FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

# Visual Analytics for Rule-Based Quality Management of Multivariate Data

DIPLOMARBEIT

zur Erlangung des akademischen Grades

**Diplom-Ingenieur**

im Rahmen des Studiums

**Visual Computing**

eingereicht von

**Florian Spechtenhauser BSc.**
Matrikelnummer 0826226

an der Fakultät für Informatik
der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Univ.-Doz. Dipl.-Ing. Dr.techn. Eduard Gröller
Mitwirkung: Dipl.-Ing. Dr.techn. Harald Piringer

Wien, 15. August 2016

| Florian Spechtenhauser | Eduard Gröller |
|---|---|

FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

# Visual Analytics for Rule-Based Quality Management of Multivariate Data

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

**Diplom-Ingenieur**

in

**Visual Computing**

by

**Florian Spechtenhauser BSc.**
Registration Number 0826226

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Ao.Univ.Prof. Univ.-Doz. Dipl.-Ing. Dr.techn. Eduard Gröller
Assistance: Dipl.-Ing. Dr.techn. Harald Piringer

Vienna, 15th August, 2016

Florian Spechtenhauser

Eduard Gröller

# Erklärung zur Verfassung der Arbeit

Florian Spechtenhauser BSc.
Buchengasse 34/3/22, 1100 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbstständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 15. August 2016

Florian Spechtenhauser

# Danksagung

Mein größter Dank gilt meinen Eltern Luis und Gabi, sowie meiner Schwester Romina, die es mir ermöglicht haben, dieses Studium in Wien zu absolvieren. Sie haben mich stets beim Studium unterstützt, sei es moralisch als auch finanziell. Besonderer Dank gilt meiner Freundin Judith, die mich jeden Tag auf meinem Weg durch das Studium und die Arbeit begleitet und mich immer wieder unterstützt und motiviert.

Ich möchte mich auch bei Harald Piringer vom Forschungszentrum VRVis bedanken, der es mir ermöglicht hat, beim VRVis diese Diplomarbeit zu erarbeiten, und mich sehr gut betreut und immer wieder wertvolle Inputs gegeben hat. Weiters bedanke ich mich bei Meister Eduard Gröller für seine Betreuung und Begleitung dieser Diplomarbeit.

# Kurzfassung

In der heutigen Zeit, in der die Menge an gesammelten und generierten Daten stetig ansteigt, ist das Sicherstellen einer ausreichenden Datenqualität ein entscheidendes Thema. Abhängig von der gegebenen Aufgabenstellung können sogar aufwändige Analysemethoden fehlschlagen oder irreführende Ergebnisse liefern, wenn der gegebene Datensatz eine unzureichende Qualität aufweist. Um Datenprobleme, wie zum Beispiel fehlende Werte oder Anomalien zu entdecken, werden häufig automatische Plausibilitätschecks verwendet, die auf definierten Regeln basieren.

Die Definition und Verwendung solcher Regeln und deren Ergebnisse stellt jedoch eine große Herausforderung dar. Visualisierung ist dabei ein mächtiges Tool, um unerwartete Datenqualitätsprobleme aufzudecken und die Ergebnisse der angewandten Regeln zu validieren. Visual Analytics schließt dabei die Lücke zwischen automatischer Datenanalyse und Visualisierung und hilft bei der Definition und Optimierung der Plausibilitätschecks, damit sie für eine wiederkehrende Analyse und Validierung der entdeckten Datenqualitätsprobleme verwendet werden können.

Diese Diplomarbeit besteht aus einer Design Study des Data Quality Overview, einem Visual Analytics Ansatz, der eine detaillierte und trotzdem skalierbare Übersicht über die Ergebnisse der definierten Plausibilitätschecks liefert, die über mehrere Detailstufen validiert und untersucht werden können. Der Ansatz basiert auf einer detaillierten Aufgabenanalyse, und wurde mithilfe einer Fallstudie basierend auf Sensordaten aus dem Energiebereich validiert. Zusätzlich wurden die Ergebnisse durch Expertenrückmeldungen bestätigt.

# Abstract

Ensuring an appropriate data quality is a critical topic when analyzing the ever increasing amounts of data collected and generated in today's world. Depending on the given task, even sophisticated analysis methods may cause misleading results due to an insufficient quality of the data set at hand. In this case, automated plausibility checks based on defined rules are frequently used to detect data problems such as missing data or anomalies.

However, defining such rules and using their results for an efficient data quality assessment is a challenging topic. Visualization is powerful to reveal unexpected problems in the data, and can additionally be used to validate results of applied automated plausibility checks. Visual Analytics closes the gap between automated data analysis and visualization by providing means to guide the definition and optimization of plausibility checks in order to use them for a continuous detection and validation of problems detected in the data.

This diploma thesis provides a design study of a Visual Analytics approach, called Data Quality Overview, which provides a detailed, yet scalable summary of the results of defined plausibility checks, and includes means for validation and investigation of these results at various levels of detail. The approach is based on a detailed task analysis of data quality assessment, and is validated using a case study based on sensor data from the energy sector in addition to feedback collected from domain experts.

# Contents

# List of Figures

CHAPTER 1

# Introduction

We are living in a world where data is collected and generated at an incredible rate. As the progress in computing power and storage capacities continues (Moore's law [Moo65]), the amount of data to be dealt with increases on a daily basis. For example, worldwide, up to 210 billion emails, four billion SMS messages and 500 million tweets are sent each day [KKEM10, Twi]. An increasing amount of data is also generated in many areas of science and industry, examples are simulation results in engineering, physics or climate research. The problem is: the possibilities to collect and store data increase at a faster rate than the human ability to use that data for making decisions [KKEM10].

This problem, referred to as "information overload", bears the danger of getting lost in the data: extracted information may be irrelevant to the current task at hand, or it may be processed or presented in an inappropriate way. As a consequence, time and money are wasted, and scientific and industrial opportunities are lost [Kei02]. Therefore, effective methods are needed to make use of the enormous potential that lies in the sheer vast amount of unexplored data, and to extract the hidden knowledge and opportunities that rest inside of it.

Even though powerful methods and tools for automated data analysis are developed, all face the same problem: they only work reliably for well-defined and well-understood problems. This is where *Visual Analytics* comes into play. Instead of just visualizing the results from automated data analysis techniques, Visual Analytics recognizes the potential of integrating the user in the whole analytical process through effective and interactive visualization techniques, aiming to make the way of how the data and information is processed transparent to the user [KKEM10]. The key of Visual Analytics is to enable a discourse between the human and the computer, both using their respective, distinct capabilities for the most effective results. Visual Analytics allows for using human knowledge, intuition and decision-making – which can not be automated – in combination with automatic methods from statistics or mathematics. Thomas et al. describe Visual Analytics as the "science of analytical reasoning facilitated by interactive visual interfaces" [Tho05].

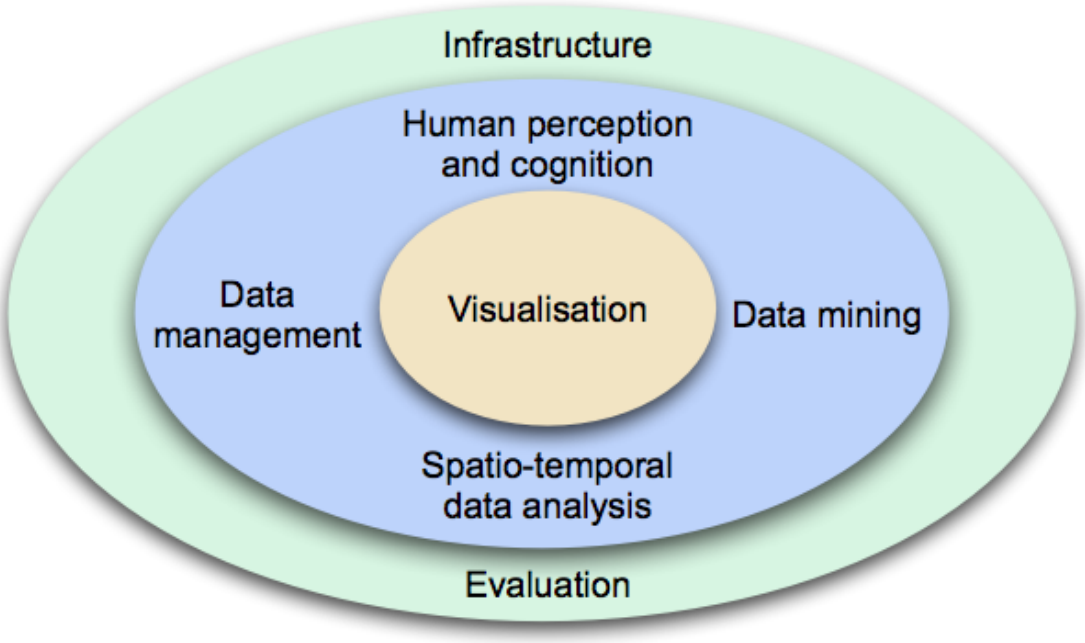


Figure 1.1: Integration of various research areas into visual analytics [KKEM10].

In many cases, analysis problems are ill specified, and data analysts sometimes do not even know which questions to ask in advance [Mun14]. Visual Analytics tools and techniques may provide insights in the vast amounts of data that are generated every day – to "detect the expected and discover the unexpected" [Tho05]. The field combines and integrates several research areas such as visualization, data mining, data management, data fusion, statistics and cognitive science and depends on the availability of the appropriate infrastructure and evaluation facilities [KKEM10] (Figure 1.1).

The Visual Analytics process tightly couples automated data analysis methods with interactive visualizations [KKEM10]. Figure 1.2 shows an overview of the process – the different stages (ovals) and their transactions (arrows). Before applying visual or automated data analysis methods, data may be transformed first, as it may originate from heterogeneous data sources or it may not be well formatted. After applying all needed transformations, the analyst can alternate between visual data exploration and automated data analysis methods. Models can interactively be built, evaluated and refined using interactive visualizations, which allow the analyst to interact with the automatic methods, by modifying parameters or by trying different analysis algorithms. This *feedback loop* is characteristic for the Visual Analytics process and leads to a continuous refinement and evaluation of the results [KKEM10].

However, even the most sophisticated analysis methods may fail due to an insufficient *data quality*. Problems like missing data, wrong data and duplicates may cause useless or misleading results or may even prevent the application of analytical methods in the

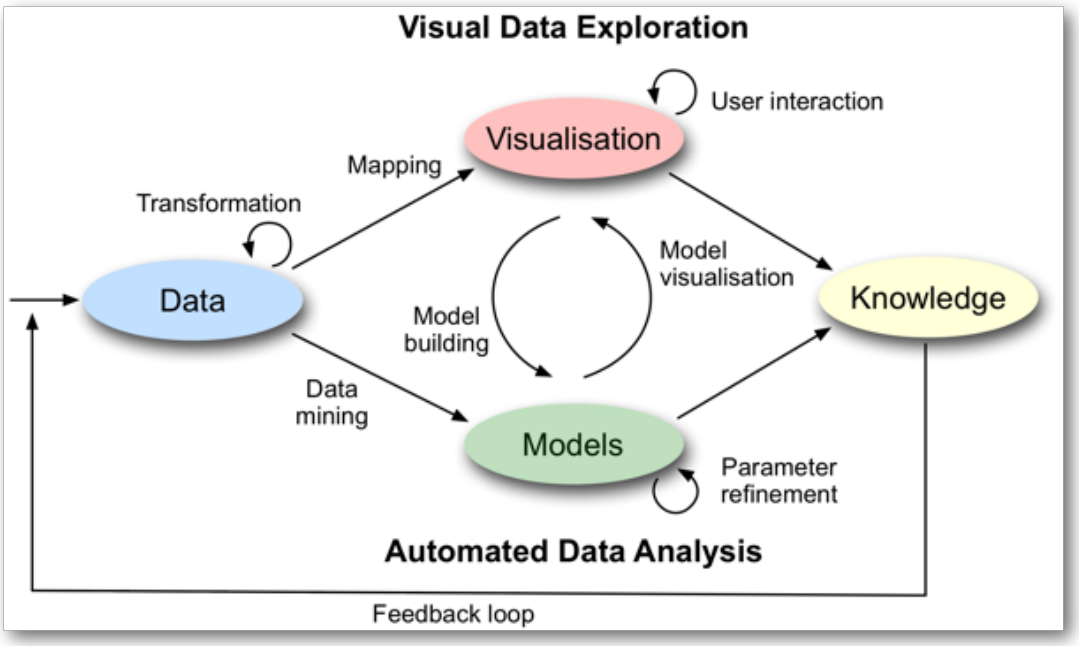


Figure 1.2: The Visual Analytics process. The figure displays the interaction between automated data analysis and visual data exploration by the user enabled through interactive visualization [KKEM10].

first place [GGAM12]. It is therefore essential to ensure a sufficient data quality for data-driven tasks like analysis, modeling or reporting in order to use the results to make correct decisions. The main problem, however, is that assessing the quality of a data set is a time-consuming task, even if domain experts are involved [KHP+11, Hel08]. For example, data quality may account for up to 80% of the time and cost in data warehousing projects [DJ03]. Ensuring an appropriate data quality is a recurring task in most operational settings and, as a result, domain experts regularly spend a large part of their time assessing the quality of the data instead of using this time to analyze the data for decision making or giving new insights. In the energy sector, for example, sensor measurements are continuously acquired and need to be processed at regular intervals, in order to be used for tasks such as forecasting. For these purposes, it is common practice to support an efficient detection of data quality problems by maintaining *plausibility checks*, and to apply them regularly to the incoming data.

The definition of automated plausibility checks has long been a topic of research in statistics and database management. There exist a variety of routines for detecting simple problems such as missing values, but also more sophisticated approaches to detect anomalies in a data set [Hel08, CBK09]. However, even if domain experts have a general idea about the variety of possible data problems that may arise in their area of expertise, actual data problems and their structure are not always evident when

analyzing a data set. In this regard, visualization is powerful in revealing unexpected data quality problems of different kind and structure, in addition to detecting expected issues in the data [Tho05, War12, Kei02, GAM+14]. Some of the most common insights gained by visualization are actually about data errors and outliers [KHP+11]. In this sense, visualizations can be used to find problems in the data and subsequently used as guidance to create plausibility checks for the detected issues. Furthermore, interactive visualizations may also allow for an iterative refinement of parameters of the identified plausibility checks, in order to reduce false positives and false negatives. Speaking in terms of the Visual Analytics process as seen in Figure 1.2, plausibility checks define models on the underlying data, which can be built and refined using suitable visualization approaches, in order to gain knowledge about the data in form of detected data quality issues.

Using the results of the plausibility checks for an efficient data quality assessment involves several challenges. As already mentioned, automated anomaly detection may not always be credible as results may contain false positives and/or false negatives. Another issue is the possibly large number of plausibility checks, data attributes and data records of a given data set. Assessing the data quality of a single time series, for example, seems to be a manageable task, but what if we need to check the plausibility of hundreds or thousands of time series? In addition, there may exist thousands of plausibility checks for a given data set. How can we manage to keep an overview of the overall data quality of the data set while simultaneously being able to examine the results of single plausibility checks for their credibility? Another scalability issue may arise with a high number of data records: How to cope with plausibility checks operating on millions of records, in order to validate data quality issues detected in single records as well as in a collection of related records?

So far, the visualization literature has not answered these questions sufficiently. Kandel et al. [KHP+11] describe dirty and ill-formatted data as an "elephant in the room" of visualization research, as most visualization research simply assumes that the data arrives in a clean state. They declare the design of visualizations for assessing data quality as an important research challenge in order to communicate the results of applied analytical techniques as well as to find possible reasons for detected quality issues in the data.

## 1.1 Contributions

The main contribution of this diploma thesis is a design study of the *Data Quality Overview*, a visualization approach that addresses the previously stated challenges. The goal of the Data Quality Overview is to support a routine quality assessment of multivariate data by visualizing the results of automated plausibility checks. Specifically, the visualization provides:

1. a scalable summary of plausibility check results,
2. drill-down features regarding plausibility checks, data attributes, and data records, and
3. an efficient quality-aware selection of data, e.g., for validating errors or defining suitable subsets for downstream processing.

Additional contributions of this thesis include:

- A task analysis of data quality assessment based on the collaboration with companies from different application sectors.
- An implementation illustrating aspects for integrating the Data Quality Overview into comprehensive systems for data quality management.
- A case study based on sensor data from the energy sector.
- Feedback from domain experts using a deployed version, and from a demonstration at Europe's premier energy fair.
- A reflection of the design process for stating lessons learned.
- An extraction of implementation details of the approach for understanding how the overview is integrated into the used framework.

## 1.2 Structure of the thesis

Chapter 2 discusses related work relevant to this thesis, and introduces the visual analytics framework Visplore, which was used for the implementation of the Data Quality Overview. Chapter 3 defines relevant tasks and extracts requirements for a Visual Analytics system to support data quality assessment. A data abstraction is given in chapter 4, explaining the underlying data model of the visualization, and also including a short explanation of the data set which is used for all examples throughout this thesis. The visualization approach called the Data Quality Overview is explained in Chapter 5, while the integration of the approach into the used framework Visplore is covered in Chapter 6. Chapter 7 provides a case study and feedback from domain experts using a deployed version of the Data Quality Overview. The design process of the visualization

approach is discussed in Chapter 8 by looking at intermediate iterations of the prototype and stating the pros and cons of each implementation. Chapter 9 summarizes some aspects of the system architecture and its implementation. A critical reflection is given in Chapter 10, along with a discussion of future work. Chapter 11 concludes this thesis.

CHAPTER 2

# Related Work

This chapter provides an analysis of state-of-the-art methods and other existing approaches regarding Visual Analytics of data quality assessment. First, an overview to the topics Visualization and Visual Analytics is given, including a short history of Visualization and basic concepts which are frequently used in Visual Analytics systems, focusing on the concepts used in the design of the Data Quality Overview. Then, the topic data quality itself is addressed, by stating existing characterizations of data quality problems and related work regarding the detection of anomalies in a data set. Existing approaches concerning visual data quality assessment are discussed, and an introduction to VISPLORE is given, a Visual Analytics system which is used as a framework for the implementation of this diploma thesis.

