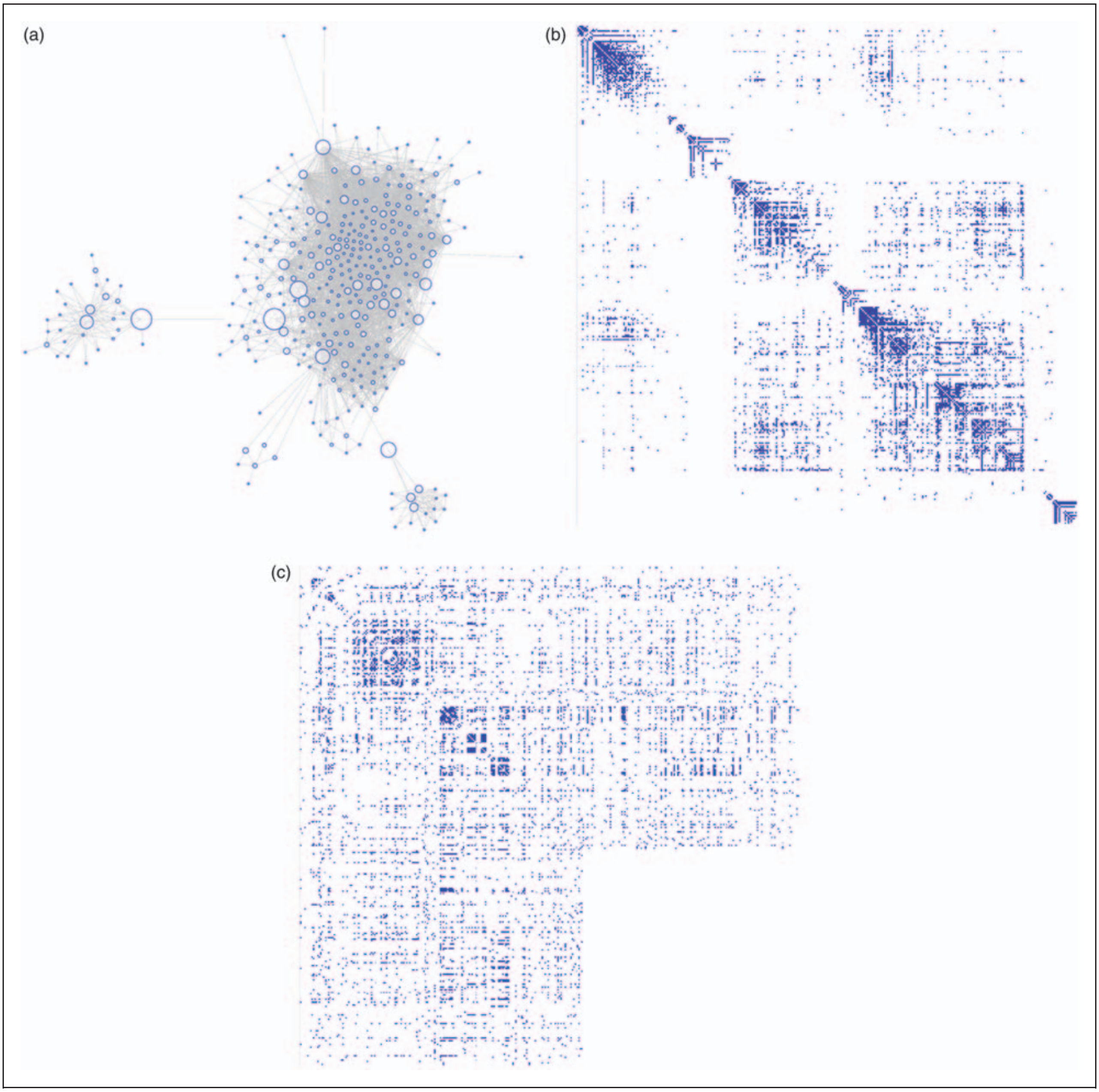


Figure 2. The choice of visual representation impacts the perception of data quality issues. (a) A node-link diagram of a social network does not reveal any irregularities. (b) A matrix view sorted to emphasize connectivity shows more sub-structure, but no errors pop out. (c) Sorting the matrix by raw data order reveals a significant segment of missing data.

1 allows viewers to quickly discern all cells containing non-zero values and thereby spot potential outliers or erroneous values.

Visual design techniques must also be coupled with interaction techniques. For example, one might assess if an outlier cell contains noteworthy values or merely errors by seeing how the points project along other data dimensions. How should we enable rapid linked selections (brushing and linking) over scalable summary visualizations?

Another approach to visualizing data at scale is sampling. Techniques such as online aggregation[18] might be applied: a visualization may show a dynamic aggregate of a sample, with error bars indicating a confidence interval. As query processing continues, the visualization can update the computed values and intervals; the analyst need not wait until completion to assess the data and proceed to other tasks. While initially proposed for 1D quantitative data, such dynamic sampling-based techniques might be

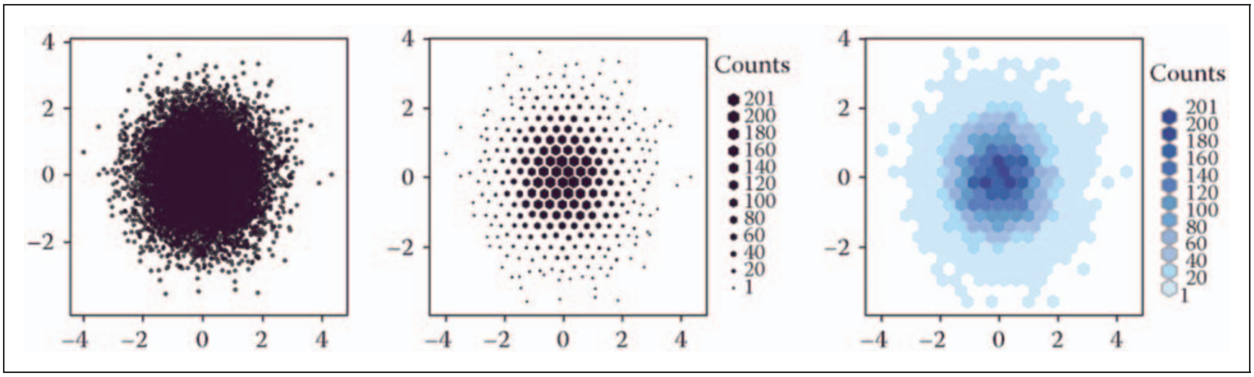


Figure 3. Visualizing 'raw' data at scale, taken from Carr et al.[16] (a) A traditional scatter plot. (b) A binned plot using a size encoding. (c) A binned plot using a color encoding. Note the discontinuity in color between 0 and 1, making cells with a single element readily apparent.

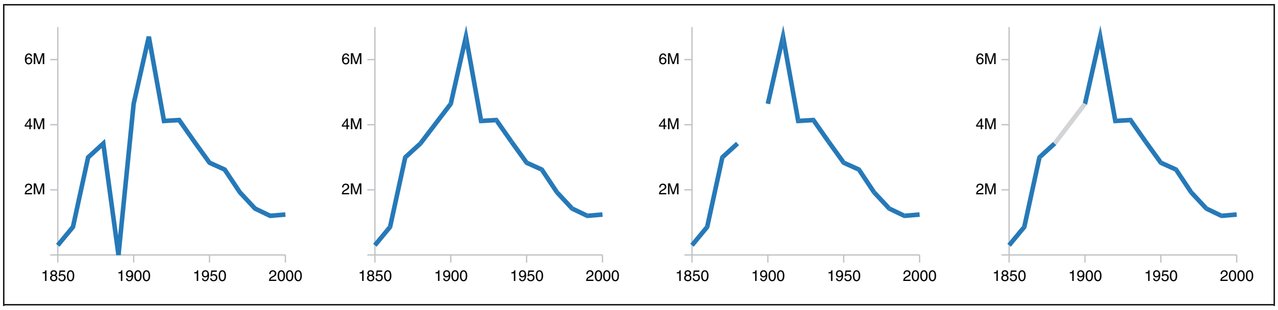


Figure 4. Alternative representations of missing data in a line chart. The data are U.S. census counts of people working as 'Farm Laborers'; values from 1890 are missing due to records being burned in a fire. (a) Missing data is treated as a zero value. (b) Missing data is ignored, resulting in a line segment that interpolates the missing value. (c) Missing data is omitted from the chart. (d) Missing data is explicitly interpolated and rendered in gray.

extended to other data types. More research is necessary to characterize the strengths and limits of such approaches.

Visual assessment and specification of automated methods

Although our discussion has focused primarily on visualization, statisticians and database researchers have developed a number of analytic techniques for assessing data quality. These techniques include algorithms for detecting outliers and discrepancies.[19,20] Other approaches range from simple validation routines (e.g. regular expression patterns) to complex data mining algorithms. How might we use visualization to best communicate the results of these routines? How can visual interfaces be used to specify or steer appropriate routines based on the semantics of the data? Can visualizations also serve as an input device for authoring new validation patterns? Moreover, we might evolve these algorithms, using approaches such as active learning,[5] so that they can improve in response to guidance and verification from analysts. These questions present important research challenges requiring the combination of data wrangling, visualization, and analysis methods.

Living with dirty data

Visualization can be a powerful tool for identifying data quality issues. However, once found, it is not always clear how (or even whether) one should modify the data in response. In fact, some may wish to proceed with visual analysis despite the presence of missing data, outliers, or other inconsistencies. Such actions naturally raise the question: how can visualizations be best designed to support reasoning with dirty or uncertain data? As in data diagnosis, one would like errors such as missing data to be visibly indicated. However, unlike data diagnosis, one may wish to reduce this visual saliency so as not to unduly distract from analysis of the rest of the data.

Visualizing missing data

What forms of visual encoding or annotation should be used to flag known data quality issues during visual analysis? A small amount of prior work has investigated this question. The example in Figure 4 shows alternative representations of missing data in a line chart. In spatial domains, such as maps or fluid flows, color interpolation techniques might be applied. For example, Restorer[21] maintains smooth luminance contours but drops hue to unobtrusively show missing values. In contrast, space-filling visualizations such as pie charts or treemaps may obscure the presence of missing data and bias the appearance of other items.

Eaton et al.[22] categorize visualization techniques based on how amenable they are to revealing missing data and compare design variants in a user study. They find that users do not necessarily realize that data are missing when they are replaced by default values. Cues that more explicitly highlight imputed elements can reduce the rate of error. They also find that, even if the missing data are noticeable, users regularly make general conclusions with the remaining partial data. This study provides evidence for a need to indicate the presence of missing information. However, we still lack a comprehensive answer to our design question. More work is needed to assess the design space of visual encodings of missing data and the impact on dependent analysis tasks.

Visualizing uncertain data

Much of the research on visualizing uncertainty has been in the fields of geographic visualization and scientific visualization. MacEachren et al.[23] report a review of models of information uncertainty with the goal of informing visualizations for geospatial information analysis. The list of challenges includes 'understanding the components of uncertainty and their relationships to domains, users, and information needs', 'developing methods for depicting multiple kinds of uncertainty', and 'developing methods and tools for interacting with uncertainty depictions'. MacEachren et al. also caution that uncertainty has been defined in many different ways and is referred to inconsistently in a variety of fields. Skeels et al.[24] create a classification of uncertainty based on the review of existing work on uncertainty from several domains and an interview-based user study. In their classification, they identify five types of uncertainty: measurement precision, completeness, inference, disagreement, and credibility.

Uncertainty arises from a number of sources, including measurement errors (e.g. sensor drift), missing data, and sampling. Uncertainty can also accumulate when data are aggregated or transformed. Techniques for visualizing uncertain data[25–28] often employ special visual encodings, including transparency, blur, error bars, and error ellipses. Olston and Mackinlay[27] describe mechanisms for visualizing uncertain data with known bounds. CandidTree shows two types of structural uncertainty using color and transparency based on the differences between two tree structures.[29] Other techniques include adding glyphs (Figure 5),[25] adding or modifying geometry,[30] and animation.[31] Listen sonifies geometric uncertainty using sound to represent the difference between geometric quantities obtained by two interpolants.[32]

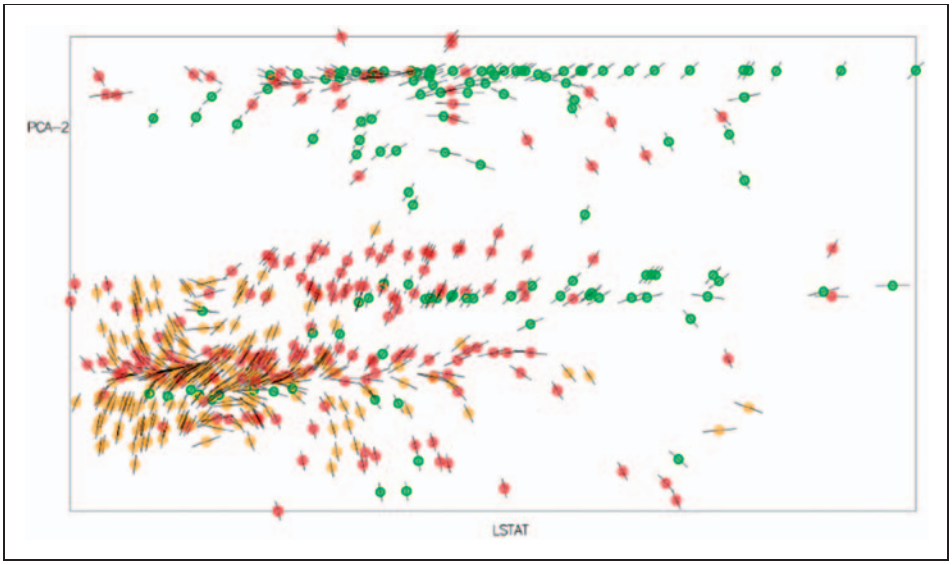


Figure 5. Visualizing Uncertainty. Correa et al.[25] add line segments to show sensitivity parameters to an input variable. Color encodes clustering with respect to a third variable; here we see a critical region where these sensitivities change sign.