## 2.1 Visualization

> *"We acquire more information through vision than through all of the other senses combined."* [War12]

This quote by Colin Ware summarizes the role of *visualization* as an interface between the computer and the human: Visualizations combine the cognitive and perceptual abilities of the human visual system with the computational power of the computer. In the time before graphical displays existed, the term visualization referred to the "process of forming a mental image of something" [Dic15]. However, the understanding of the term has changed, as today it could be summarized as a "graphical representation of data or concepts" [War12]. According to Keim et al. [KMS+08], there exist three major goals of visualization:

1. *Presentation* - to efficiently and effectively communicate results of an analysis to a user. The choice of the appropriate visualization technique depends on the user as well as on the facts which are fixed a priori.

2. *Confirmatory analysis* - to visually examine hypotheses about the data, and the visualization either confirms or rejects these hypotheses.

3. *Exploratory analysis* - to search and analyze data to gain new insights and find potentially useful information about the data through visualization, without prior hypotheses.

The first visualizations - mainly used for *presentation* purposes - date back several thousand years, and range from cave paintings that were used as hunting guides, to plots of the movement of the stars or maps for navigation purposes [Fri06]. However, documentations of using visualization to not only display already known information did not appear until the 19th century. A well documented example for such a type of visualization can be seen in Figure 2.1: In the mid-19th century, many people died from a cholera epidemic in Great Britain, but the cause of the disease was long unknown. However, in 1855, Dr. John Snow created a map of a district of London displaying the locations of water pumps along with bars indicating the number of deaths in each household [Sno55]. The result was a visible cluster of deaths around a specific water pump, which confirmed the water-borne cause of the epidemic.

By the mid-19th century, many of the nowadays widely used visualization techniques such as bar charts, histograms and scatterplots were already invented in the context of statistical data analysis. However, all visualizations of that time were limited to printed graphics, which changed when the age of electronic computers began. Computers opened new possibilities, as they allowed fast image generation and therefore enabled to develop *interactive visualizations*.

> *"Interaction enables a discourse between the human brain and the data that, for example, allows to focus on interesting structures and to rapidly try many what-if scenarios in an ad-hoc fashion."* [Pir11]

Interactive visualization opens new possibilities for the user to focus on specific information by filtering data or zooming into an interesting part of the visualization, as well as to change its parameters at run-time. A new research area was born, which can roughly be subdivided into scientific and information visualization [KKEM10]. *Scientific visualization* deals with data which can directly be mapped to a 2D or 3D virtual environment. An example is the visualization of a computer tomography (CT) scan. On the other hand, *information visualization* mostly deals with data without an inherent spatial mapping. Examples are categorical or textual data from a survey. Visual metaphors are needed to display this kind of data.

## 2.2 Visual Analytics

> *"Overview first, zoom and filter, then details-on-demand."* [Shn96]

The "Visual Information Seeking Mantra" as identified by Ben Shneiderman [Shn96] is often followed when designing a Visual Analytics system. By providing an *overview*


Figure 2.1: Dr. John Snow's map of the distribution of the cholera deaths of a district of London, 1855. Each bar represents the number of deaths in a household, revealing a cluster of deaths around the water pump located in broad street [Sno55].

the user gets a picture of the entire data set, *zooming* and *filtering* allows the user to focus on specific interesting parts of the data, by zooming in on items of interest or by filtering out uninteresting items. Then, the user may select data records of interest to get *details-on-demand*. However, Visual Analytics is more than just visualizing a given data set as described by the Visual Information Seeking Mantra. Data sets may be complex and too large to be visualized in a straightforward way. As the amount of generated and collected data continuously increases [KKEM10], combining and integrating data analysis techniques with visualization is becoming more and more important.
Keim et al. [KMS+08] therefore extend the Information Seeking Mantra, and identified the "Visual Analytics Mantra":

*"Analyze first, show important, zoom, filter and analyze further, details on demand."* [KMS+08]

In order to avoid unimportant information to conceal important parts of the data, large data sets may need to be analyzed first before being able to show an overview of the important parts of the data. For example, in the context of this diploma thesis, the given data set is first analyzed using automated plausibility checks (see Section 4.2), and the important information in form of detected indications is visualized (see Chapter 5). Based on this initial visualization, further steps such as zooming, filtering or additional analysis may follow. Details on demand enable to examine the visualized data in more detail, in order to extract insights sought-after, or to identify further analysis steps or interesting parts to zoom or filter. For example, the Data Quality Overview provides details on demand by selecting detected data quality problems and visualizing the temporal context of the affected data records in a linked time series view (see Section 5.4).
The following sections include frequently used techniques and concepts of Visual Analytics. The focus in these sections primarily lies on techniques used for the implementation of the Data Quality Overview.

### 2.2.1 Data Aggregation

In order to cope with the ever-increasing size of data sets, *data aggregation* covers a set of transformations which can be used to reduce the amount of data to be displayed. This allows for a manageable overview of the data, while still preserving important information about the data set [EF10]. In the context of this diploma thesis, data aggregation allows the proposed visualization to scale regarding the number of data records in the data set, as well as regarding the number of data attributes and plausibility checks (see Sections 4.2 and 4.3).

**Hierarchical aggregation** Hierarchical aggregation techniques iteratively build a tree of *aggregate items*, each one consisting of one or more children, which are either original data records, representing the leaves of the tree, or aggregates of data records, representing the intermediate nodes [EF10]. The root node is an aggregate item representing the whole data set. In order to support hierarchical aggregation, visualization techniques have to provide visual representations for original data records as well as for the aggregate items.
Elmqvist and Fekete [EF10] state that the hierarchical aggregation "turns any visualization into a multi-scale structure that can be rendered at any desired level-of-detail". The user is provided a manageable overview of the data, while visual aggregates still indicate the data size, extents, or distribution. Hierarchically aggregated visualization techniques typically support the Visual Information Seeking Mantra stated before, as the user gets an overview of the data, basic interaction techniques allow the user to drill down in the hierarchy and retrieve details-on-demand. Figure 2.2 shows hierarchical aggregation techniques in the example of a 2D scatterplot visualization.

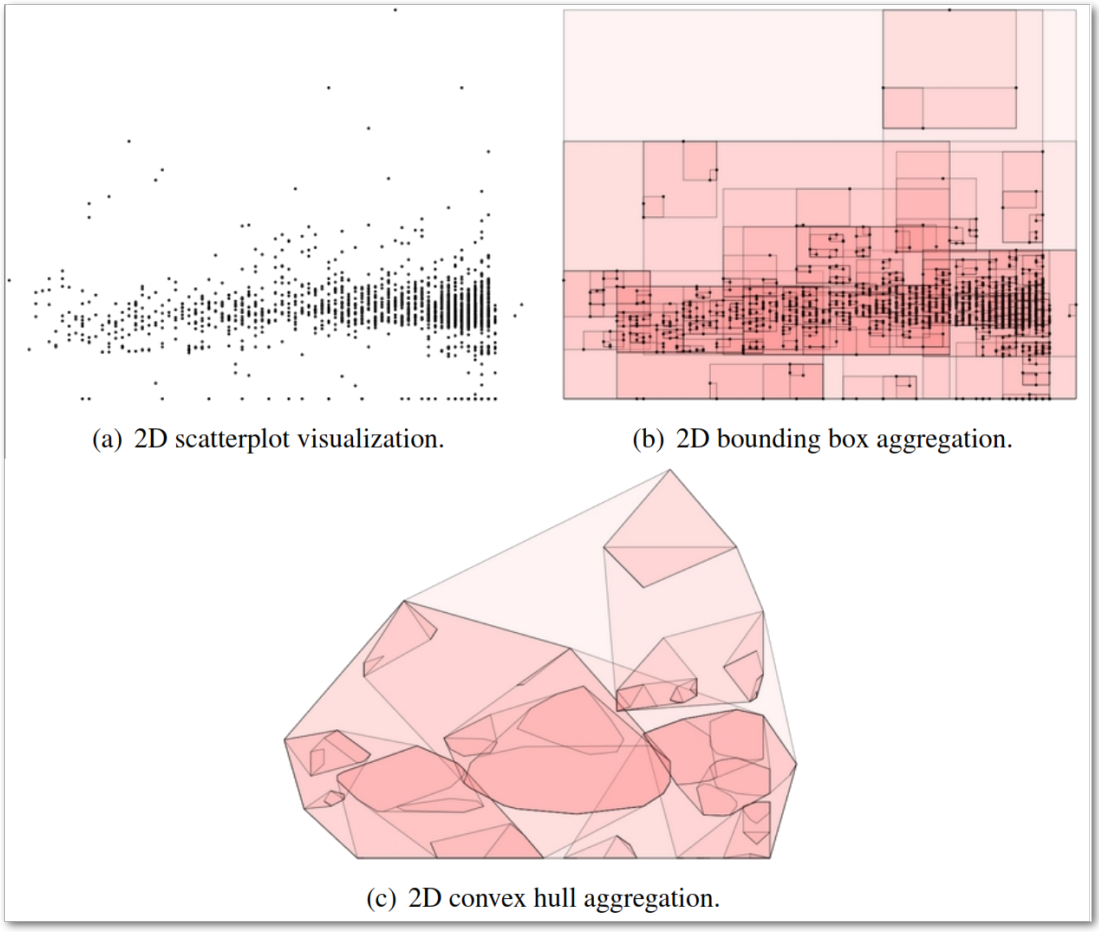


Figure 2.2: An example of hierarchical visual aggregation of a 2D scatterplot visualization [EF10].

**Binning** Binning is a process where a continuous data space is divided into a set of bins defined by intervals in the underlying data space. The data records are then categorized by the created bins, and for each one the number of data records is determined. When visualizing a data set with a large number of data records, binning is useful to reduce the visual clutter, while preserving the distribution of the data. For example, most implementations of histograms use univariate binning with equally sized bins in order to estimate the density of a given data dimension [Sil86]. The height of each bar in the histogram represents the number of data records that fall into each bin.
Depending on the data type and use case, different binning types may be appropriate. Hao et al. [HDS+10] for example use a variable (non-uniform) binning approach to reduce visual clutter of overlapping points in scatter plots. In their approach, each bin is represented by a square area in the scatter plot, and the areas are scaled, so that

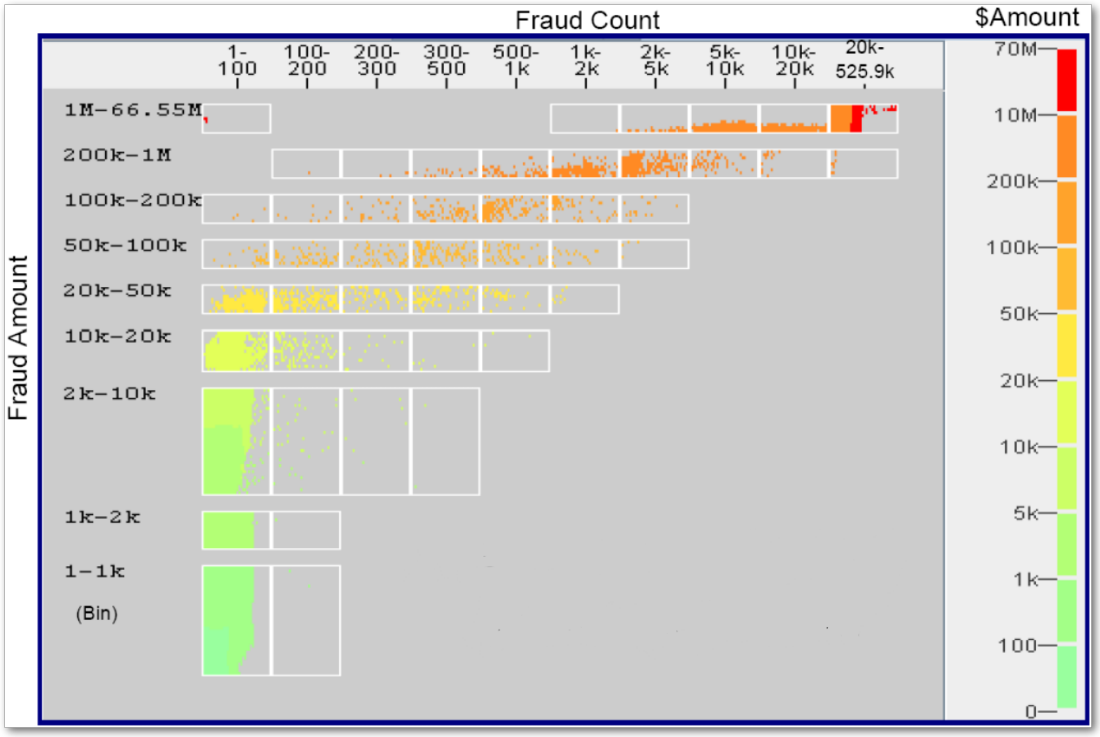


Figure 2.3: Variable Binned Scatter Plot as proposed by Hao et al. [HDS+10]. Each bin (visible as square area in the visualization) is scaled to show each contained data point without overlap [HDS+10].

each data point of the plot is shown without overlap. Figure 2.3 shows the approach visualizing credit card fraud data.

### 2.2.2 Filtering

By applying a *filter* to a data set, the set of data items to be visualized is reduced, based on specified conditions. Filtering irrelevant parts of the data may reduce the computational and visual complexity of a visualization [Pir11].
Although filtering can be applied statically at the beginning of the visualization pipeline, for example when importing a data set, interactive filtering approaches are more interesting in the context of Visual Analytics, and are considered as a basic type of interaction [Shn96, Kei02]. In the case of filtering categorical data attributes, interaction concepts range from check boxes to dragging categories onto a dedicated panel [AS94, STH02]. In order to control the displayed range for continuous data, range sliders are a frequently used element [TSDS96, ARP14]. A filter may be defined globally for the entire visualization of multiple views, or only for a part of the visualization.

### 2.2.3 Multiple Coordinated Views

As a result of the complexity and diversity of available data, there exist various visualizations that are great for a specific task, but inappropriate for others. Combining views has been established as a suitable method that allows to explore the data by viewing it through different representations and simultaneously enabling an interaction between them [Rob07]. Operations in the views are automatically coordinated, making it possible to perceive new and insightful relationships that lie in the data.

Designing effective systems supporting multiple coordinated views is challenging, as often a sophisticated coordination mechanism and layout are needed. Baldonado et al. [WBWK00] define guidelines for when multiple views should be considered and how they should be used. They identified a set of design rules to help designers and users whether or not multiple coordinated views are appropriate:

- **Rule of Diversity**
  Multiple coordinated views can be used when "there is a diversity of attributes, models, user profiles, levels of abstraction, or genres". Using a single view for many different needs may not be optimal for any of those needs, resulting in a "least-common-denominator" view. Diversity may require to simultaneously display a multitude of diverse data, creating a cognitive overhead for the user.

- **Rule of Complementarity**
  Visual comparison is easier to accomplish than memory-based comparison. Therefore, multiple views support possibilities to improve the understanding of complex relationships in a data set, to bring out correlations and/or disparities. Using a single view, the user has to memorize aspects of the visualization, before comparing it to another view on the data by switching between the provided views, which can be cognitively demanding.

- **Rule of Decomposition**
  A partitioning of a complex data set and displaying manageable parts of the data help the user to reduce the amount of data they need to consider at a time ("divide and conquer"). Multiple coordinated views can display multiple dimensions of the data, providing insight into the interaction between those dimensions.

- **Rule of Parsimony**
  Multiple views may introduce additional system complexity and computational costs, and take more display space. A single view provides a more stable context for the analysis, simultaneously requiring less learning costs for the user. Therefore, multiple views should be used minimally.

The interaction with and between the views plays a key role for systems supporting multiple coordinated views. Interaction techniques can either be *indirect*, when the interaction happens outside of the visualization (e.g., defining a global filter), or *direct*, when the interaction is performed within the visualization, such as *brushing* [Rob07].

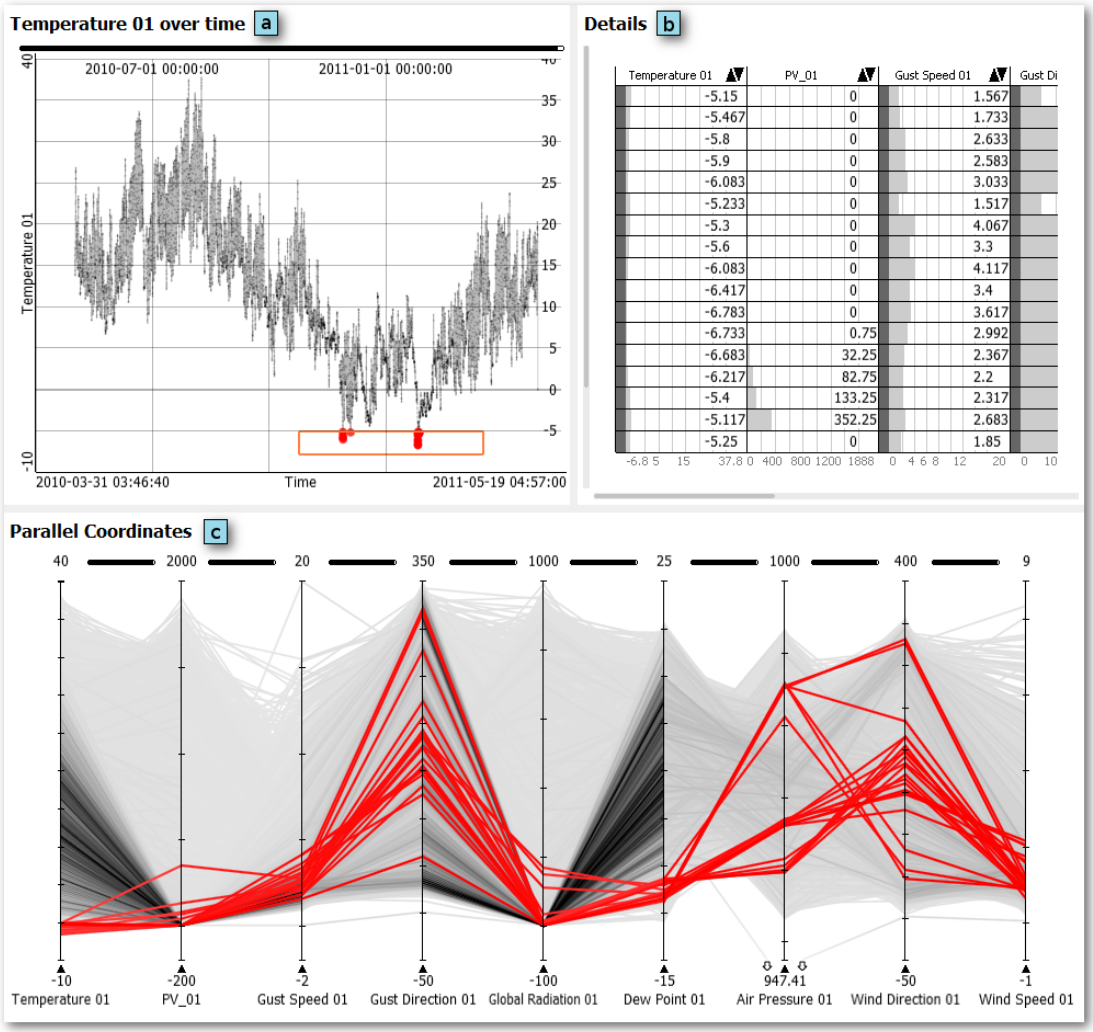


Figure 2.4: An example of multiple coordinated views in VISPLORE (see Section 2.5), including a time-series visualization (a), a data table (b) and a parallel coodinates view (c) [ID91]. The axis-aligned region brush as seen in the time-series visualization selects the underlying data items, whose individual data values are shown in the connected table. Additionally, the selected records are also highlighted in the linked parallel coordinates view.

Brushing is the most common type of direct interaction technique, where different degrees of interest are defined in the data [Pir11]. Brushing techniques range from simple approaches such as selecting individual data items by clicking on them, to selecting axis-aligned regions in scatterplots [BC87] or angular brushes in parallel coordinates [HLD02]. Figure 2.4 shows brushing in multiple coordinated views in an example of VISPLORE (see Section 2.5 for more details to VISPLORE).

### 2.2.4 Overview and Detail

When displaying all records of a data set in one image, it is typically not possible to visualize all details of the data simultaneously. Although traditional interface mechanisms such as paging, scrolling or panning allow for a navigation in the displayed data set, using these mechanisms introduces a discontinuity between the information which is displayed at different times and places [CKB09]. The user is burdened with mentally assimilating the overall structure of the data set, based on his navigation through parts of the data. In order to solve these problems, there exist several approaches to combine an overview visualization of a data set with a more detailed visualization of a subset of the data. This allows the user to examine interesting parts of the data in more detail without losing the overview of the data set as context information.
Existing approaches can be classified as follows [CKB09, KHG03]:

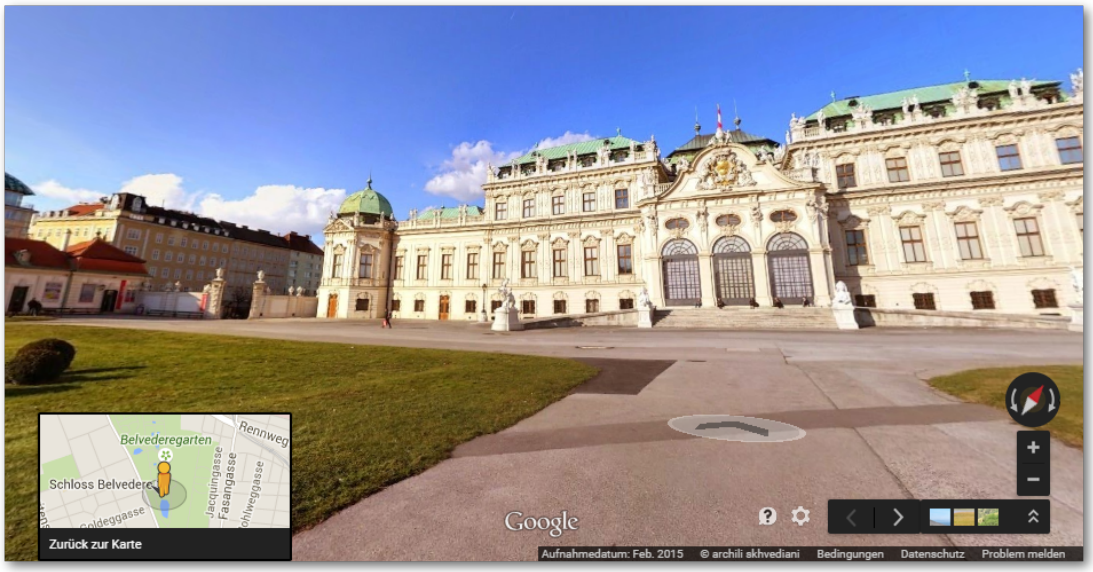


Figure 2.5: An example of an overview + detail visualization from Google Street View. The image shows the Upper Belvedere palace in Vienna (detail) as well as a map of the surroundings in the lower left corner (overview) [Goo].

- **Overview + detail**
  The visualization simultaneously displays the overview and the detail information, spatially separated into two views. The user can interact with both views separately, however, the interaction in one view often immediately also affects the other one. Beard and Walker [BI90], for example, describe techniques to navigate in large two-dimensional spaces using *map windows* - miniature visualizations of the entire space - which are used as an overview and navigation aid for a linked detail view displaying only a part of the data space. Figure 2.5 shows such an example from Google Street View [Goo], where the picture of a place (detail) is shown together with the map of the location (overview) - representing the map window as described by Beard and Walker [BI90].

- **Zooming**
  The zooming interaction technique allows for a temporal separation between depicting the overview and a detail visualization. The level of detail can be adjusted by magnifying (zoom in to focus) or by demagnifying (zoom out for context) a data set, instead of showing both views at the same time. Design issues arising when designing zoomable interfaces include the coupling between panning and zooming [FB95] or the transition between zoom states such as choosing between continuous and discrete zoom levels or creating animated transitions [vWN04, BB99].

- **Distortion-oriented focus + context**
  The image of a view is distorted geometrically, so that both overview and detail are integrated into a single visualization. An example is shown in figure 2.6, which displays a geometrically distorted temperature time-series in VISPLORE. The distortion is achieved by zooming into the interesting area using the slider visible in the bottom area [ARP14]. Possible problems when distorting an image include misinterpretation of the underlying data and difficulties in selecting parts of the distorted view [CKB09].

- **In-place focus + context**
  In-place focus + context techniques, instead of distorting the image, employ visual cues to discriminate focus from context. A subset of the data is highlighted by modifying visual attributes such as the color or the size of glyphs. Figure 2.7 shows an example where a depth-of-field effect is used to distinguish between objects in focus (sharp) and in context (blurred) [KMH01].

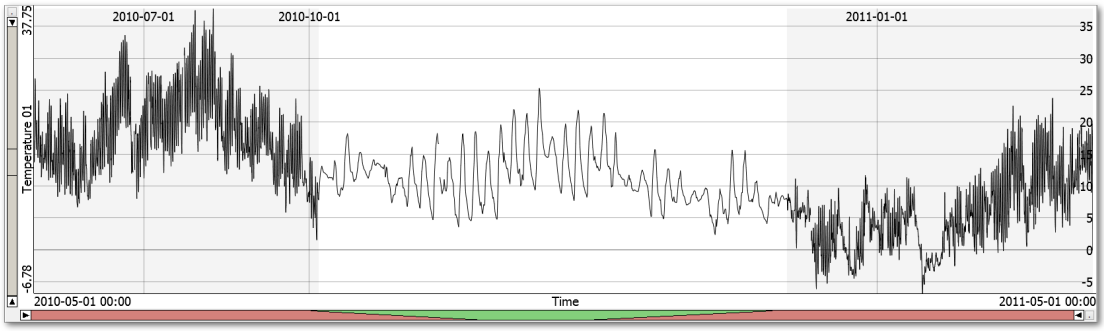


Figure 2.6: Distortion-oriented focus + context in an example from Visplore. Using the slider in the bottom part of the view, the user can zoom into interesting areas of the time-series, distorting the image to show the zoomed area in more detail, while still maintaining the context.

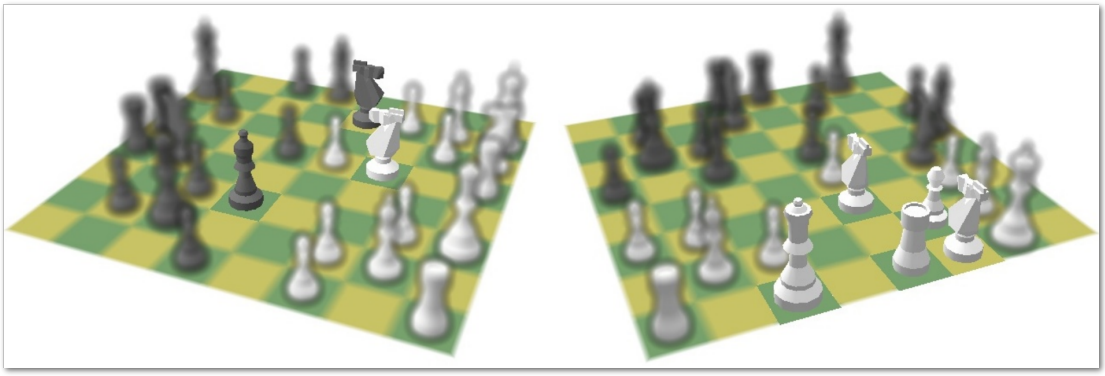

Figure 2.7: Semantic Depth of Field visualization of a chess tutoring system. Pieces in focus are depicted sharply, the pieces in the context are blurred [KMH01].

## 2.3 Data Quality

Data quality has long been the target of intensive research in multiple fields, for example by business analysts, solution architects, database experts, or statisticians [SYI11]. From the consumer's point of view, the term *fitness for use* is taken as a widely agreed definition of the term "data quality" [WS96].

### 2.3.1 Characterization

There are different approaches to characterize data quality. Multiple researchers focus on the dimensions of data quality, other work focuses on the classification of specific data quality problems. Pipino et al. [PLW02], as well as Batini et al. [BS16] identified dimensions of data quality such as accessibility, completeness, or consistency (among

others). *Accessibility* addresses the extent to which data is available, or easily and quickly retrievable, *completeness* refers to the amount of missing data as well as to the sufficient breadth and depth of the data needed for a specific task, while *consistency* refers to the extent to which data is represented in the same format.

Rahm and Do [RD00] focus on the classification of data problems by their sources. They distinguish between single-source and multi-source problems and further between schema- and instance-related problems (see Figure 2.8). Multi-source problems arise when multiple data sources need to be integrated: the sources typically are maintained independently from each other, resulting in different representations of the data as well as overlaps or contradictions. The problems are additionally aggravated as each of the sources may contain dirty data themselves. Schema-related problems refer to problems regarding the design of the schema itself such as wrong integrity constraints, or, in the case of multiple data sources, schema design differences between the different sources. Instance-related problems refer to errors that cannot be prevented at the schema level, such as misspellings.

Kim et al. [KCH+03] provide a comprehensive classification of dirty data, beginning with a simple subdivision of dirty data into missing and not-missing, and continuing by further refining these two categories into finally 33 primitive dirty data types, adopting the standard "successive hierarchical refinement" approach. The fan-out factor is kept at a small level to make it intuitively obvious that no other meaningful child-nodes are possible at any given node.

Müller and Freytag [MF05] roughly classify data anomalies into syntactical, semantic, and coverage anomalies. *Syntactical anomalies* describe anomalies concerning the format and values of the representation of the data records. *Semantic anomalies* characterize

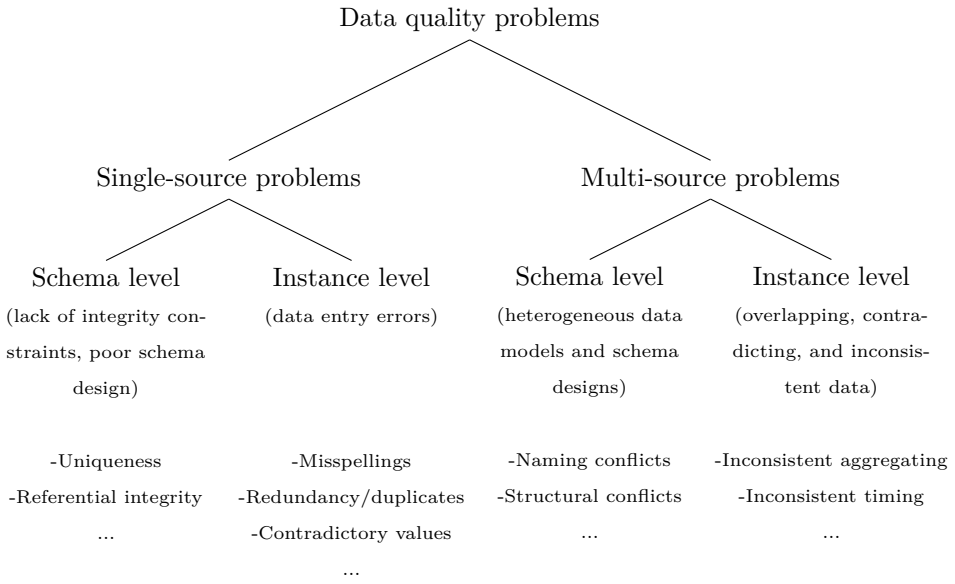


Figure 2.8: Classification of data problems by Rahm and Do [RD00]

| | | Description<br>*Example* | non-rastered | | | rastered | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | Point in time | Start/End of interval | Duration | Point in time | Start/End of interval | Duration | Time-dependent values |
| **Single source** | | | | | | | | | |
| **Missing data** | Missing value | Missing time/interval and/or missing value<br>*(Date: NULL, items-sold: 20)* | ● | ● | ● | ● | ● | ● | ● |
| | | Dummy entry<br>*(Date: 1970-01-01); (duration: -999)* | ● | ● | ● | ● | ● | ● | |
| | Missing tuple | Missing time/interval + values<br>*(The whole tuple is missing)* | | | | ● | ● | ● | ● |
| **Duplicates** | Unique value violation | Same time/interval (exact same time/interval though time/interval is defined as unique value)<br>*(Holidays: 2012-04-09; 2012-04-09)* | ● | ● | ● | ● | ● | ● | |
| | Exact duplicates | Same time/interval and same values<br>*(Date: 2012-03-29, items-sold: 20 is in table twice* | ● | ● | ● | ● | ● | ● | ● |
| | Inconsistent duplicates | Same real entity with different times/intervals or values *(patient: A, admission: 2012-03-29 8:00) vs. (patient: A, admission: 2012-03-29 8:30)* | ● | ● | ● | ● | ● | ● | ● |
| | | Same real entity of time/interval (values) with different granularities (rounding)<br>*(Time: 11:00 vs. 11:03); (Weight: 34,67 vs 35)* | ● | ● | ● | | | | ● |
| ⋮ | ⋮ | ⋮ | | | | | | | |
| **Multiple Sources** | | | | | | | | | |
| **Heterog. syntaxes** | Different data formats/ synonyms | Different date/duration formats<br>*(Date: YYYY-MM-DD) vs. (date: DD-MM-YYYY); (Date: 03-05 (March 5)) vs. (date: 03-05 (May 3))* | ● | ● | ● | ● | ● | ● | |
| | Different table structure | Time separated from date vs. date+time or start+duration in one column<br>*(Table A: start-date, start-time) vs. (table B: start-timestamp)* | ● | ● | ● | ● | ● | ● | |
| **Heterog. semantics** | Heterogeneity of scales (measure units / aggregation) | Different granularities; different interval length<br>*(Table A: whole hours only) vs. (table B: minutes)* | ● | ● | ● | ● | ● | ● | |
| | Information refer to different times/intervals | Different times/intervals<br>*(Table A: current sales as of yesterday) vs. (table B: sales as of last week)* | ● | ● | ● | ● | ● | ● | |
| ⋮ | ⋮ | ⋮ | | | | | | | |

Figure 2.9: Subset of the taxonomy of dirty time-oriented data by Gschwandtner et al. [GGAM12]

inconsistencies of the data set itself or when multiple data sets are merged. Examples are integrity-constraint violations or duplicates. *Coverage anomalies* describe missing entities or entity properties.

Gschwandtner et al. [GGAM12] summarize and compare a set of existing taxonomies and address the special characteristics of dirty *time-oriented data* in an own taxonomy. The concept of distinguishing data quality problems by single-source and multi-sources was adapted from Rahm and Do [RD00], some categories were re-arranged, refined and extended. The considered data types were categorized into non-rastered and rastered data types, with each category containing the temporal units "point in time" and "interval" (defined by a start- and end-point or a duration). With respect to the defined categories, it is outlined which data quality problems arise for which data type. Figure 2.9 shows a subset of the taxonomy defined by Gschwandtner et al. [GGAM12].

### 2.3.2 Anomaly Detection

Anomalies in a data set describe patterns which do not conform to expected behaviour, such as values lying outside of an expected range or sudden unplausible value changes. The automated detection of *anomalies* has long been a key topic in statistics and database management [Hel08, CBK09] and finds use in many application domains such as credit card fraud detection or intrusion detection for cyber-security. There exist various anomaly detection techniques that have been specifically developed for certain applications, as well as other techniques which are more generic.
Chandola et al. [CBK09] provide a survey of anomaly detection techniques, including a discussion of advantages, disadvantages and the computational complexity. Additionally, different aspects of anomaly detection problems are identified, such as the nature of input data, the type of anomaly, data labels and the output of the anomaly detection. Chandola et al. classify anomalies into three different types:

- **Point Anomalies** refer to data values which can individually be considered as anomalous with respect to the rest of the data. An example is shown in Figure 2.10, which depicts two normal regions $N_1$ and $N_2$, where most of observations lie. In this case, the points $o_1$ and $o_2$ as well as the points of $O_3$ depict point anomalies, as all those points lie outside of the normal regions.