How effective are these techniques? Kosara,[33] for instance, has found that people have difficulty identifying different levels of blur, implying that blur is a relatively ineffective encoding for multiple levels of uncertainty. It is important to note that most of the proposed solutions for visualizing uncertainty have not been empirically evaluated. The field would benefit from a deeper understanding of how these various representations of uncertainty affect perception and reasoning. Moreover, many techniques for handling uncertainty require choosing an underlying statistical model. Interactive visualization might aid in both selecting and evaluating such choices.

Adapting systems to tolerate error

Finally, the goal of living with dirty data suggests important criteria for visual analysis systems. Do the data models provided by our systems explicitly support missing values or values that deviate from a schema? For example, a collection of numbers with a few erroneous string values interspersed should not prevent a tool from visualizing most values along a numeric axis. In such cases, the visualization might also include an indication of the presence and amount of deviant data. More advanced systems might also consider the semantics of uncertainty when transforming data – for example, how uncertainty propagates across aggregations and other analysis routines[25,34] – and use this information to incorporate uncertainty into the visualization.

Transforming data

As we use visualizations to identify and represent data quality issues, we might also interact with the visualizations to correct those issues. A variety of data transforms may be needed throughout the wrangling process; examples include reformatting, extraction, outlier correction, type conversion, and schema mapping. In this section, we consider the interactive tasks that data wrangling systems need to support.

Data formatting, extraction, and conversion

One challenge of data wrangling is that reformatting and validating data requires transforms that can be difficult to specify and evaluate. For instance, splitting data into meaningful records and attributes often involves regular expressions that are error-prone and tedious to interpret.[35,36] Converting coded values, such as mapping Federal Information Processing Standards (FIPS) codes to US state names, may require integrating data from multiple tables.

Several interactive systems apply direct manipulation and programming by demonstration (PBD) methods to assist in specific cleaning tasks. Toped++[36] is an interface for creating *topes*, objects that validate data and support transformations such as text formatting and lookups. PBD systems infer a user's desired transform from examples provided via direct selection. SWYN[35] infers regular expressions from example text selections and provides visual previews to help users evaluate their effect. Potluck[37] applies simultaneous text editing[38] to merge data sources. Users of Karma[39] build text extractors and transformations for web data by entering examples in a table. Vegemite[40] applies PBD to integrate web data, automates the use of web services, and extends CoScripter[41] to generate shareable scripts. These systems introduce powerful tools to support text extraction and transformation, but they are insufficient for iterative data wrangling: each supports only a subset of needed transformations and lack operations such as reshaping data layout, aggregation, and missing value imputation.

Other work has introduced automated techniques for information extraction[42,43] or interactive interfaces to a more general transformation language. Potter's Wheel[9] provides a transformation language for data formatting and outlier detection. Ajax[44] contains facilities for data transformation and entity resolution. These tools enable a variety of transforms, including data reshaping and reformatting. However, both tools provide only limited support for direct manipulation: interaction is largely restricted to menu-based commands or entering programming statements.

Analysts could benefit from interactive tools that simplify the specification of data transformations. Can transformations be communicated unambiguously via simple interactive gestures over visualized data? If not, can relevant operations be inferred and suggested? Direct manipulation and PBD techniques might allow both expert and novice users to construct data transforms. However, users may have to provide a multitude of examples from which a system can infer appropriate extraction or transformation rules. Visualization and interaction techniques might help users find appropriate examples to contribute. Visualization might also reveal incorrect inferences by PBD routines, and users could update these examples interactively to improve inferred patterns and transforms. As an example, Wrangler[45] suggests relevant transforms based on the current context of interaction. Wrangler also provides visual previews of operations that are intended to facilitate rapid evaluation and refinement of suggested transforms.

Another common hurdle in data wrangling is converting data values to different types. An example is converting zip codes into the latitude–longitude centroids of their regions; a precomputed lookup table is sufficient to perform this conversion. Another is adjusting a currency for inflation or converting one currency to another. These transforms require multiple inputs, as they are parameterized by a specific date or year. These transforms could be facilitated by semantic data types that include parsing, validation, and transformation rules to aid data wrangling. Although a few data types occur regularly (e.g. dates and geographic locations), creating an exhaustive set of semantic data types a priori is infeasible. As we discuss later, peer production and sharing of new semantic data types by domain experts may provide one solution.

Correcting erroneous values

Once data found 'in the wild' have been extracted and reformatted, an analyst can begin assessing and correcting problematic values. Transforms of this type include outlier detection, missing value imputation, and resolving duplicate records.

Consider the problem of outlier detection: although automated outlier detection can highlight potential errors in the data, human judgement is often necessary to verify these errors and choose an appropriate transform to correct them. For instance, outlier detection might flag states with a large number of crimes (e.g. greater than three standard deviations from the mean) as errors. An analyst might assess whether this value is high because of some error (e.g. incorrectly entered into the database) or because it accurately reflects a real-world occurrence. After verifying that an error is in fact an error, there are still multiple ways to correct it. In this case the analyst could decide to remove only specific outliers or decide to set bounds on the data values. A better test of abnormality may be high crime despite low population. Existing errors can make it difficult to detect other errors; by cleaning errors as they are discovered, automated detection algorithms are generally more effective. A common example of this effect is masking – when an abnormally large value in a data set affects the modeled distribution so much that other extreme values appear 'normal'. In this case, an analyst could iteratively run outlier detection and transforms until he is satisfied with the results. Interaction is needed to accelerate these iterative loops of assessment and action.

Open questions concern how best to specify corrections. In the case of misspellings, text editing and batch updates may suffice. For missing values, filling in or interpolating nearby values are options. In the case of outlier correction, one could simply select and delete (or regress) values, but this may prove unsatisfying. Such an operation is highly specific to the selected data point(s); how might the transform generalize to cover new data as they arrive? Rather than make selections in data space, an alternative may be to make selections within a model space. For example, in addition to raw value ranges, a visualization may show standard deviations or quantiles of the data. Selections (perhaps with interactive 'snap to' quantile boundaries or increments of the standard deviation) could then be made with respect to a more robust model, rather than absolute value ranges. Future work should investigate what forms of constrained interaction with visualizations best support data wrangling.

Another common problem is entity resolution, or de-duplication. Duplicate records often arise within a data set, for example addresses or names representing the same entity may be expressed using different strings. A number of automated techniques have been proposed to perform entity resolution,[2–4,46] but eventually reconciling duplicate records requires human judgement as well, requiring an interactive interface. One example is Google Refine,[8] which leverages Freebase to enable entity resolution and discrepancy detection. Another example is D-Dupe system,[7] which helps users to perform entity resolution. Human input is used to improve the system's suggestions via active learning. The example of Figure 6 shows that two instances of George (W.) Fitzmaurice are correlated and may refer to the same person. Human judgement can help determine if these names refer to the same author.