- **Contextual anomalies** refer to data instances which are anomalous only in a specific context. Figure 2.11 shows an example of a temperature time series, where the temperature at time $t_1$ is considered as normal for winter time, whereas the context of the temperature at time $t_2$ indicates an anomaly.

- **Collective anomalies** refer to a collection of related data records that are anomalous regarding the entire data set. An example can be seen in Figure 2.12, which shows a time series of the production of a photovoltaic power plant. In this case, for a longer time period, all production values seem to have an offset to where the values would lie normally.

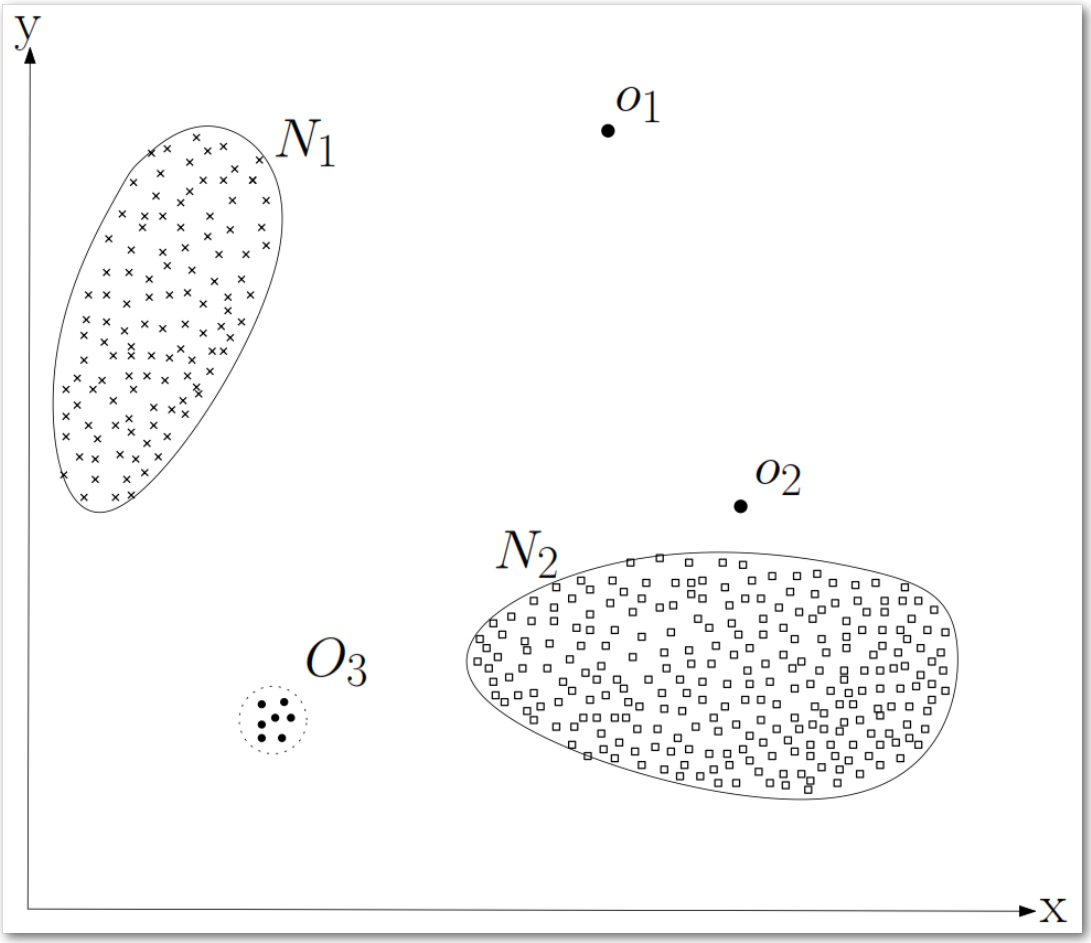


Figure 2.10: Point anomalies in an example of a two-dimensional data set. Most observations lie in the two regions $N_1$ and $N_2$. The points $o_1$, $o_2$ and the points in $O_3$ lie outside of these normal regions [CBK09].

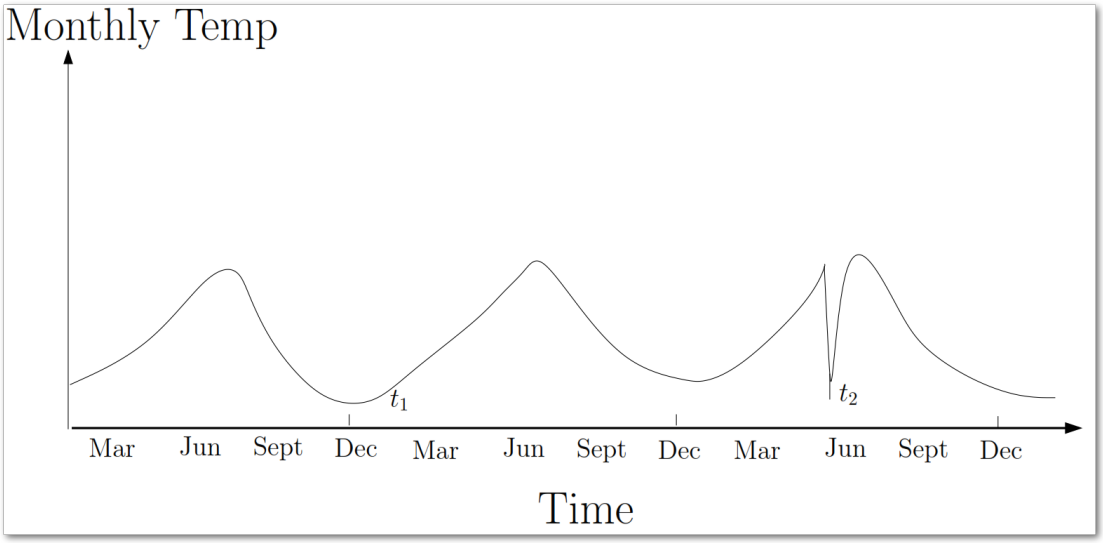


Figure 2.11: An example of a contextual anomaly in a temperature time series. The value at $t_2$ is considered as an anomaly, as the temperature at this point is not plausible with regard to its context (summer). The same value at $t_1$, however, is not considered as anomaly, as it occurs in a different context (winter) [CBK09].

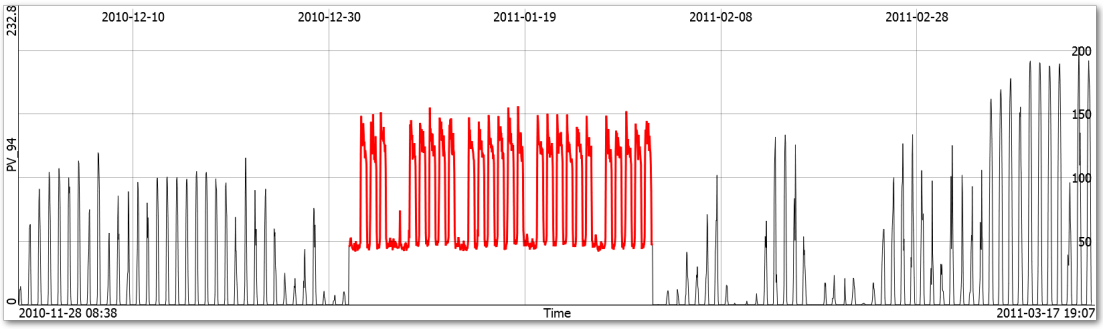


Figure 2.12: Collective anomalies shown in a time-series of the production of a photovoltaic power plant.

Another aspect is the output of the techniques for anomaly detection. Typically, the output is either defined by *scores*, where an anomaly score is assigned to each instance in the data, or binary *labels*. Scores state to which degree the entry is considered an anomaly, while binary labels do a classification of records into normal or anomalous. Anomaly detection techniques based on scores can produce binary labels using a domain-specific threshold to select only the most relevant anomalies [CBK09].

Pre-defined *labels* associated with the data records, which mark them as normal or anomalous, provide opportunities for *supervised* or *semi-supervised* anomaly detection instead of *unsupervised* anomaly detection [CBK09]. Supervised anomaly detection techniques use an available training data set with labeled instances for both the normal and the anomaly class. Based on this training set a model is built, and new data instances then are compared against the model to classify them. Semi-supervised anomaly detection techniques on the contrary have a training data set where only normal data records are labeled. Unsupervised techniques do not require training data, and are therefore easier to apply. These techniques simply assume that normal data records are far more frequent than anomalous ones. However, if this assumption is not true, unsupervised techniques suffer from a high number of false positives and false negatives.

## 2.4 Visual Data Quality Assessment

### 2.4.1 Data Wrangling

Kandel et al. [KHP+11] describe visual data quality assessment as an opportunity in the broader context of *data wrangling*, which they define as a "process of iterative data exploration and transformation that enables analysis" or simpler, "the process of making data useful". The output of this process should not only comprise a usable data set for a specific application context, but additionally include the set of transformations which led to the cleaned data set. An important requirement for a system to support data wrangling therefore is to provide the given set of transformations in an interactive transformation history, allowing users to insert additional transformations at every moment in the history, as well as to edit or remove existing transformations. In order for such a system to be successful, the transformations and their history must be represented in a comprehensible way so that they are understandable for auditors. [KHP+11]

Figure 2.13 shows the iterative process of wrangling and analysis, as identified by Kandel et al., including a typical trajectory of the process. A tool for data wrangling ideally should combine wrangling and analysis (indicated by the light yellow square in the figure), as data wrangling may take place at all the stages of the analysis: before loading a data set into visualization and analysis tools, during analysis itself, as well as when data is updated or additional data is added. As the analysis of the data may reveal unexpected data quality problems, users may have to iteratively switch between data wrangling and analysis. However, wrangling tools tend to be separated from the visual analysis tools, rending the iterative process cumbersome, as tools have to be switched frequently.

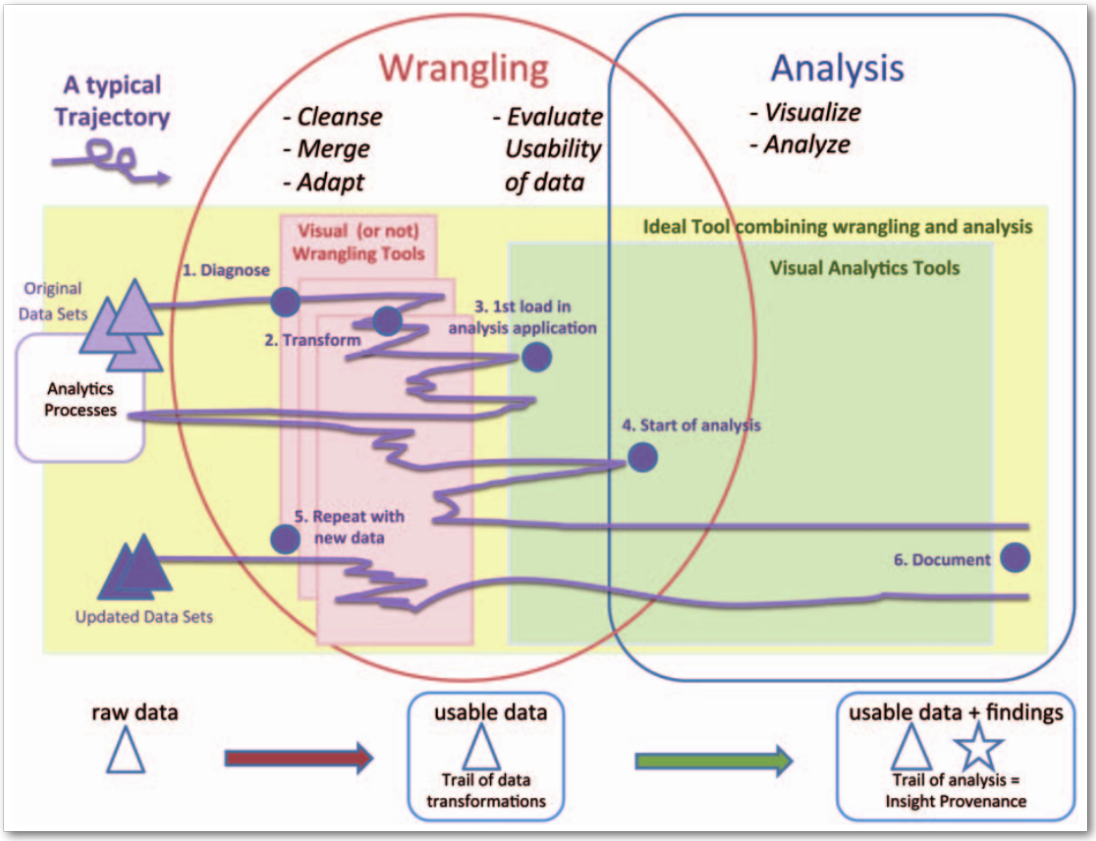


Figure 2.13: The iterative process of wrangling and analysis [KHP+11].

There exist several visualization tools that address aspects of data wrangling such as data cleansing or data transformation. For example, Kandel et al. provide a system called "Wrangler" [KPHH11], which implements the previously stated requirements, and allows users to interactively specify data transformations by combining direct manipulation of visualized data with an automatic suggestion of relevant transformations. Wrangler includes an interactive history viewer to support review, refinement, and annotation of the applied transformations. However, Wrangler restricts the visualization of the data to a simple data table, without providing further visualizations of the data, which limits the possibilities to explore the data for potential data quality problems and to visually verify found problems. Figure 2.14 shows the interface of Wrangler, the left panel giving a history of the applied transformations, the right panel displays the data records in an interactive data table.

Raman and Hellerstein introduced "Potter's Wheel" [RH01], an interactive data cleansing system. It supports defining transformations to clean the data graphically, and their effects are immediately shown on the visible records. If the effects of a transformation are not as desired, it can be undone. The detection of data quality problems is done in

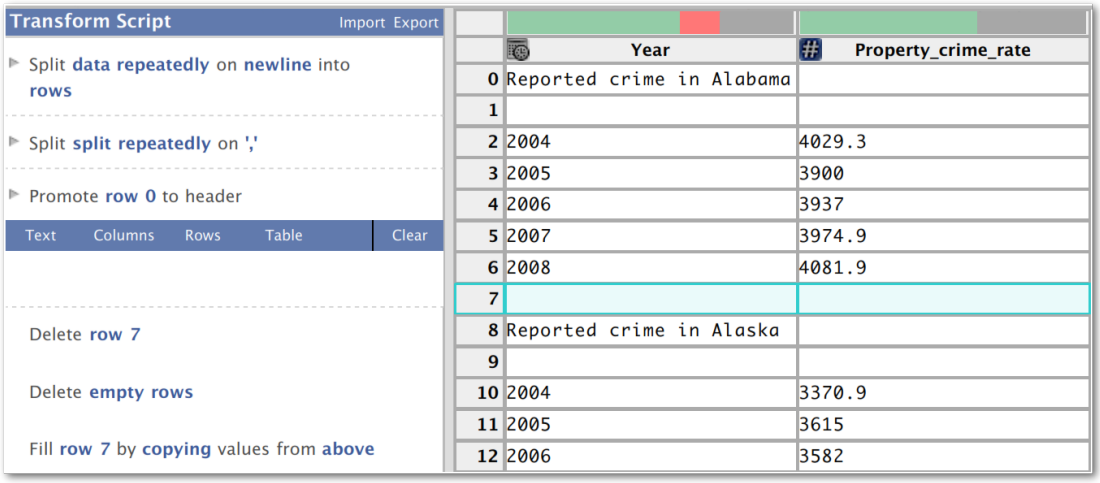


Figure 2.14: The Wrangler user interface [KPHH11].

the background, on the latest version of the data modified by the current transformations.

Additionally, publicly and commercially available tools such as OpenRefine [Ope] and Trifacta [Tri] support the user in defining data transformations intuitively and in creating scripts for a repeated application – similar to "Wrangler" and "Potter's Wheel". The goal of all these tools is to support data quality management on a regular basis, which is similar to ours. However, we address a different stage of the workflow, i.e., the inspection of automated check results, while modifying data is outside of the scope of this diploma thesis.

### 2.4.2 Visualization of Dirty Data

Once data problems are identified, and dirty data records are found, it is not always clear how or whether the records should be modified. In some cases, it could be useful to proceed with the analysis in the presence of inconsistencies such as missing data or outliers [KHP+11], and to include such data in the visualization. The challenge is to encode dirty data records in the visualization without distracting from an analysis of the rest of the data.

There exist several approaches to explicitly convey data quality aspects during visual analysis. For example, Ward et al. [WXYR11] describe how data quality aspects can be measured, displayed and utilized in visualizations, subsequently helping to raise the awareness of analysts of the accuracy and completeness of the visualized information. In the approach of Sulo et al. [SEG05] data quality problems such as duplicates or missing values are highlighted in a tabular reduced visualization on a data-record level. However, the scalability of the proposed visualization is limited, due to the lack of aggregation.

The visualization of missing data in particular is repeatedly mentioned as important to consider in visual analytics systems [WV12, KHP+11] and there exist several studies

on displaying and dealing with missing data [EPD05, TAF12, HHSU97, FG14]. On the other hand, some visualization techniques try to convey uncertainty information, arising from possible measurement errors, missing data, and sampling [KHP+11]. These techniques encode the uncertainty in the data values using, e.g., transparency, blur, error bars, or error ellipses [GS06, CCM09, OM02, PWL97].

### 2.4.3 Visual Data Profiling

Exploratory visualization is considered as suitable in order to get an intuition of the data and possibly arising data quality problems [War12, Kei02]. There exist several well-known systems such as Tableau [Tab], Spotfire [TIB], GGobi [SLBC03] and many others, which provide means to visually explore the properties of a given data set and subsequently be used for data quality profiling.

In order to assess the quality of a data set, several approaches focus on specific domains. As an example, there exists a series of visualization systems to support anomaly detection within sensor networks. In the approach of Shi et al. [SLH+11], topological, correlational, and dimensional views of anomalies enable expert users to facilitate sensor failure diagnosis. Steiger et al. [SBM+14] try to identify anomalous patterns in time series data in sensor networks using dimension reduction. More related to our goals, Janetzko et al. [JSMK14] apply anomaly detection techniques to power consumption data and visualize the results in a pixel-based time series and anomaly visualization. While their pixel-based approach provides many details, it is specific to the data and lacks summary values for non-experts.

In contrast, there exist several commercial systems which provide coarse summaries of data quality indicators also for non-expert users, without focusing on any specific domain. The Talend Open Studio for Data Quality [Tal] highlights data quality problems in corresponding tables, using the results of data quality indicators, which are either pre-defined or user-defined. IBM Watson Analytics [IBM] assigns an overall data quality score to a data set and to individual data attributes. The main drawback of both Talend and IBM Watson is that their visualization is limited to mostly static diagrams, which reduces the possibilities for an interactive identification of plausibility checks or an exploration of the respective results.

Gschwandtner et al. support the cleansing of time-oriented data with an interactive visual analytics system called "TimeCleanser" [GAM+14]. Issues in the data are detected by predefined as well as user-defined checks, and corrected using a wizard-like guide through different cleansing steps. Figure 2.15 shows one of the cleaning steps, in this case correcting the syntax of of data entries. A bar chart (Figure 2.15a) shows the types of identified syntax errors and the number of detected errors. Corrections of date, number or text entries can be selected and executed as seen in Figure 2.15b. A table showing erroneous values (Figure 2.15c) is provided to manually correct the raw values of the data entries. A wizard-like navigation through the different cleansing steps is provided by the interface elements as seen in Figure 2.15d.

We regard Profiler [KPP+12] as most related system to this diploma thesis. Problematic data is automatically detected using data mining methods for a visual summary

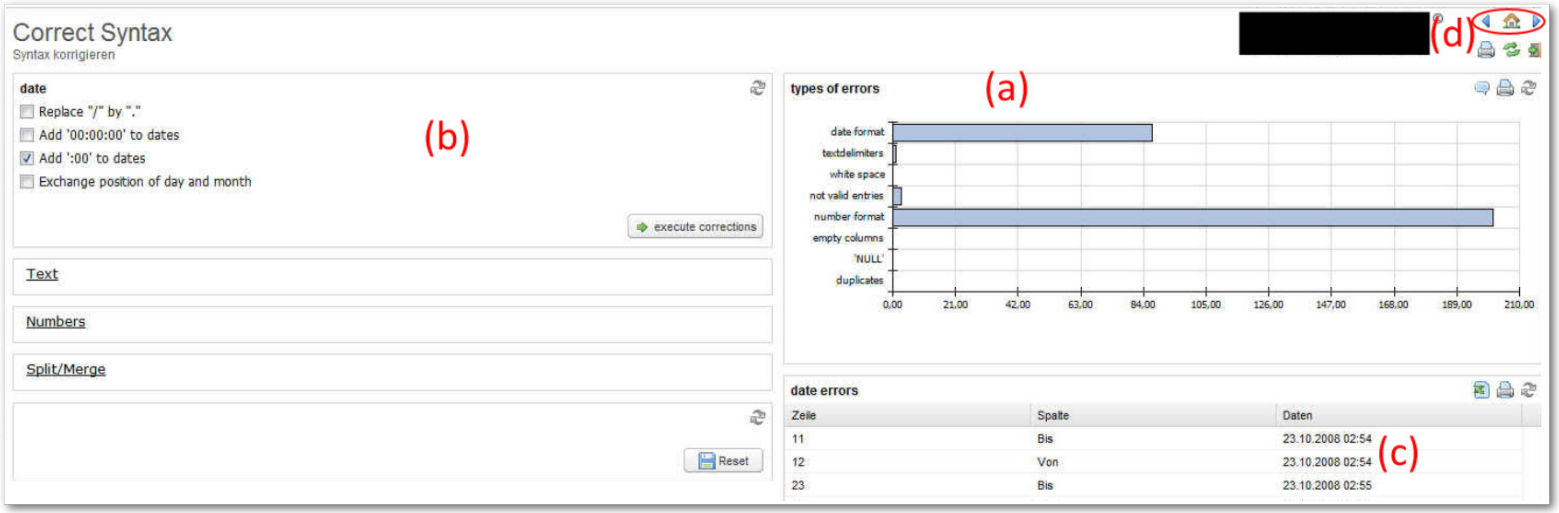


Figure 2.15: The TimeCleanser user interface [GAM+14].

of the overall data quality. Based on an analysis of mutual information between table columns and the detected anomalies, sets of binned views are suggested, indicating possible sources of found problems. Figure 2.16 shows this system with an example of movie data. Depending on the selection of the anomaly in the anomaly browser, the system automatically recommends relevant visual summaries that may explain the selected anomaly found in the data set. In the example, possible causes of missing MPAA movie ratings are investigated. The first (primary) view of the linked summary visualizations always shows the set of columns that contain the anomaly, in this case a chart of the MPAA rating itself. All other views are recommended using a model based on mutual information, which "quantifies how much knowing the value of one variable reduces the uncertainty in predicting a second variable". A grey bar above the MPAA rating chart indicates the number of missing values of this column, and can be selected to highlight matching records in other recommended views. Selecting the missing values indicates a correlation with early release dates, which is shown in the second view.

Although Profiler offers numerous detection routines, it does not support an in-situ identification or optimization of user-defined plausibility checks. Moreover, our approach provides more flexibility for hierarchically structuring and aggregating check results than the anomaly browser of Profiler. Moreover, the linear layout of the Data Quality Overview enables a direct comparison of results and their distribution.

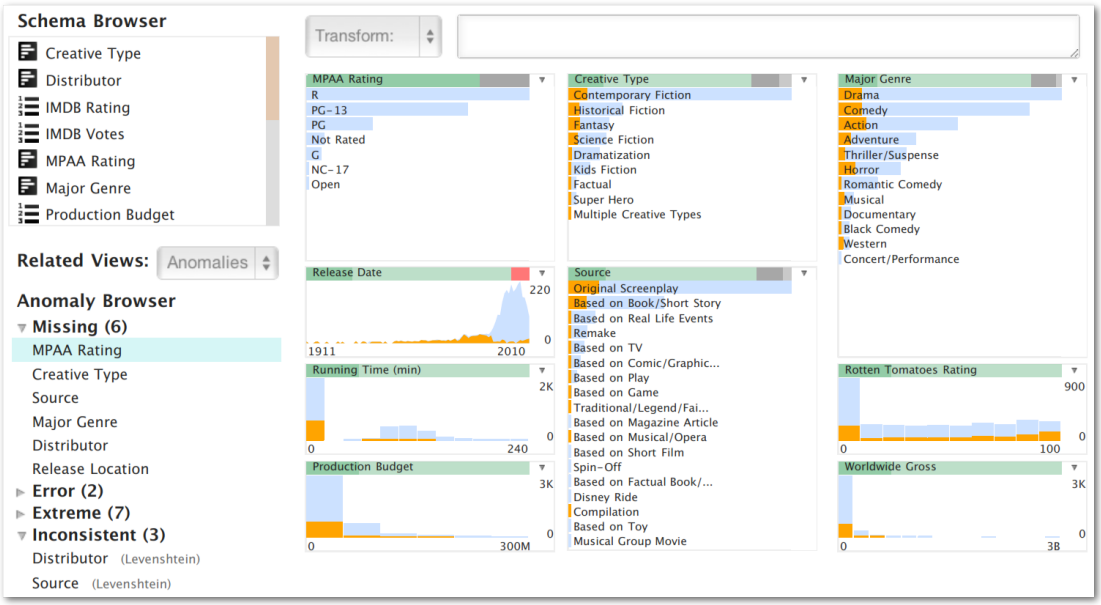


Figure 2.16: The Profiler user interface [KPP+12].

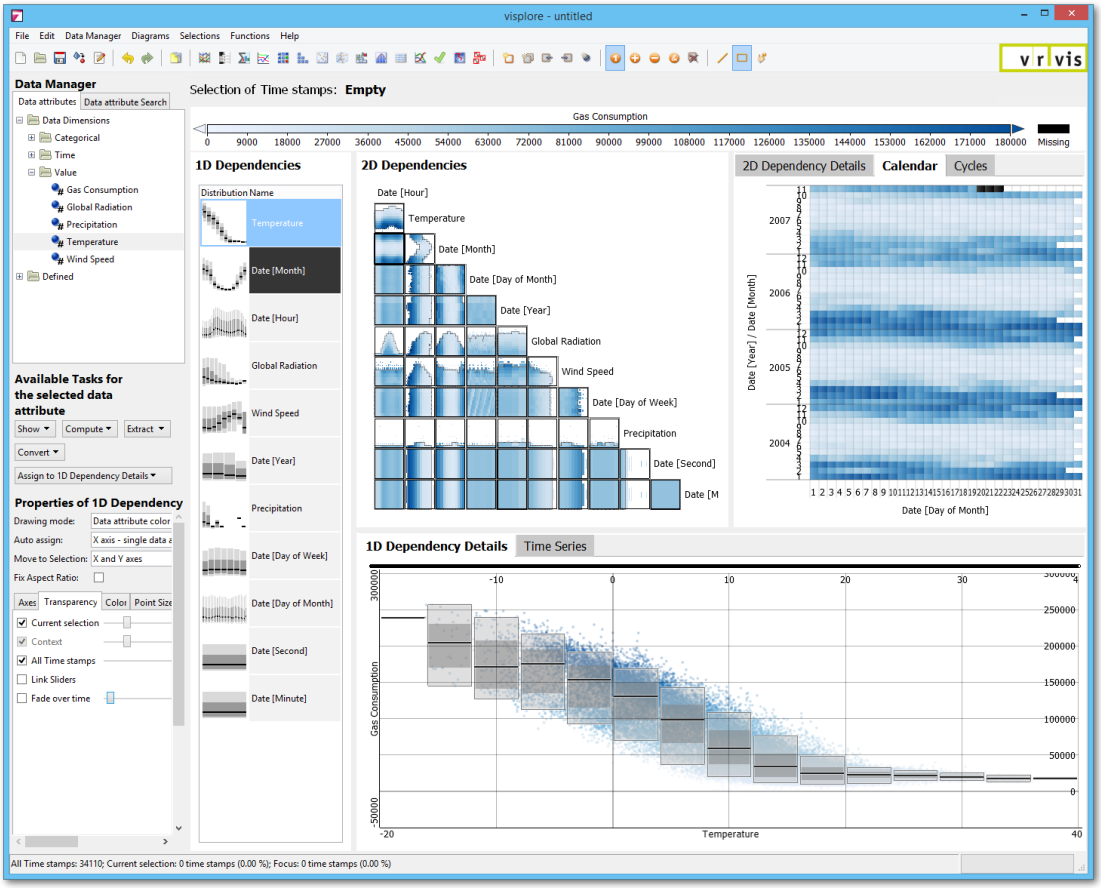


Figure 2.17: An example configuration of VISPLORE as used in the energy sector. The use case in this example is to identify dependencies of a data attribute ("Gas Consumption").

## 2.5 Visplore

The implementation of this diploma thesis was developed as a plugin for VISPLORE, which is an existing visual analytics software developed at the VRVis Research Center in Vienna[1]. VISPLORE focuses on the visual exploration of large and complex data in various application scenarios, and is used as a framework for basic research projects along with application-oriented research in cooperation with industrial partners. The software supports a variety of visualization techniques and incorporates basic methodologies such as multiple coordinated views (see Section 2.2.3). Figure 2.17 shows an example visualization setup.

Visualization techniques provided by VISPLORE range from simple techniques such as 2D scatterplots [GTC01] or histograms to more complex visualizations such as Rank by Feature View [SS05], which has been extended to a partition-based framework for building

[1]VRVis Zentrum für Virtual Reality und Visualisierung Forschungs-GmbH, Vienna, *www.vrvis.at*

and validating regression models [MP13]. The following list gives a short overview of the variety of available visualization techniques:

- **2D and 3D Scatterplot**
  For the visualization of two- and three-dimensional data, a scatterplot is a well-known technique, where each data entry is represented by a glyph, aligned by two respectively three orthogonal axes [GTC01, PKH04].

- **Histogram**
  A Histogram provides a quick overview of the distribution of a data attribute by applying univariate binning to the value range of the assigned attribute and displaying the number of contained entries for each bin (see Section 2.2.1).

- **Parallel Coordinates**
  Parallel Coordinates is a visualization technique frequently used for the visualization of multivariate data. The visualization consists of parallel lines representing the assigned data attributes, and a *polyline* is drawn for each entry in the data set [ID91].

- **Aggregate View**
  The Aggregate View allows for a partitioning of the data set into cells using categorical dimensions assigned by the user. The main feature of the view, also giving it its name, is the possibility to assign an aggregate function such as average, min, max or a simple count (but also more sophisticated functions) to one of the two axes, or to visual attributes such as size or colors. The aggregate is applied to each cell of the configuration, the overall visualization resulting in a bar chart, heatmap, line graph or scatterplot, depending on the given configuration.

- **TableLens View**
  The Table Lens is a technique introduced by Rao and Card [RC94] for the visualization of large data tables. It uses a distortion-oriented focus+context technique (see Section 2.2.4), where the zoomed area ("lens") displays the full information provided by the data entries in this area, whereas entries in the context are mapped to less space.

- **1D Rank By Feature View**
  This view provides a set of common statistical measures for assigned data attributes such as min, max, average or standard deviation (among others) [SS05, PBH08], as well as a histogram combined with a box plot, providing a comprehensive statistical overview of the attributes. The attributes can further be ranked by a selected statistical measure.

- **2D Rank By Feature View**
  The 2D Rank By Feature View provides a bivariate analysis instead of the univariate analysis supported by the 1D counterpart. Statistical measures are applied to each pair of assigned data attributes, displaying a scatterplot-matrix containing all

combinations of the attributes. One of the main features of this view is the partition-based framework for building and validating regression models [MP13].

- **Function View**
  In order to visually validate regression models for real-time simulations, the Function View provides the visualization of n-dimensional functions in combination with known validation data [PBK10].

In order to support interactive visual exploration, the system is based on a multi-threading architecture [PTMB09], which ensures immediate visual feedback to interactions as well as scalability to very large data sets. This architecture is designed to guarantee responsiveness to the user at all times, and to provide visual feedback as much and as quickly as possible. VISPLORE further supports a Focus+Context system (see Section 2.2.4), directly connected to interactive brushing. In this regard, four data layers are defined:

- **Focus**
  The focus layer contains all data entries currently hovered by the user. The implemented brushing technique called Peek Brush [BP10] ensures scalability and interactivity of this layer.

- **Current Selection**
  This layer contains the entries of all applied selections. Selections can be defined by interactively brushing data entries, and can be combined by logical operations such as AND, OR and SUBSTRACT.

- **Context**
  In some cases, not only the selected data entries are important for an analysis, but also their context. The context layer contains data entries which may be relevant to the selection. In the example of a temperature time series, where the temperature around midday is selected, the context may include the whole day of the selected data entries.

- **All Entries**
  This layer simply contains all entries loaded into the system.

As only a small subset of available statistical methods or data transformations are implemented in VISPLORE directly, an integration of scripting frameworks and languages such as Matlab, R or Python allows for much more advanced data operations. Regarding the implementation of the diploma thesis, this integration provides the foundation for defining more sophisticated data checks, as implementing all needed data checks directly in the software would be impracticable.
Kehrer et al. [KBFP12] provided a generic model for the integration of the statistical computation package R in a visual analytics system, which is implemented as a plugin in VISPLORE. Figure 2.18 shows the basic workflow of the integration (a) and the how

it is implemented in VISPLORE (b). The interaction loop enables for an interactive creation and evaluation of statistical models, by combining the comprehensive collection of statistical methods and calculations provided by R with the dynamic graphics generated by VISPLORE. On the part of VISPLORE, the R *object browser* ensures the synchronization between both environments by showing the objects of the R workspace, whereas the R *console* is used to write R commands and scripts.

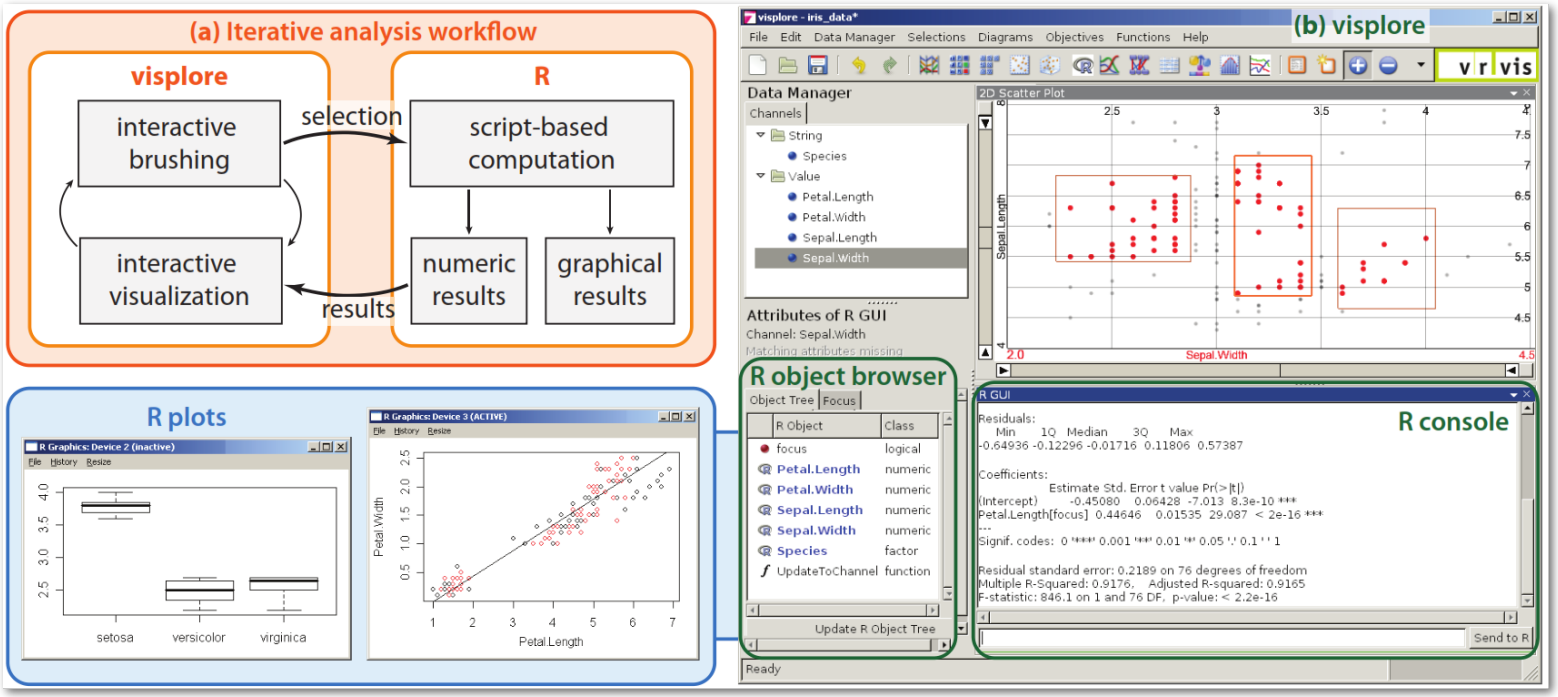


Figure 2.18: The integration of the statistical computation package R and VISPLORE [KBFP12]. (a) shows the iterative analysis workflow of the integration, (b) shows the implementation in VISPLORE consisting of the R *object browser*, which is used to show the objects of the R workspace and for the synchronization between both environments, and the R *console*, which is used to write R commands and scripts.