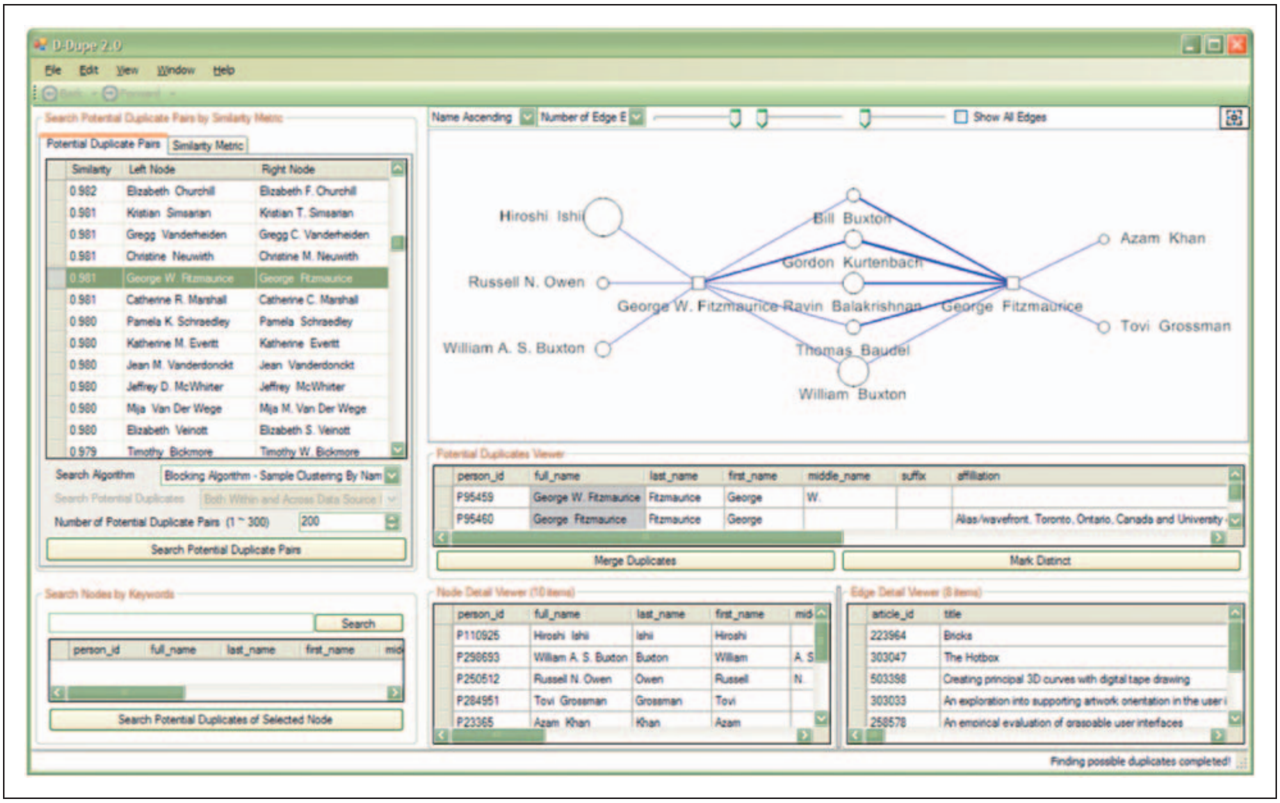
Future research might further improve and integrate such approaches.

Integrating multiple data sets

Analysis of data frequently requires integration of data from different sources. Integration requires being able to join or link data sets together along one or more shared dimensions. A number of the previously considered techniques contribute to the integration process: resolved entities or semantic data types may be used to match data together. A common subproblem is schema matching: mapping the dimensions of one data set onto the dimensions of another. However, even with matching data types, integration may be difficult. For example, how should one join sensor measurements taken at different time intervals?

To support this process, a number of algorithms[4,47–49] and interactive tools have been developed for data integration. Clio[50] uses semi-automated methods to help users map schemas. Schema Mapper[6] (Figure 7) adopts appropriate visualization and animation to enable more efficient navigation and mapping of large, complex schemas. One of the main problems addressed by Schema Mapper is the scalability of schema-to-schema connections.

Interfaces also allow users to choose between possible valid merged schemas.[51] A number of commercial ETL (extract, transform, load) tools contain



Similarity	Left Node	Right Node
0.982	Elizabeth Churchill	Elizabeth F. Churchill
0.981	Kristian Simsarian	Kristian T. Simsarian
0.981	Gregg Vanderheiden	Gregg C. Vanderheiden
0.981	Christine Neuwirth	Christine M. Neuwirth
0.981	George W. Fitzmaurice	George Fitzmaurice
0.981	Catherine R. Marshall	Catherine C. Marshall
0.980	Pamela K. Schraedley	Pamela Schraedley
0.980	Katherine M. Everitt	Katherine Everitt
0.980	Jean M. Vanderdonckt	Jean Vanderdonckt
0.980	Jeffrey D. McWhirter	Jeffrey McWhirter
0.980	Mija Van Der Wege	Mija M. Van Der Wege
0.980	Elizabeth Veinott	Elizabeth S. Veinott
0.979	Timothy Bickmore	Timothy W. Bickmore

Search Algorithm: Blocking Algorithm - Sample Clustering By Nam

Search Potential Duplicates: Both Within and Across Data Source

Number of Potential Duplicate Pairs (1 ~ 300): 200

person_id	full_name	last_name	first_name	mid

person_id	full_name	last_name	first_name	middle_name	suffix	affiliation
P95459	George W. Fitzmaurice	Fitzmaurice	George	W.		
P95460	George Fitzmaurice	Fitzmaurice	George			Alias/wavefront, Toronto, Ontario, Canada and University

person_id	full_name	last_name	first_name	mid
P110925	Hiroshi Ishii	Ishii	Hiroshi	
P298693	William A. S. Buxton	Buxton	William	A. S
P250512	Russell N. Owen	Owen	Russell	N.
P284951	Tovi Grossman	Grossman	Tovi	
P23365	Azam Khan	Khan	Azam	

article_id	title
223964	Bricks
303047	The Hotbox
503398	Creating principal 3D curves with digital tape drawing
303033	An exploration into supporting artwork orientation in the user i
258578	An empirical evaluation of graspable user interfaces

Figure 6. A network diagram produced by the D-Dupe[7] tool for entity resolution. Connections between clusters of suspected duplicates are shown among authors of ACM SIGCHI publications. Users can then interactively select which entities to merge. D-Dupe[7] allows users to perform entity resolution, here in a paper citation data set. On the left we see the list of potential duplicate author pairs that were identified based on the user-defined similarity metric. On the upper right the relational context viewer visualizes the coauthorship relation between the author pair selected in the duplicate viewer. The data detail viewer (lower right) shows all the attribute values of the nodes (authors) and edges (papers) displayed in the relational context viewer.