CHAPTER 3

# Task and Requirement Analysis

This chapter gives a characterization of the problems which arise when dealing with data quality issues, along with a definition of the key tasks and requirements for visual data quality assessment.

## 3.1 Methodology

In order to extract all the requirements for a visualization system supporting data quality assessment, collaborations with partners in the energy sector (since 2012) and the Austrian healthcare system (since 2010) helped to understand sub-tasks of data quality assessment in the context of the respective domains. In the energy sector, time series data including power supply and demand, prices, and meteorological measurements support tasks like power grid control and risk management [MP13]. Frequent data quality problems include missing or outlying values caused by sensor malfunction. In healthcare, routinely collected accounting data is increasingly used as a basis for decision making and planning [PMZ+13]. In this context, typical data quality problems include inconsistencies in names, duplicate entries, missing values, and anomalies introduced by the integration of data from multiple sources. In both domains, data is acquired on a regular basis and must be processed in intervals ranging from days to months. The routine quality assessment of newly acquired data based on automated plausibility checks is thus an important, frequent, and time-critical activity in both domains.
The tasks and requirements, as defined in the following sections, were identified during collaborations with the previously stated partners, based on insights gained from interviews, contextual inquiries [HJ93], and data analysis sessions with domain and data management experts.

## 3.2 Problem Characterization and Task Analysis

To find out whether a data value is plausible or not, can often only be determined by examining the context of the value. Let us have a look at the example temperature time series as seen in Figure 3.1. A value of 0 °C may be plausible if it was measured in winter, however, the context of the measured value clearly reveals the value as implausible. The value not only is uncommon in the context of August, but it also diverges by a large amount from the values measured immediately before and after, thus possibly indicating a temporary sensor failure.

Implausible data values such as the one from the previous example can be detected by *plausibility checks*, which are used as key entities in this thesis. A detailed explanation of the form and structure of such checks is given in Section 4.2. Plausibility checks can be defined for a variety of possible data quality problems ranging from a rather simple detection of missing values to complex outlier detection algorithms. Identifying and creating a suitable set of plausibility checks for a given data set is a challenging task. After identifying possible data quality issues by domain experts or data analysts, who have a special knowledge of the properties of the data, an initial set of checks for a data set may be defined. However, even the most sophisticated checks may not detect all data problems and suffer from false negatives and/or false positives. As an example, even a straightforward check, which is based on a region representing "normal" values, marking any data entry outside of this region as implausible or anomalous can be very

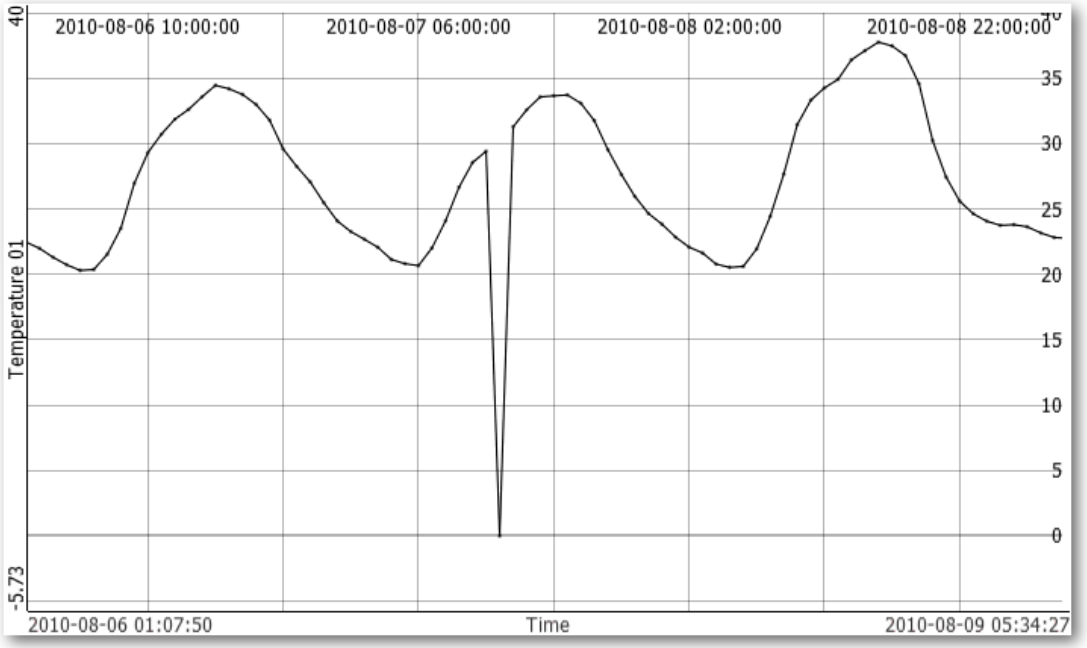


Figure 3.1: An example of an implausible value in a temperature time series.

challenging. The boundary between normal and anomalous behavior may not be precise, resulting in anomalous observations close to the boundary actually appearing as normal, and vice-versa. Another challenge is noise in the data which may be similar to anomalous behavior [CBK09]. In most cases, the domain expert can only make an initial guess for the value of a parameter like the boundary of such a region. Oftentimes, the number of false negatives and/or false positives exceeds an acceptable value, or, on the other hand, not all occurring anomalies are detected by the correspondingly defined check. The parameters of the checks therefore need to be optimized to reduce these errors. In order to support this task, a visualization system for data quality assessment should support the exploration and refinement of the values of plausibility check parameters for better results. Providing immediate visual feedback after parameter changes enables such an exploration along with an access to the distribution of the results at various levels-of-detail.

Analyzing the data quality of a large data set with dozens of attributes may soon end up in a large number of plausibility checks, caused by the diversity of possible data quality issues. Just to name a few, there may exist plausibility checks for missing values, local or global outliers, or sudden value changes as seen in Figure 3.1, and each of these checks may be applied to each attribute in the data set. From this resulting large number of plausibility checks arises the risk of losing track of all computed check results. To address this issue, it is important for a visualization system for data quality assessment to provide a simple and easy to understand overview of the overall analysis result, as well as to provide access to multiple levels-of-detail facilitated by drill-down possibilities to single or a subset of plausibility check results. This includes a flexible summarization of check results, for example by problem types, involved data attributes, or the severity of detected issues. The challenge for the visualization in this case is to work independently of the number of visualized checks and detected indications. A sub-task in this case is to use the overview of the overall analysis result to generate reports of the quality of a given data set for project managers, stakeholders, or collaborating departments. For such reports it is additionally important for the complexity of the visualization to support also non-expert users.

One of the most important tasks, identified during interviews with domain and data management experts, is to identify whether the data quality is sufficient for the data set to be used for the current task at hand, ultimately increasing the confidence in the results of the performed task. In some cases, the data quality may be sufficient for certain tasks, but insufficient for others. Equally important for the experts performing a specific task is to identify potentially needed preprocessing steps such as a selection of a “good enough” data subset or a cleansing of implausible data values. A visual overview needs to be designed in a way that the expert immediately can identify whether the data can be used as-is, and additionally providing means to select data subsets without data quality issues. Enabling the access to details of detected data quality issues allows experts to determine whether and how implausible values may be cleansed in a further step before using them for the current task at hand.

Another important task is to generate hypotheses about possible causes of indicated

data quality issues. Finding causes of implausible values helps data providers to improve the overall quality of their data sets and further reduce the costs of data analysis and cleansing steps. In order to generating hypotheses, a user for example may want to identify data attributes related to detected data quality problems. Considering the photovoltaic example set, which is used throughout this thesis (see Section 4.1), detected missing values of a sensor, measuring the photovoltaic production of a power plant, may correlate with low temperature values measured by another sensor, as the given sensor may only work in a certain temperature span. Another example could be a correlation between missing values of a sensor with a certain day of month where all sensors are turned off for maintainance. To support the generation of hypotheses about possible causes, the visualization may directly include information about the distribution of the affected data records over time or other attributes of the data set. For the previous example, the visualization may provide means to explore the temporal distribution of the indicated data records, in order to see a possible correlation between data entries with indications and the time at which they occurred. This further facilitates the communication of found hypotheses to data providers.

## 3.3 Requirements

Based on the previous problem characterization and the task analysis, the following list presents requirements for the design of a visualization system to support data quality assessment:

- **Requirement 1**: Overview of the overall data quality by summarizing the proportion of data indicated by plausibility checks.
- **Requirement 2**: Flexible summarization of check results, for example by problem types and involved data attributes.
- **Requirement 3**: Characterization of data quality problems in terms of properties of affected data records, including their distributions over time and regarding quantitative and qualitative data attributes.
- **Requirement 4**: Efficient selection of data subsets based on check results.
- **Requirement 5**: Fast access to details of affected data records, data attributes, and plausibility checks.
- **Requirement 6**: Suitable visual analysis and validation of indicated data quality problems in the context of data without indications.
- **Requirement 7**: Export of data subsets to external data sources.
- **Requirement 8**: Scalability for up to millions of data records, hundreds of data attributes, and thousands of plausibility checks.

- **Requirement 9**: Scalable visual complexity supporting non-expert as well as expert users.
- **Requirement 10**: Flexible in-situ identification, definition, and modification of plausibility checks with immediate visual feedback.

CHAPTER 4

# Data Model

The following sections describe the underlying data model of the Data Quality Overview. The data used for data quality assessment in this diploma thesis is assumed to be a generic multivariate data table consisting of $n$ data records (i.e., rows), $p$ data attributes (i.e., columns) and $n \times p$ data values (i.e., cells). The data attributes may be of arbitrary type such as quantitative, categorical, or time stamps. This includes, e.g., time series data and event logs. The data model of the visualization approach consists of a hierarchically structured set of plausibility checks, which are applied to a set of data attributes.

## 4.1 Example Data: Photovoltaic Production

In all examples throughout this thesis, a real, but anonymized data set from the energy sector is used. The data set consists of the production data of 95 photovoltaic power plants (data attributes PV01 - PV95), hourly measurements of global radiation and temperatures from 20 weather stations as well as humidity, wind speed, gust speed, wind direction, air pressure, and dew point for four weather stations, from May 2010 to April 2011. This data is a typical example of sensor measurements that require a regular data quality assessment prior to tasks such as forecasting or statistical modeling.
We chose this data set for the examples in this thesis, as it contains data quality issues such as missing data (Figure 4.1a and b) or anomalies as classified by Chandola et al. [CBK09]: point anomalies (Figure 4.1b), contextual anomalies such as sudden implausible drops in temperature as seen in Figure 4.1c, or collective anomalies (Figure 4.1d).

## 4.2 Plausibility Checks

*Plausibility checks* (subsequently also simply called "checks") are the key entities in the context of this diploma thesis. In order to detect data quality problems such as missing values, syntactical errors, constraint violations, or anomalies, plausibility checks are

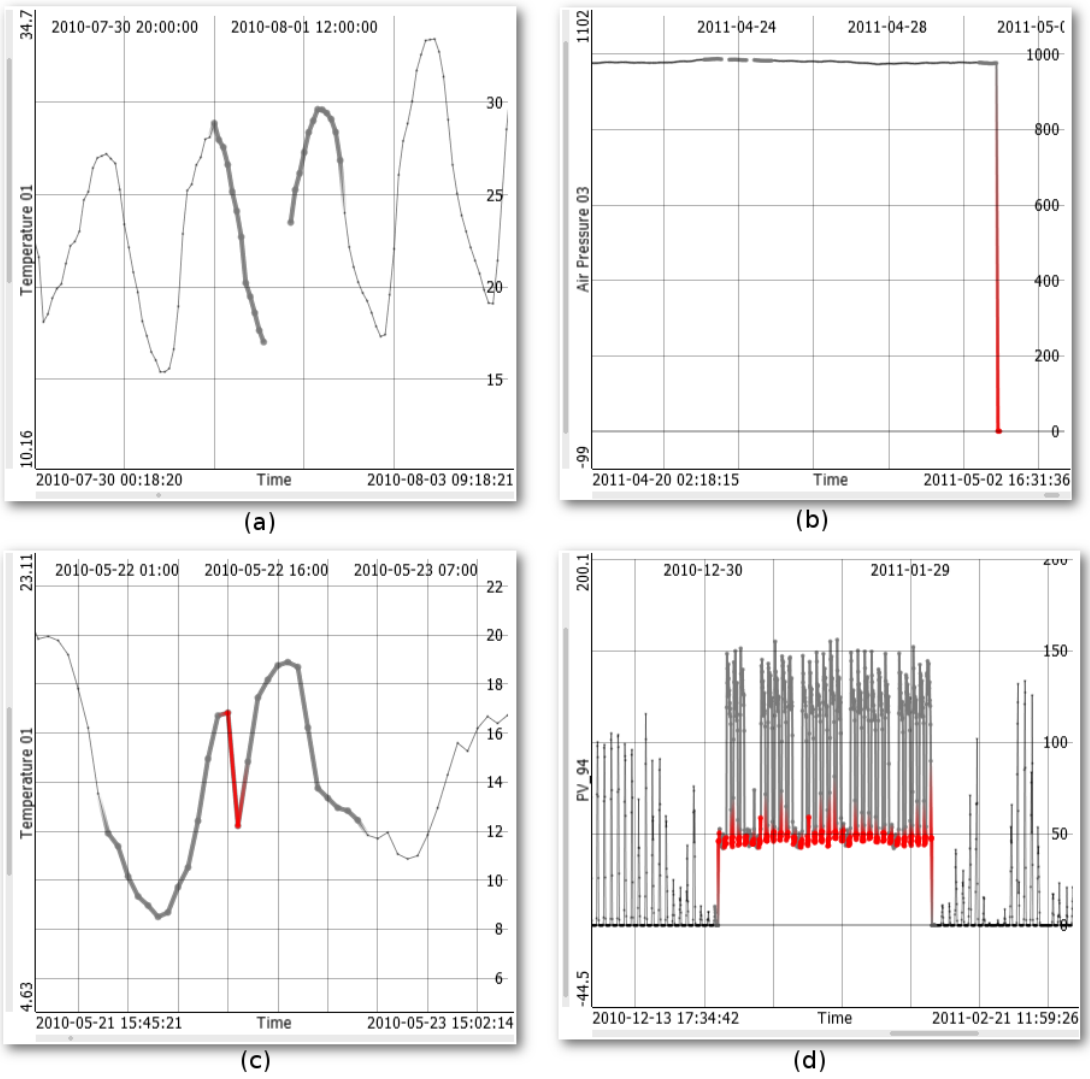


Figure 4.1: Anomalies contained in the example data set, including (a) missing data, (b) point anomalies, (c) contextual anomalies and (d) collective anomalies.

applied to the values of one or multiple data attributes of a data set, in order to produce boolean- or real-valued results. A simple check for missing values, for example, produces a boolean result for each value of a given data attribute, whereas more sophisticated checks, e.g., advanced anomaly detection algorithms, may produce a real-valued result for each data record. Plausibility checks frequently also consider multiple independent variables for determining the plausibility of one dependent variable. In the context of photovoltaic production, an example could be a check to detect non-zero production values at nighttime. In addition to the production value of a photovoltaic power plant (dependent variable), a time stamp is needed as additional input variable (independent variable) in order to assess the plausibility of the production value.

Each plausibility check consists of a *rule implementation* along with attributes such as a *classification* (for example "constraint violation" or "anomaly", see Section 4.2.1), *implementation-specific parameters* (for example the value of a threshold), and *meta-information* such as user-defined tags. The system currently supports four built-in rules and a user-defined rule based on scripting. The built-in rules comprise:

- a check for missing values which are explicitly supported by the multivariate data model in VISPLORE,
- checks for boundary violations defined by one or two thresholds (a lower and/or an upper boundary),
- selection-based checks defined by interactively brushing data in one of the provided visualizations of VISPLORE, and
- model-based anomaly detections operating on regression models, which can be built interactively in the system [MP13].

By defining script-based checks, users may apply any check on the data by using the scripting languages R, Matlab or Python. For more information about the integration of these scripting frameworks and languages in VISPLORE, see Section 2.5. The case study (Section 7.1) describes specific examples for script-based checks for the example data set. Parameters of each plausibility check, such as thresholds in the case of a check for boundary violations, or the script text, in the case of script-based checks, can be modified at any time and immediately trigger a re-computation of the check, ultimately resulting in an update of the visualization.

### 4.2.1 Classification

The *classification* of a check refers to the semantic class of data quality problem it detects. Motivated by existing data quality taxonomies (see Section 2.3.1), the classification is defined hierarchically, where *class* denotes the category of the first level, while *sub-class* denotes a second level category. The employed classification scheme can be defined as suitable for a particular application context and can even be extended by users at run-time. In the context of the data which is used throughout this thesis, useful first-level classes include

1. “missing”, e.g., for NULL values,

2. “syntactic error”, e.g., for unknown time stamp formats,

3. “constraint violation” for impossible values or relations between values, and

4. “anomaly” for suspicious yet potentially correct values.

A check for univariate outliers, for example, could be classified by the class “anomaly” and a *sub-class* “outlier”. This classification scheme was defined for the examples in this diploma thesis, however, depending on the data and application context, also other classification schemes can be used. The visualization (see Chapter 5) does not rely on any particular scheme. However, using an appropriate classification scheme for the application context at hand is an important aspect for the visualization.

### 4.2.2 Plausibility Check Results

The output of a plausibility check may either be boolean-valued or real-valued. However, in order to enable a unified visual representation, real-valued checks discretize the output of their evaluations by one or more thresholds to obtain boolean-valued outputs. A single check, such as the previously mentioned check for outlier detection, may thus have multiple boolean-valued outputs for each data record, discriminating increasing tolerance thresholds. In order to support these thresholds, each output is labeled by an associated *severity level*. Throughout this thesis, three severity levels called *uncritical*, *warning*, and *critical*, are used, which can be modified and extended by users at run-time for specific application scenarios. In the case of real-valued outputs, this enables for a comparison of defined thresholds, e.g., for a sensitivity analysis. For boolean-valued checks, severity levels are useful to, e.g., mark missing values for some data attributes as critical and uncritical for others, for example “comment” fields of survey data.

Within the data values of the target data attributes, a check may produce an output per single data record, per data category, i.e., a meaningful subset of data records, or for all data records. This definition generalizes the granularity levels of the quality space defined by Ward et al. [WXYR11]. For a consistent treatment of all checks and their combinations, however, we break down the results of all checks on a record level. For example, each result of a per-category check is assigned to all data values within the target data attribute of the respective category.

In subsequent sections, in order to simplify the terminology, we refer to each set of boolean-valued outputs of a check as *plausibility check result* (or simply “check result”). Using this terminology, the previously mentioned check for outlier detection may have multiple check results, one for each severity level.

In addition, we refer to the subset of data values marked by a check result as *check indications*. We decided for this term because it is neutral and does not forestall assessments about correctness, e.g., in contrast to “error”.

## 4.3 Hierarchy of Plausibility Check Results

A hierarchical structure is a key concept in order to ensure scalability of the visualization [EF10] (see Section 2.2.1), which is one of the requirements specified for the design of the system (*Requirement 8*, as defined in Section 3.3). The data model of the Data Quality Overview structures the set of plausibility check results in a hierarchy, each node consisting of a set of check results. The root node of the hierarchy contains the set of all existing check results, while adding a *hierarchy level* refines the set of results by one of the following properties of a check:

1. The *class* of a check result such as "missing" or "constraint violation"
2. The *sub-class(es)*, i.e., further levels of the hierarchical classification such as "outlier"
3. The *target data attributes* (dependent variables)
4. The *severity level* of check results (e.g., *uncritical*, *warning* and *critical*)
5. User-defined *tags* on the checks
6. The *indication occurrence* as distinction of checks with and without indications
7. Groups of checks defined by identical *indication patterns*

Figure 4.2 shows a schematic representation of an example hierarchy of the Data Quality Overview. For an easier illustration of the concept, the example consists of 10 check results operating on a table with only 8 data records per data attribute. The root node simply consists of all 10 check results. The hierarchy is refined by two additional hierarchy levels, where *L1* refines the set of check results by their check class, which

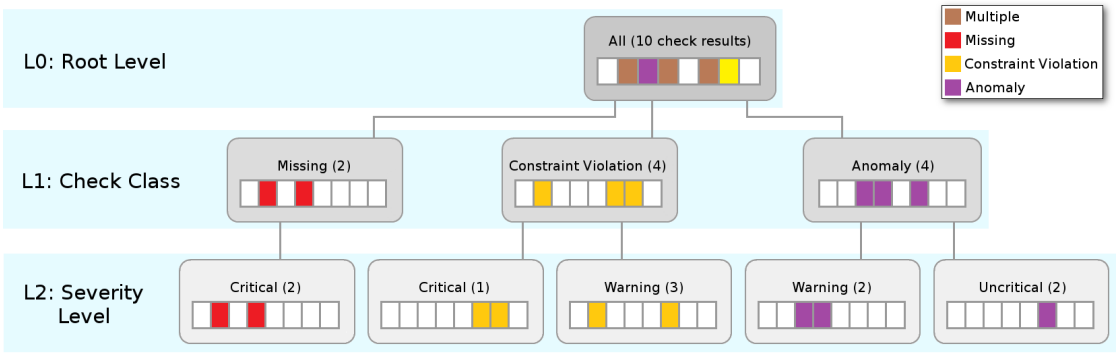


Figure 4.2: Schematic representation of an example hierarchy of plausibility check results. The root node contains an aggregation of all plausibility check results. Additional hierarchy levels refine the hierarchy by a property such as the check class (L1) or the severity level (L2).

results in a node for “Missing”, one for “Constraint Violation”, and one for “Anomaly”. The level *L2* further refines these nodes and their set of check results by the respective severity level. Nodes are only created for properties which occur in the given set of check results, for example when refining the node “Missing” by the severity level, only one node for “Critical” is created, as both checks in the node have this severity level. Nodes for “Warning” and “Uncritical” are omitted in this case.

As one design decision, the hierarchical definition does not imply disjoint check result sets. For example, a subdivision by the target data attributes defines one hierarchy node per data attribute. Check results with more than one target data attribute will thus be referred to by multiple hierarchy nodes. Depending on the definition of the subdivision, enforcing a disjoint partitioning of check results would be possible. For example, a subdivision by target data attributes could define a node for each occurring combination of data attributes. However, user feedback suggested to favor understandable nodes. Moreover, no aspect of the Data Quality Overview relies on a disjoint subdivision of check results.

### 4.3.1 Check Result Aggregation

In order to compute measures such as the relative frequency of check indications, each node aggregates the set of underlying check results. Per default, the check results within each node are aggregated on a record-level (*record-based* aggregation), marking those data records as implausible which are indicated in at least one of the underlying check results (a logical *OR*-combination). Following the example of Figure 4.2, the root node, containing all results, shows check indications in 5 of the 8 data records. Each one of these indications emerges from at least one of the underlying check results. Moving further down in the hierarchy shows how the indications are distributed between check classes and severity levels.

However, if the checks of a node target multiple data attributes, the distribution of values with data quality problems may not be visible due to potential masking between the data attributes. Therefore it is possible to aggregate check results in terms of data values instead of data records (*value-based* aggregation). In this case, the results of the plausibility checks are first aggregated on a record level for each data attribute, and the overall number of check indications is then computed by adding the resulting number of indications of each target attribute.

Let us explain the difference between the aggregations in a simple example as seen in Figure 4.3. In the example, we see results of two checks (“Missing” and “Boundary”) applied to the data attributes “Temp01” and “Temp02”, which represent two temperature time series in the example data set. In order to compute the relative frequency of check indications for an aggregation of the four given check results, in record-based aggregation, all results are combined on a record-level, resulting in check indications in 6 out of 8 data records (75%). However, when using value-based aggregation, the results are first aggregated on a record-level for each data attribute, and the overall number of check indications is computed by adding the number of indications for each data attribute. In

this case, we obtain 9 indications out of 16 total entries ($2 * 8$, as the indications of 2 data attributes are aggregated), which results in a relative frequency of $56,25\%$.

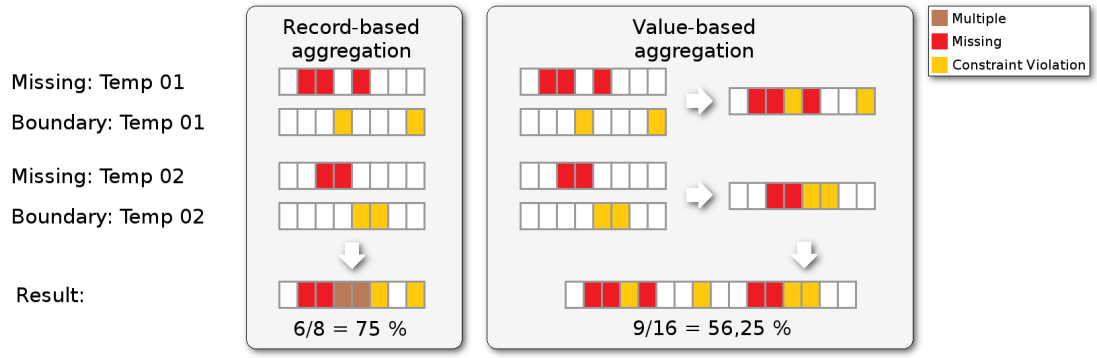


Figure 4.3: Schematic representation of the difference between record-based and value-based aggregation.

CHAPTER 5

# Visualization

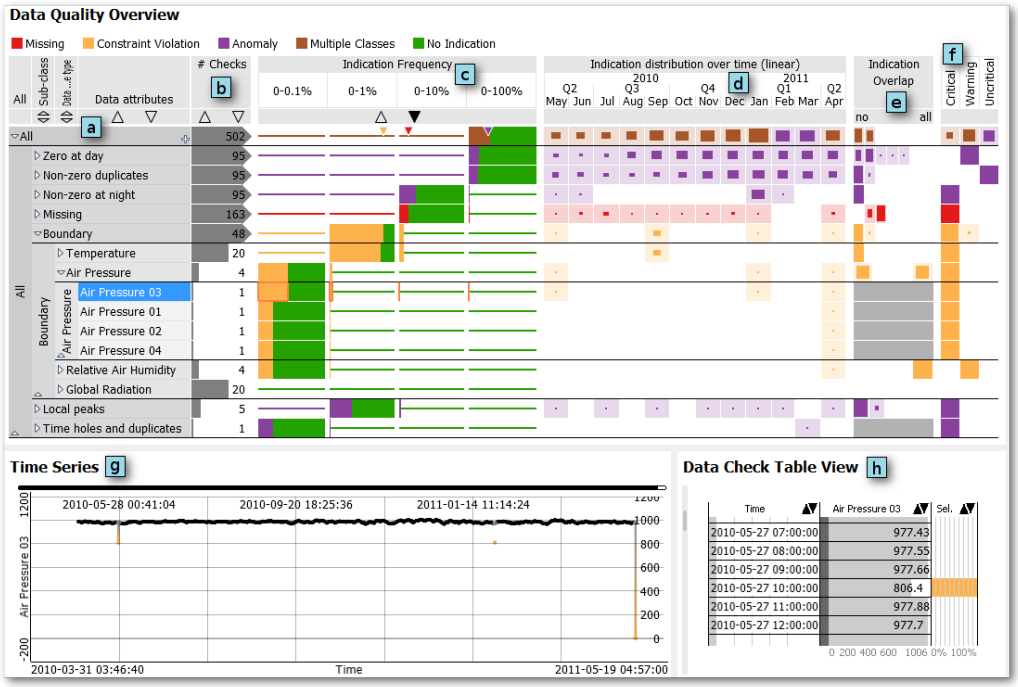


Figure 5.1: The Data Quality Overview as used for data quality assessment in the energy sector. (a) Hierarchical aggregation of plausibility check results by their properties. The columns display information for each row such as (b) the number of checks, (c) the percentage of check indications, (d) the distribution of check indications over time, (e) the overlap of check results in each row, and (f) the severity of the data quality problems. (g, h) Linked views provide details of the selected indications.

This chapter describes the Data Quality Overview, a novel visualization method for check-based data quality assessment. The key idea of the visualization is to structure the overview of data quality problems as a hierarchical table of plausibility check results (see Figure 5.1). Each hierarchy level is defined by a check attribute like the class or involved data attributes, which partitions the set of plausibility check results in each level (see Section 4.3). The table rows thus correspond to results of single checks or the aggregation of the results of multiple checks. The level-of-detail can be adjusted by adding or removing hierarchy levels, and by expanding or collapsing nodes in the hierarchy. The table columns provide information about the check results, including the relative frequency of indications per class, the distribution of indications over time or other data attributes, and the overlap of indications by multiple checks. The table can hence be interpreted as an intersection of the check-space and the multivariate data-space.

The design of the Data Quality Overview is guided and motivated by the identified tasks and the resulting requirements defined in Chapter 3 for an application to the data and plausibility checks as described in Chapter 4. The decision for a table-based layout was made because it is a familiar concept to users and enables independent encodings of multiple structurally different properties of checks as columns [LGS+14]. Moreover, control matrices for data quality assessment are suggested by the literature [Pie04].

## 5.1 Rows: Hierarchically Structured Checks

Each row in the tabular layout of the Data Quality Overview is the representation of a node in the hierarchical structure as defined in Section 4.3. The level-of-detail of the hierarchy can be chosen interactively: it is possible to add check properties as additional hierarchy level, and to remove and reorder existing levels. Each node can individually be expanded or collapsed. Following the example in Figure 5.2, initially, all check results are aggregated as a single row, which corresponds to the root node of the hierarchy (Figure 5.2a). To investigate check results by their target data attributes, the user can add a hierarchy level for this check property (Figure 5.2b). Adding a further refinement by check class or sub-class (see Section 4.2.1) enables the user to drill down on data attributes, which are affected by indications in multiple classes (Figure 5.2c). The order of the two hierarchy levels can be switched by drag-and-drop, which re-defines the hierarchy by discriminating checks first by sub-class, and then by the target data attributes (Figure 5.2d). This concept is similar to interactive approaches for pivotizing categorical data [Tab] and is essential to meet *Requirement 2* (flexible summarization of check results) and *Requirement 8* (scalability for the number of attributes and checks, see Section 3.3).

Visually, the hierarchical subdivision of checks is located in the left-hand part of the layout. The table header shows one segment per hierarchy level, and is labeled with the name of the check property, which defines the subdivision in the level. The root level of the hierarchy, which is always visible as the left-most column, is labeled by “All”. This level simply consists of the root node of the hierarchy, containing the set of all currently available plausibility check results. The set of available checks can be restricted by a filter

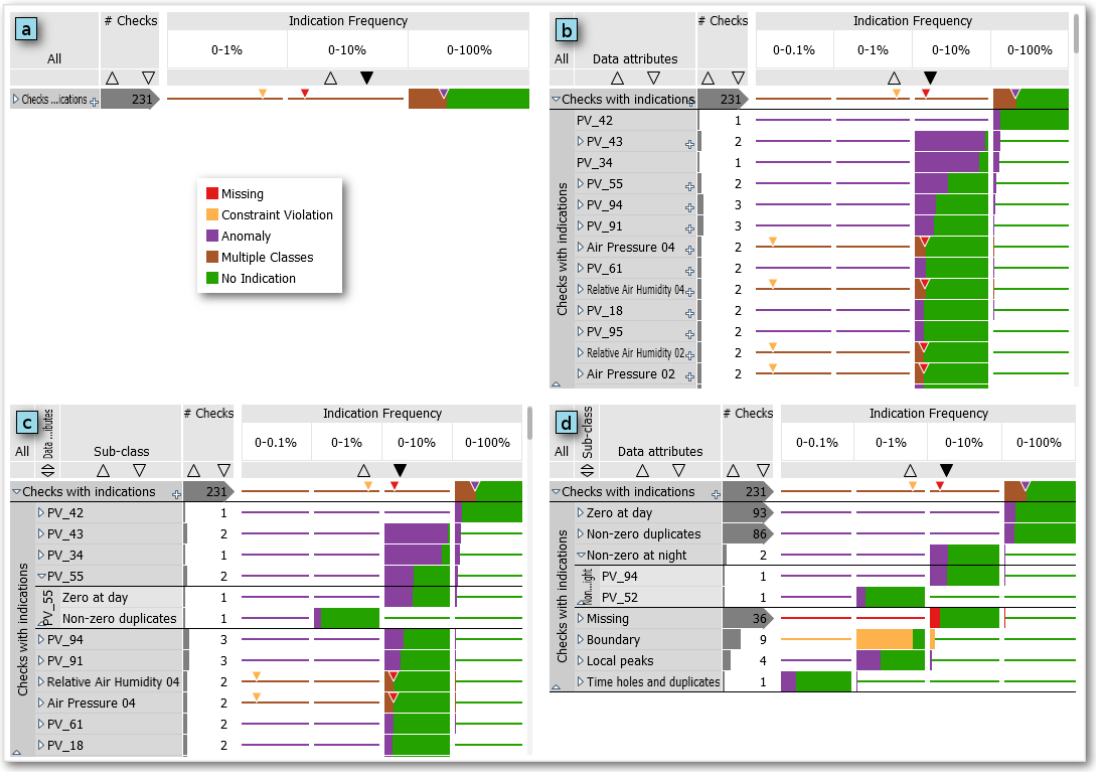


Figure 5.2: The frequency of check indications at four stages of a drill-down scenario. (a) Aggregation of all checks, (b) drill-down by targeted data attribute, (c) a local drill-down on the attribute "PV_55" by check classification, and (d) another local drill-down after swapping the hierarchy levels.

(see Section 5.3.2), and the root node is named accordingly. In the example of Figure 5.2, the Data Quality Overview only shows checks which resulted in indications in at least one data record, and therefore the root node is named "Checks with indications".

The width of each segment can be adjusted, and all segments, except for the "All" segment, can be re-ordered by drag and drop, which re-defines the hierarchy accordingly (see the example in Figure 5.2). Hovering a segment overlays temporary controls (*Hover Triggered Controls* – see Section 5.3.3) which allow to replace one check property by another one, to remove hierarchy levels, or to globally expand or collapse all nodes of the hovered hierarchy level.