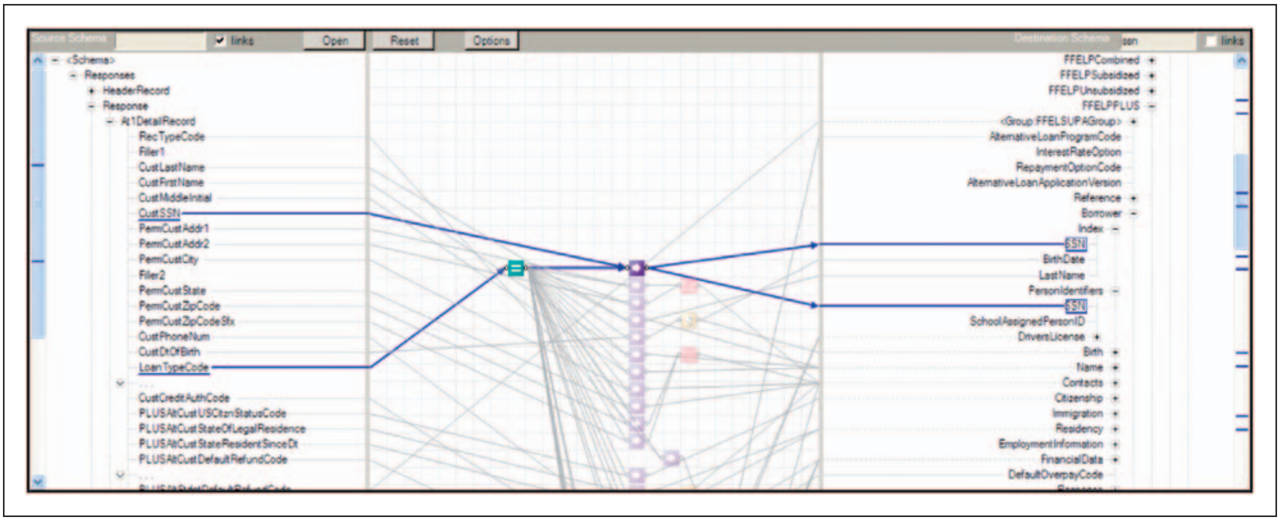


Figure 7. Robertson et al's[6] schema mapping visualization tool. The transformation mapping one XML schema to another is shown. The interface has three main sections: the left area shows the source schema and the right area shows the destination schema. The area in the middle shows the connections between source and destination. Schema Mapper introduces the concept of 'coalescence trees' to hide less interesting items. Notice that in the left part of the screen some items have been replaced by three dots. This visual cue indicates that there are hidden items that can be revealed moving the mouse pointer over the dots.

graphical interfaces for data integration.[52–54] Other interfaces[55] generalize copy and paste actions to integrate data. Future research might further investigate how visual interfaces and automated approaches to data integration could be more deeply combined.

Of course, some desired integrations are simply unattainable. Consider changes to category schemas: the passing of the North American Free Trade Agreement (NAFTA) led to the creation of a new classification system for companies in participating countries, the North American Industrial Classification System (NAICS). This scheme replaced the previously used (and increasingly antiquated) Standard Industrial Code (SIC). The dramatic reorganization of companies between the two systems leaves them nearly incomparable, as there are no reliable correspondences between high-level categories within the two taxonomies. Sometimes there is a limit to what one can wrangle.

Editing and auditing transformations

Transforming a data set is only one part of the larger data life cycle. As data update and schemas evolve, reuse and revision of transformation scripts becomes necessary. The importance of capturing data provenance is magnified when teams of analysts share data and scripts.

Existing research in visualization highlights the value of explicitly recording the provenance of an analysis. For example, the VisTrails[56] system provides a general infrastructure for authoring and reviewing visualization workflows. VisTrails maintains a detailed history for each workflow, including the insertion, deletion, and parameterization of visualization operators. However, VisTrails, along with most other visualization history tools,[57–60] focuses on analysis and does not support the process of data transformation necessary to use the visualization tools in the first place. More general scientific workflow tools[61–63] enable the creation and maintenance of workflows, but often by providing access to heterogeneous tools and scripting languages. Provenance-aware database systems[34] can track the lineage of data over multiple transformations and joins, but rarely support the steps necessary for transforming raw data into an appropriate format for import.

Informed by this prior work, we contend that the proper output of data wrangling is not just transformed data, but an *editable and auditable description* of the data transformations applied. High-level transformation descriptions will enable repeatability, modification, and recording of data provenance. Transforms could then be indexed and shared, enabling analysts to benefit from the work of others. Such transforms might also provide an artifact that can be annotated, enabling analysts to share their rationale for various data cleaning decisions.

Modification and reuse

Analysts frequently face the challenge of repeating a transformation process, whether due to the discovery of previously unnoticed errors, the arrival of new data, or changes to the data schema. In manual tools such as spreadsheet applications, this results in a great deal of tedious replicated effort. When using transformation scripts, simply rerunning a script is easy, but modifying it to handle changes to the data may be difficult or error-prone.

As a result, we believe an important requirement for data wrangling tools is not only to store a previously executed chain of data manipulations, but to facilitate interactive editing of transforms. Editing a transform may be necessary at multiple levels: one may wish to remove or insert additional operations, or refine the parameters within a particular step. Providing interactive transform histories are critical not only for repurposing existing scripts to meet new data demands, but also for enabling exploration of alternatives by skeptical analysts. With current tools, it can be difficult to determine if a provided data set has been manipulated in an unseemly fashion, perhaps done (un)consciously to hide the ‘flaws’ that might complicate an analyst or decision maker’s desired story.

An equally important part is not what data operations were performed, but why they were performed in the first place. A precise description of the changes made, and the rationales behind them, allows us to reconstruct the data wrangling process post hoc and assess the impact of each change on the data. Provenance is a common theme in modern data management[64]; although both origin and process are important in provenance, data wrangling generally concerns itself with the process. This part typically involves annotating the manipulations with metadata, such as the reason for performing the manipulation or the number of records affected by the manipulation. The combination of actions with rationale provides a richer picture of the data wrangling process and results.

Of course, data transforms (and their consequences) may be difficult to understand. For wrangling tools to be successful, transform histories must be quickly and accurately apprehended by auditors. Various techniques might be applied to reduce this gulf of evaluation.[65] For example, transform histories might be presented using natural language descriptions, enabling broader audiences to assess the transforms applied. This requires the development of techniques to enable both representation and manipulation of operators within a transformation. Moreover, we might develop visualizations that communicate the effects of various transforms (as in the Wrangler system[45]). Visual previews of transform effects should facilitate both the specification and review of data transformations.

Data transformation languages

One step towards achieving this goal is to create a declarative language for data transformation that provides a high-level representation of data transforms. Interactive operations performed by analysts within a visual interface could be mapped to statements in this language. Interactive wrangling would produce reusable scripts that can be modified to wrangle a new data set, inspected to communicate data provenance, and annotated to indicate an analyst's rationale. Using a high-level language also enables wrangling systems to generate code for a variety of platforms; for example, a transformation could be translated into a Python script or MapReduce code to run on a Hadoop installation. As a starting point, we might look to prior work from the database community,[9,66] which has developed expressive languages for data reformatting. We might extend these approaches with additional support for discrepancy detection and correction.

Along the way, we will need to assess how visual analytics tools might be designed in response to data wrangling needs. Should analysis tools take data sets only as input (as is typically done) or be extended to become 'provenance aware'? What is the right separation of concerns for tool modularity? System design questions arise both for lower-level performance issues – how to support rapid editing and rollback, for example by caching intermediate transformation states – and for user interface design – how might data transformations and annotations be surfaced in analysis tools to aid reasoning?

Wrangling in the cloud

One of the insights motivating our interest in data wrangling tools is that algorithms are not enough. Nuanced human judgements are often necessary throughout the process, requiring the design of interactive tools. One avenue for further reducing the costs associated with data preparation is to consider collaboration. To amortize wrangling costs and improve the scalability of data cleaning in the wild, we might cast data wrangling as an exercise in social computing.

Sharing data transformations

As a first step, we can consider how the wrangling efforts of one analyst might be picked up and used by others. Indexing and sharing of data transformation scripts would allow analysts to reuse previous data wrangling operations, with the goals of saving time and improving data consistency. Transformation revisions submitted by other collaborators could improve the quality or reliability of shared transforms. By deploying wrangling tools on the public web, a large audience (analysts, journalists, activists, and others) might share their transformations, and thereby further open data access. Research challenges arise in how to search for, present, and suggest transformations, or transformation subsets, developed by others.

Mining records of wrangling

While the sharing of individual scripts has a clear utility, additional benefits might arise from analyzing a large corpus of wrangling scripts. For example, one could analyze data set features (e.g. data types, columns names, distributions of values) to learn mappings to probable transformations or infer higher-level semantic data types. These data could lead to better automatic suggestions.[56] Such a corpus would also be a valuable resource for studying data cleaning strategies and informing the iterative design of wrangling tools.

User-defined data types

Another opportunity lies in providing mechanisms for user-contributed type definitions: how can we best enable data domain experts to define new semantic data types? Analysts might author and share domain-specific data type definitions enabling verification, reformatting, and transformation (e.g. mapping between zip codes and latitude–longitude pairs). Incorporating domain-specific knowledge can improve validation and might also facilitate data integration. Though type authoring is probably feasible for only a cadre of advanced users, a broad class of analysts might benefit by applying those types to their data. We might look for guidance from existing systems for end-user authoring of data reformatting and validation rules.[36]

Feedback from downstream analysts

Finally, we can consider how data quality might be improved by social interactions occurring across different phases of the data life cycle. Although data wrangling typically seeks to improve data quality prior to more sustained analyses, inevitably the process will be imperfect. Downstream analysts or visualization users, who might not have been involved in the initial data preparation, may also discover data errors. Indeed, such discoveries appear to be a common occurrence in social data analysis environments.[67,68] What interaction techniques might allow such users to annotate, and potentially correct, data quality issues discovered during subsequent analysis? How can these discoveries be fruitfully propagated into data transformation scripts and brought to the attention of other users of the data?

Conclusion

In this article we have examined the practical problems and challenges that regularly occur when an analyst tries to work with a real-world data set. Although data quality problems are commonplace and all of the authors have experienced them in one form or another, we found there was a gap in the literature concerning the challenges and potential solutions. In the previous sections we have highlighted broad research directions which, in our opinion, warrant further research (Table 1).

Future work should extend visual approaches into the data wrangling phase. Visualization can aid in the detection of potential problems in the raw data as a counterpart to fully algorithmic approaches. Ideally, we see a promising route in integrated approaches that allow a human to visually steer statistical algorithms. Visualization is also useful in the communication of data errors and uncertainties. When designing new visual metaphors we should always be mindful that our input data may not be pristine, and that our chosen visual encoding should indicate any missing values and data uncertainties. Finally, when it comes to correcting data errors, visual approaches could integrate with automated approaches to allow an interactive editing cycle.

We have argued that data wrangling should be made a first-class citizen in the data analysis process. Typical research papers tend to showcase the result of visualizing previously cleaned data, but more often than not neglect to mention how data errors were found and fixed. Ideally, the output of a wrangling session should be more than a clean data set; it should also encompass the raw data coupled with a well-defined set of data operations and potentially some metadata indicating why these operations were performed. These operations should be auditable and editable by the user. Secondary benefits of a high-level data transformation language include easier reuse of previous formatting efforts and an increased potential for social, distributed collaboration around data wrangling.

In current practice, wrangling often consists of manual editing and reshaping using a general purpose tool such as Microsoft Excel. Although this approach is feasible for smaller data sets, a great deal of effort is wasted on relatively menial tasks and no audit trails are stored. We expect that this way of working will become increasingly rare in the near future for two reasons. First, in an increasingly data-driven society we need auditable information on the data sets on which we intend to base our decisions. Without ways of explicitly storing the edits we make to a raw data set, we cannot guarantee that the pristine data we are looking at has not been substantially altered and is thus no longer credible. Second, as the number and size of data sets continues to grow, a completely manual approach will become infeasible. Currently, in order to work with many data analysis tools an analyst also needs to have significant expertise in programming and/or scripting in addition to the domain knowledge needed to make sense of the data. If we do not address this issue in the near future, we run the risk of disenfranchising the very domain experts on whom we depend for our data analysis.

In the previous sections we have provided a number of potentially interesting research directions, but there are some challenges that we have not touched upon. One obvious shortcoming is that there is very little empirical work that studies how day-to-day users wrangle with their data. To confirm and gauge the importance of data wrangling, and to inform the design of novel wrangling tools, it might be useful to collect data on how data cleaning is currently performed, both in the information visualization community and elsewhere. A survey could ask researchers and practitioners to report on the amount of effort spent on data wrangling, the tools they use, successes and failures, and if they are in agreement with the directions proposed in previous sections.

The VAST challenge[69] might provide a valuable resource for studying and comparing how users deal with data wrangling issues. Many of the challenges require participants to preprocess data in order to conduct their analysis. For example, the 2010 entries for the contest mentioned a multitude of manual approaches for cleaning the data, leading to different end results. Similar to the research literature, little was said in the entries about how this was done, and only a few people reported on how long it took to clean the data.

Throughout this article we have assumed that our data are stored in a structured format. Although this is increasingly the case for many data sources, there are also plenty of cases where the data are stored in a format that is not directly amenable to computational analysis. Data dissemination (especially by governments) has traditionally been done in the form of printed reports, and there are still data providers that consider a PDF scan of a report a digital version of the data. There is no simple answer to these types of problems and turning an unstructured raw file into a structured format typically involves a lot of manual work.

Finally, we wish to reiterate that there is not one definition of ‘clean data’ and that overly cleaned data are probably just as problematic as dirty data. For this reason we always have to be aware of how data operations we perform could affect the outcome of

Table 1. Research directions in data wrangling and the sections in which they are discussed

No.	Wrangling step	Research challenge
3	Diagnosing data problems	How to tightly integrate visualization in the iterative process of data wrangling?
3.1	Visualizing 'raw' data	What forms of summary visualizations best assist analysts as they profile their data?
3.2	Scaling to large data sets	How should we enable rapid linked selections over scalable summary visualizations, such as dynamic aggregate views?
3.3	Visual assessment and specification of automated methods	How might we use visualization to best communicate the results of analytic techniques for assessing data quality?
		How can visual interfaces be used to specify or steer analytic data quality algorithms based on the semantics of the data?
4	Living with dirty data	How can visualizations be best designed to support reasoning with dirty or uncertain data?
4.1	Visualizing missing data	What forms of visual encoding or annotation should be used to flag known data quality issues during visual analysis?
4.2	Visualizing uncertain data	How effective are the existing techniques to visualize uncertain data?
4.3	Adapting systems to tolerate error	Do the data models provided by visual analysis systems explicitly support missing values or values that deviate from a schema?
5	Transforming data	What interactive tasks do data wrangling systems need to support to correct data quality issues?
5.1	Data formatting, extraction, and conversion	How can data transformations for reformatting and validating data be specified and evaluated?
		How can conversions between data values of different types be facilitated by semantic data types?
5.2	Correcting erroneous values	What forms of constrained interaction with visualizations best support the specification of corrections?
		How can automated techniques for entity resolution be improved by human input?
5.3	Integrating multiple data sets	How can visual interfaces and automated approaches to data integration be more deeply combined?
6	Editing and auditing transformations	How can data provenance be captured and managed?
6.1, 6.2	Modification, reuse, and understanding of a transformation	How can interactive transform histories be used to represent, annotate and modify the data transformation process?
6.3	Data transformation languages	How to integrate visual interfaces with data transformation languages to aid discrepancy detection and correction?

(continued)

Table 1. Continued

No.	Wrangling step	Research challenge
		How should visual analytics tools be architected in response to data wrangling needs? Should they not only load data, but become provenance aware?
7	Wrangling in the cloud	How to cast data wrangling as an exercise in social computing?
7.1	Sharing data transformations	How to search for, present, and suggest transformations developed by others?
7.2	Mining records of wrangling	How to build, manage, and analyze a large corpus of wrangling scripts?
7.3	User-defined data types	How can we best enable data domain experts to define new semantic data types?
7.4	Feedback from downstream analysts	What interaction techniques might allow downstream users to annotate, and potentially correct, data quality issues discovered during subsequent analysis, and how can these discoveries be fruitfully propagated into data transformation scripts and brought to the attention of other users of the data?

an analysis. In developing sophisticated capabilities for data wrangling, we must be careful to define an 'error' not as an incompleteness, inconsistency, or incorrectness that is intrinsic to a particular data representation, but rather as a judgement of the suitability of that representation's format and semantics for data analysis.

In closing, we argue that data wrangling has long been an elephant in the room of data analysis. Extraordinary amounts of time are spent getting a data set into a shape that is suitable for downstream analysis tools, often exceeding the amount of time spent on the analysis itself. At the same time, all this effort is wasted when the data set changes and may be duplicated by many other analysts looking at the same data. Data cleaning cannot be done by computers alone, as they lack the domain knowledge to make informed decisions on what changes are important. On the other hand, manual approaches are time consuming and tedious. The principled coupling of visual interfaces with automated algorithms provides a promising solution, and we hope visual analytics research will contribute a lasting impact to this critical challenge.

Acknowledgements

Partial support for this work was provided by a Stanford Graduate Fellowship, NSF grant CCF-0964173, NSF 0947358 and NIH-NCI grant RC1CA147489-02. The authors also thank Martin Graham for his feedback.

References

1. Dasu T and Johnson T. *Exploratory Data Mining and Data Cleaning*. New York: John Wiley & Sons, Inc, 2003.
2. Bhattacharya I and Getoor L. Collective entity resolution in relational data. *ACM Trans Knowl Discov Data* 2007; 1(1): 5.
3. Elmagarmid AK, Ipeirotis PG and Verykios VS. Duplicate record detection: A survey. *IEEE Trans Knowl Data Eng* 2007; 19(1): 1–16.
4. Gravano L, Ipeirotis PG, Jagadish HV, Koudas N, Muthukrishnan S, Pietarinen L and Srivastava D. Using q-grams in a dbms for approximate string processing, 2001.
5. Sarawagi S and Bhamidipaty A. Interactive deduplication using active learning. Proceedings of ACM SIGKDD (Edmonton, Alberta, Canada), 2002.
6. Robertson GG, Czerwinski MP and Churchill JE. Visualization of mappings between schemas. *ACM CHI* 2005; 431–439.
7. Kang H, Getoor L, Shneiderman B, Bilgic M and Licamele L. Interactive entity resolution in relational data: A visual analytic tool and its evaluation. *IEEE Trans Visual Comput Graph* 2008; 14: 999–1014.
8. Huynh D and Mazzocchi S. *Freebase GridWorks*. http://code.google.com/p/google-refine/
9. Raman V and Hellerstein JM. Potter's wheel: An interactive data cleaning system. *VLDB* 2001; 381–390.
10. Thomas JJ and Cook KA. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. National Visualization and Analytics Center, 2005.
11. Li L, Peng T and Kennedy J. Improving data quality in data warehousing applications. *12th International Conference on Enterprise Information Systems* 2010.
12. Kim W, Choi BJ, Hong EK, Kim SK and Lee D. A taxonomy of dirty data. *Data Min Knowl Discov* 2003; 7: 81–99.

13. Müller H and Freytag JC. Problems, methods and challenges in comprehensive data cleansing. *Technical Report HUB-IB-164*, ed. Humboldt-Universität zu Berlin. Berlin: Institut für Informatik, 2003.
14. Ludscher B, Lin K, Bowers S, Jaeger-Frank E, Brodaric B and Baru C 2005; Managing scientific data: From data integration to scientific workflows.
15. Keim DA. Designing pixel-oriented visualization techniques: Theory and applications. *IEEE Trans Visual Comput Graph* 2000; 6: 59–78.
16. Carr DB, Littlefield RJ, Nicholson WL and Littlefield JS. Scatterplot matrix techniques for large N. *J Am Stat Assoc* 1987; 82: 424–436.
17. Utwin A, Theus M and Hofmann H. *Graphics of Large Datasets: Visualizing a Million*. Springer, 2006.
18. Hellerstein JM, Haas PJ and Wang HJ. Online aggregation. *ACM SIGMOD* 1997; 171–182.
19. Hellerstein JM. *Quantitative Data Cleaning for Large Databases*. White Paper. United Nations Economic Commission for Europe, 2008.
20. Hodge V and Austin J. A survey of outlier detection methodologies. *Artif Intell Rev* 2004; 22: 85–126.
21. Twiddy JC and Shiri SM. Restorer: A visualization technique for handling missing data. *IEEE Visualization* 2004; 212–216.
22. Eaton C, Plaisant C and Drizd T. The challenge of missing and uncertain data. *VIS '03: 14th IEEE Visualization 2003 (VIS'03)*. Washington, DC: IEEE Computer Society, 2003, p.100.
23. MacEachren AM, Robinson A, Gardner S, Murray R, Gahegan M and Hetzler E. Visualizing geospatial information uncertainty: What we know and what we need to know. *Cartogr Geogr Inform Sci* 2005; 32: 139–160.
24. Skeels M, Lee B, Smith G and Robertson GG. Revealing uncertainty for information visualization. *Inform Visual* 2010; 9: 70–81.
25. Correa C, Chan YH and Ma KL. A framework for uncertainty-aware visual analytics. *IEEE Visual Analytics Science and Technology* 2009; 51–58.
26. Griethe H and Schumann H. The visualization of uncertain data: Methods and problems. *SimVis* 2006; 143–156.
27. Olston C and Mackinlay J. Visualizing data with bounded uncertainty. *IEEE Symposium on Information Visualization* 2002; 37–40. (Boston, MA).
28. Pang T, Wittenbrink CM and Lodha SK. Approaches to uncertainty visualization. *Vis Comput* 1997; 13: 370–390.
29. Lee B and Robertson GG. *Czerwinski M and Parr CS*. CandidTree: Visualizing structural uncertainty in similar hierarchies, 2007.
30. Grigoryan G and Rheingans P. Point-based probabilistic surfaces to show surface uncertainty. *IEEE Trans Visual Comput Graph* 2004; 10: 564–573.
31. Gershon ND. Visualization of fuzzy data using generalized animation. *Proceedings of the 3rd Conference on Visualization '92, VIS '92*, IEEE Computer Society Press: Los Alamitos, CA, 1992; 268–273.
32. Lodha SK, Wilson CM and Sheehan RE. Listen: Sounding uncertainty visualization. *Proceedings of the 7th Conference on Visualization '96, VIS '96*, IEEE Computer Society Press: Los Alamitos, CA, 1996; 189–196.
33. Kosara R. *Semantic Depth of Field Using Blur for Focus+Context Visualization*. PhD Thesis, Vienna University of Technology, Vienna, Austria, 2001.
34. Benjelloun O, Sarma AD, Halevy A and Widom J. Uldbs: Databases with uncertainty and lineage. *VLDB '06: 32nd International Conference on Very Large Data Bases* 2006; 953–964. (VLDB endowment).
35. Blackwell AF. SWYN: A visual representation for regular expressions. *Your Wish Is My Command: Programming by Example* 2001; 245–270.
36. Scaffidi C, Myers B and Shaw M. Intelligently creating and recommending reusable reformatting rules. *ACM IUI* 2009; 297–306.
37. Huynh DF, Miller RC and Karger DR. Potluck: Semi-ontology alignment for casual users. *ISWC* 2007; 903–910.
38. Miller RC and Myers BA. Interactive simultaneous editing of multiple text regions. *USENIX Technical Conference* 2001; 161–174.
39. Tuchinda R, Szekely P and Knoblock CA. Building mashups by example. *ACM IUI* 2008; 139–148.
40. Lin J, Wong J, Nichols J, Cypher A and Lau TA. End-user programming of mashups with vegemite. *IUI* 2009; 97–106.
41. Leshed G, Haber EM, Matthews T and Lau T. CoScripter: Automating & sharing how-to knowledge in the enterprise. *ACM CHI* 2008; 1719–1728.
42. Arasu A and Garcia-Molina H. Extracting structured data from web pages. *ACM SIGMOD* 2003; 337–348.
43. Soderland S. Learning information extraction rules for semi-structured and free text. *Mach Learn* 1999; 34: 233–272.
44. Galhardas H, Florescu D, Shasha D and Simon E. Ajax: An extensible data cleaning tool. *ACM SIGMOD* 2000; 590.
45. Kandel S, Paepcke A, Hellerstein J and Heer J. Wrangler: Interactive visual specification of data transformation scripts. *ACM Human Factors in Computing Systems (CHI)* 2011.
46. Benjelloun O, Garcia-Molina H, Menestrina D, Su Q, Whang SE and Widom J. Swoosh: A generic approach to entity resolution. *VLDB J* 2008.
47. Cafarella MJ, Halevy A, Wang DZ, Wu E and Zhang Y. Webtables: Exploring the power of tables on the web. *PVLDB* 2008; 1: 538–549.
48. Doan A, Madhavan J, Dhamankar R, Domingos P and Halevy A. Learning to match ontologies on the semantic web. *VLDB J* 2003; 12: 303–319.
49. Rahm E and Bernstein PA. A survey of approaches to automatic schema matching. *VLDB J* 2001; 10: 334–350.
50. Haas LM, Hernández MA, Ho H, Popa L and Roth M. Clio grows up: From research prototype to industrial tool. *ACM SIGMOD* 2005; 805–810.
51. Chiticariu L, Kolaitis PG and Popa L. Interactive generation of integrated schemas. *ACM SIGMOD* 2008; 833–846.
52. Altova. *Data Integration: Opportunities, Challenges, and Altova Mapforce*. White Paper. http://www.altova.com/whitepapers/mapforce.pdf, accessed July 2010.
53. CloverETL. *Cloveretl Overview.* http://www.cloveretl.com/products/designer, accessed July 2010.
54. Informatica. *The Informatica Data Quality Methodology: A Framework to Achieve Pervasive Data Quality through Enhanced Business–IT Collaboration.* http://www.informatica.com/downloads/7130-DQ-Methodology-wp-web.pdf, accessed July 2010.
55. Ives ZG, Knoblock CA, Minton S, Jacob M, Pratim P, Tuchinda TR, Luis J, Maria A and Gazen MC. Interactive data integration through smart copy & paste. *CIDR* 2009.
56. Callahan SP, Freire J, Santos E, Scheidegger CE, Silva CT and Vo HT. Vistrails: Visualization meets data management. *SIGMOD '06: 2006 ACM SIGMOD International Conference on Management of Data*. New York: ACM, 2006, pp.745–747.
57. Heer J, Mackinlay J, Stolte C and Agrawala M. Graphical histories for visualization: Supporting analysis, communication, and evaluation. *IEEE Trans Visual Comput Graph* 2008; 14: 1189–1196.
58. Jankun-Kelly T, Ma KL and Gertz M. A model and framework for visualization exploration. *IEEE Trans Visual Comput Graph* 2007; 13: 357–369.
59. Kreuseler M, Nocke T and Schumann H. A history mechanism for visual data mining. *IEEE InfoVis* 2004; 49–56.

60. Shrinivasan YB and van Wijk J. Supporting exploration awareness in information visualization. *IEEE Comput Graph Appl* 2009; 29: 34–43.
61. Chappell D. *The Workflow Way: Understanding Windows Workflow Foundation*.
62. Michener WK, Beach JH, Jones MB, Ludäscher B, Pennington DD, Pereira RS, Rajasekar A and Schildhauer M. A knowledge environment for the biodiversity and ecological sciences. *J Intell Inf Syst* 2007; 29: 111–126.
63. Oinn T, Addis M, Ferris J, Marvin D, Senger M, Greenwood M, Carver T, Glover K, Pocock MR, Wipat A and Li P. Taverna: A tool for the composition and enactment of bioinformatics workflows. *Bioinformatics* 2004; 20: 3045–3054.
64. Buneman P, Khanna S and Chiew Tan W. Data provenance: Some basic issues. *In Foundations of Software Technology and Theoretical Computer Science* 2000; 87–93. (Springer).
65. Norman DA. *The Design of Everyday Things*. Basic Books, 2002.
66. Lakshmanan LVS, Sadri F and Subramanian SN. SchemaSQL: An extension to SQL for multidatabase interoperability. *ACM Trans Database Syst* 2001; 26(4): 476–519.
67. Heer J, Viegas F and Wattenberg M. Voyagers and voyeurs: Supporting asynchronous collaborative information visualization. *ACM CHI* 2007; 1029–1038.
68. Viegas FB, Wattenberg M, McKeon M, Ham FV and Kriss J. Harry Potter and the meat-filled freezer: A case study of spontaneous usage of visualization tools. *HICSS '08: 41st Annual Hawaii International Conference on System Sciences*. Washington, DC: IEEE Computer Society, 2008, p.159.
69. Costello L, Grinstein G, Plaisant C and Scholtz J. Advancing user-centered evaluation of visual analytic environments through contests. *Inform Visual* 2009; 8: 230–238.