For controlling the visible set of table rows, each hierarchy node can individually be expanded or collapsed, which simply shows or hides its child nodes. This provides additional flexibility in specifying the level-of-detail of the visualization, in addition to the definition of the hierarchy levels. Table rows are displayed for all nodes including intermediate nodes and the root node. This was different for early design stages (see Chapter 8), where table rows were only displayed for leaf nodes, and intermediate

nodes were replaced when expanded. However, user feedback indicated a preference for extending rather than replacing rows at drill-down, as they are more familiar with this concept, which is frequently used in the graphical user interfaces of file explorers in operating systems. Another advantage of this approach is that the context of each node in the form of the parent or sibling nodes is preserved. In order to examine the results of each check individually, each leaf node containing more than one check result can be expanded, showing a table row for each of the check results. As a result, the Data Quality Overview can optionally be configured as a non-hierarchical table of check results, by simply expanding the root node of the hierarchy ("All") without adding further hierarchy levels.

The visualization of table rows reflects their hierarchical definition by an indentation according to the user-defined width of the header segments and is additionally enhanced by a gray background, whose intensity is reduced for each hierarchy level (see Figures 5.1 and 5.2). Each node displays the respective check property, which defines the partition in the according hierarchy level, as horizontal text. Expanded nodes additionally show their name as vertical text in order to provide the context for their child nodes. Intermediate nodes display arrows for expanding or collapsing the respective sub-tree. Leaf-nodes containing more than one plausibility check result provide a click-able "+" sign for specifying an additional hierarchy level and expanding the node as local drill-down. Horizontal lines support the perception of the hierarchical structure of table rows across all columns of the Data Quality Overview, however, lines are only drawn between rows with different parent nodes in order to facilitate the perception of siblings (see Figure 5.2).

## 5.2 Columns: Aspects of Check Indications

The Data Quality Overview supports multiple types of columns. Each column provides information for the subset of check results in each row. In order to support a flexible data quality assessment for multiple scenarios and target user groups, columns can be added, removed or replaced at all times, similar to hierarchy levels. The complexity of the visualization can therefore be steered by configuring the number and type of visible columns, thus supporting expert users as well as non-expert users (*Requirement 9*).

As a general design decision, hue consistently encodes the class of data quality problems in all columns, as user feedback revealed that the class of detected indications should be visible regardless of the drill-down of rows. Using hue as visual attribute to encode qualitative information is suitable for at most 12 categories [War12], which in the case of "class" is typically the case. For some columns, this scheme is extended by brown indicating "multiple classes", and green for representing "no indications" (see Chapter 8 for more details about this design decision).

### 5.2.1 Indication Frequency

The relative frequency of data records with indications is essential information, and also represents the first of our defined requirements for visual data quality assessment

(*Requirement 1*). The *Indication Frequency column* displays this information for each row of the hierarchy by computing the percentage of data records having indications in one or more of the contained check results. The main challenge for the visualization in this column is the largely varying scale of the percentages. Indicating a few anomalous data records for one data attribute is often as important as conveying large percentages of, e.g., missing values for another one. Using bars with linear scaling may be suitable for displaying large percentages of data indications, however, checks frequently result only in a small percentage of indications, which may not be perceptible, as (depending on the width of the column) the size of the bar may even fall below one pixel in screen space. In such a case, the respective cells may fail to display the relative frequency, even if the underlying set of check results contains indications. Using bars with logarithmic scaling would solve this problem, however such a scaling is much harder to interpret, as it makes it difficult to quantitatively compare different values across the visible rows [HSBW13].

As a solution, the concept of scale-stacked bar charts [HSBW13] is adopted. The key idea of scale-stacked bar charts is to use multiple scales to cover a large value range, while simultaneously preserving the ability to compare values using a linear mapping within each scale. In the case of the Indication Frequency column (see Figures 5.1 or 5.2 for examples), each order of magnitude is represented as a sub-column in decimal steps, i.e., $0-100\%, 0-10\%, 0-1\%, 0-0.1\%$, and so on. The smallest order of magnitude is determined by $1/N$. The main advantage of this concept is that it allows to compare the computed percentages with linear scaling, while small percentages still remain perceptible.

For each table row, the smallest scale containing the percentage of indicated data problems displays the stacked bar in full height. This supports an immediate perception of the magnitude of the percentage, even in case of little horizontal space for the Indication Frequency column. Scales of higher magnitudes still show the indications in full height, as by definition they cover at most 10% of their width, and for the rest of the sub-column a thin line in the color of "no indications" is drawn. Hence, all orders of magnitude – and particularly the 100% sub-column – can still be read as linear bar charts, and therefore allow for a comparison across all rows. Cells of sub-columns, whose order of magnitude is smaller than the percentage to be displayed, show a thin line in the color for indications. For an easier understanding, let us follow the example depicted in Figure 5.2b, where the percent of data records with indications of the photovoltaic power plant PV_43 lies at approximately 9%. As the $0-10\%$ sub-column is the smallest scale containing this percentage, it is displayed in full height, while sub-columns with smaller magnitude display a thin line in the color of the class of the underlying checks (in this case purple for *anomaly*). The $0-100\%$ sub-column, however, still displays the percentage of indications at full height, in order to allow for an easier comparison over all rows, and additionally displays a thin green line for "no indications".

If all indications of a row share a single class (as in the previous example), by chance or by hierarchy definition, their encoding uses the hue of that class. Otherwise, the hue for "multiple classes" is applied for all indications. This was different in previous versions of the Indication Frequency column, where the percentages of the classes were stacked in each cell. However, this concept posed difficulties in comparing the percentages between

each class, as well as over each row. Additionally, a problem was again the visibility of small percentage values in higher orders of magnitude. In order to prevent the loss of information by introducing a color for "multiple classes", small triangles encode the percentage of data records with indications for each individual class, colored by the hue of the respective class (see Figure 5.2). Again, for an immediate perception of the magnitude of the percentage and to avoid visual clutter and redundancy, the triangles are only displayed in the smallest order of magnitude containing the percentage. This technique enables users to quickly perceive the overall percentage of data with indications in each row by considering the drawn bar, as well as to get an idea which classes the percentage is composed of and how it is distributed among those classes. The displayed triangles also enable the selection of all data records with indications of the respective class (see Section 5.4). In order to magnify the percentages of a specific class for a quick comparison over all rows, a class can be hovered in the color legend of the Data Quality Overview, overlaying bars of the hovered class in each row.

Expressing the relative frequency in terms of data records is appropriate for many downstream analysis steps and is used as default. In this case, the results of the plausibility checks within each row are aggregated using *record-based* aggregation (see Section 4.3.1). Alternatively, we support to express the indication frequency in terms of data values instead of data records (*value-based* aggregation). The visualization itself is independent of the chosen aggregation, as only the computation of the percentages is changed. For a better understanding of the difference between the two aggregation techniques, let us consider a simple example where we have a row containing two checks, each one assessing the data quality of a different data attribute. Now just for explanation, if one of the checks results in no indication, and the other results in indications in all entries of the target data attribute, the resulting percentage would lie at 100 % in the case of record-based aggregation, as the overall result is a simple *OR*-combination of both check results. However, in the case of value-based aggregation, the percentage would lie at 50 %, indicating that half of the data entries of the target attributes are records with indications.

### 5.2.2 Indication Distribution

The distribution of records with indications is important information in order to assess the quality of a data set. Check indications may follow a specific pattern, examples could be indications only occurring at a certain time of the day, or a correlation of indications with the corresponding value of a given data attribute. For example, missing values of a sensor may directly correlate with a change in temperature measured by another sensor, as the sensor may only work in a certain temperature range. In order to display such information about the distribution of check indications, with respect to temporal, quantitative, or categorical data attributes (*Requirement 3*), the Data Quality Overview provides the *Indication Distribution column*, which is based on a disjoint partitioning of data records (see Figure 5.3).

Partitioning enhances the scalability regarding data size [LJH13] (*Requirement 8*) and supports an analysis in terms of semantically meaningful categories. In the Indication

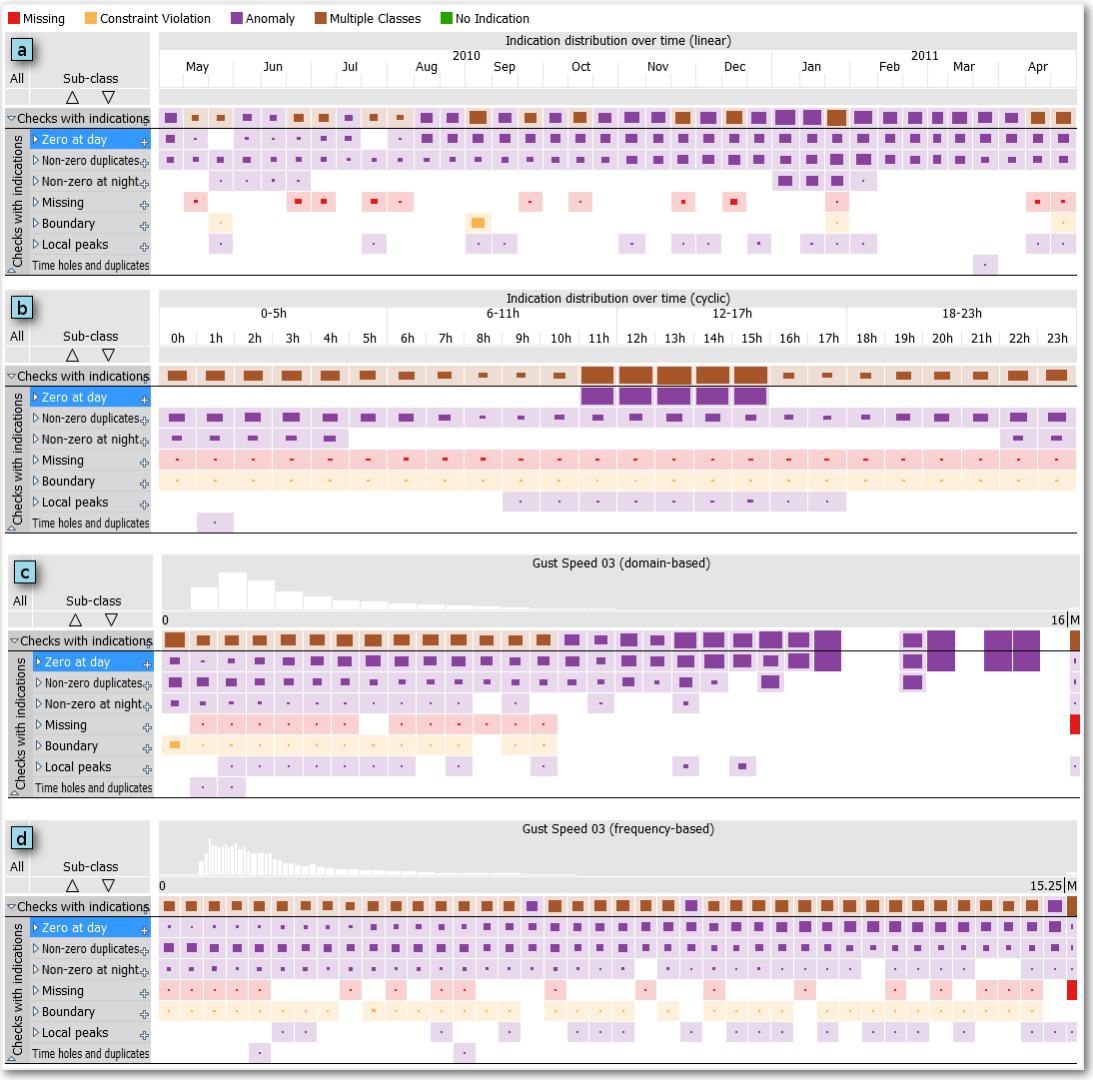


Figure 5.3: Four examples of check indication distributions. (a) Linear temporal partitioning in steps of month-thirds, (b) Temporal partitioning in daily cycles, (c) domain-based quantitative partitioning of Gust Speed 03, and (d) frequency-based quantitative partitioning.

Distribution column, the number of partitions always adapts to the column width and the underlying value range. Users may thus define a level of detail by resizing the column or by restricting the value range using a range slider (see Figure 5.4). During the interaction, the column header showing the partitioning scheme is immediately updated for instant feedback while the table rows are redrawn upon completion of the computation. If the underlying data attribute of the column contains missing values, all partitioning schemes add a partition for these values, which is always shown as right-most partition with fixed size (see Figure 5.3c and d). As a requirement for all data types, user feedback stressed the importance of interpretable partitions, such as months or days in the case of time-based partitions or steps of 5 or 10 in the case of quantitative partitions.

**Linear temporal partitioning** refers to a time-typed data attribute, e.g., a time stamp (Figure 5.3a). Partitions correspond to the temporal units year, quarter, month, third of month, day of month, quarter of day, hour, minute, and second. Depending on the specified data range and the column width, an actual partitioning could, for example, include two years, refined as quarters and months. The column header hierarchically displays these temporal units. A hover-triggered slider [ARP14] enables to restrict the shown range and to optionally define a focal range with more fine-grained partitions [LA94] (see Figure 5.4).

**Cyclic temporal partitioning** also refers to a time-typed data attribute, but defines partitions as cycles of years, months, weeks, or days [AMM+07] (Figure 5.3b). A cyclic distribution supports, for example, a detection of seasons, days of the week, or times of the day with an over-proportional number of data quality problems. Depending on the column width, the chosen cycle is hierarchically sub-divided by appropriate temporal units. A weekly cycle, for example, is refined in terms of days of the week, hours per day, and so on.

**Quantitative partitioning** may reveal relationships between check indications and a quantitative data attribute, e.g., sensor measurements. Depending on the distribution of the data attribute, meaningful partitions can be defined as domain-based (Figure 5.3c) or frequency-based (Figure 5.3d) [MP13]. Domain-based partitions have intuitive boundaries and an equal size in terms of the value domain, e.g., in steps of 10. Frequency-based partitions contain a roughly equal number of values, e.g., 5% of the data. A domain-based partitioning is easier to interpret if the quantitative data attribute is approximately uniformly distributed, while a frequency-based partitioning is more robust in case of skew, peaks, and outliers. For both schemes, the partition boundaries are ultimately defined as thresholds within the data attributes. This ensures that identical values of a data attribute are assigned to the same partition. It also enables a consistent visualization of the partitions as bars of a histogram in the column header. Each bar encodes the position and width in domain space of the underlying quantitative data attribute, and the frequency as area. Taking the vertical space as maximum uniquely defines the height

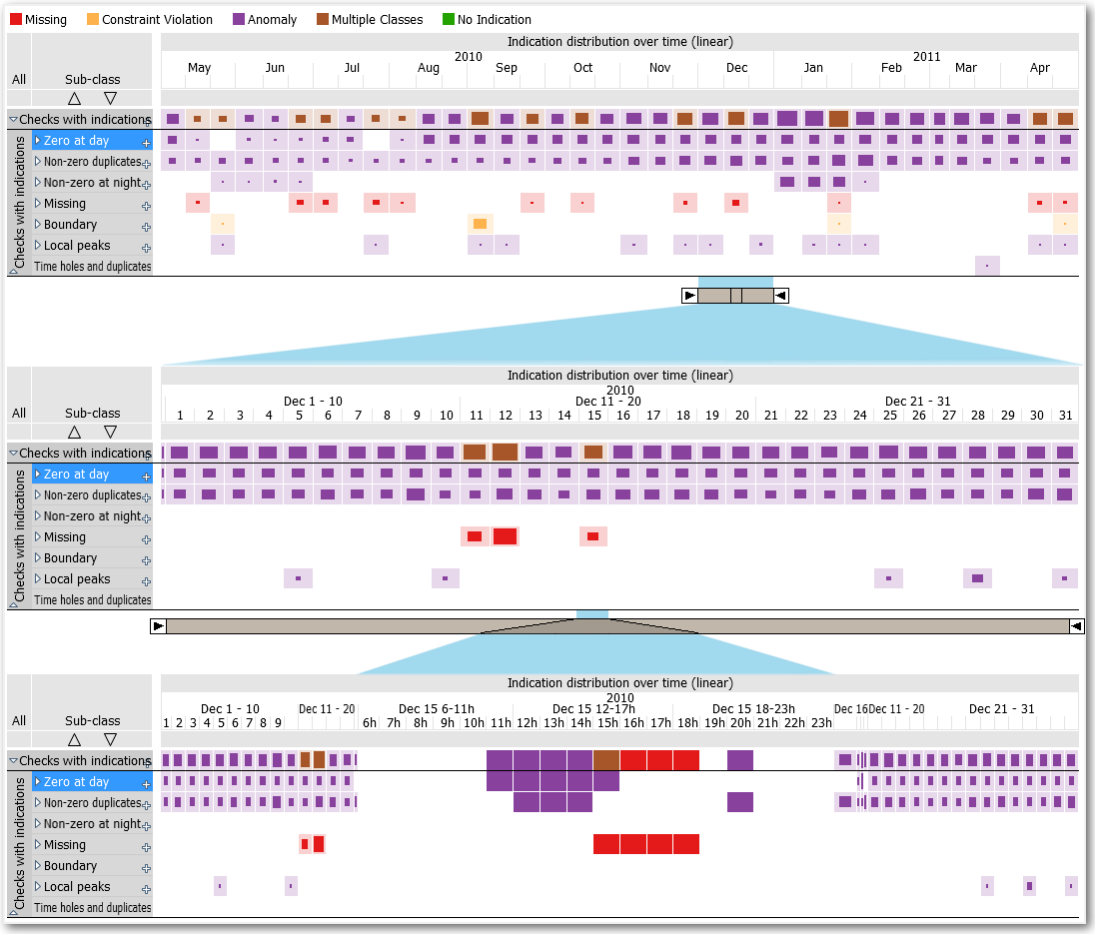


Figure 5.4: Restricting the temporal range for increasing the level of detail. The first step drills down on December 2010, the second one uses distortion for inspecting single hours of December $15^{th}$ in the context of the rest.

of each bar. As for temporal partitioning, a hover-triggered slider [ARP14] may restrict and refine the shown range.

**Categorical partitioning** refers to a nominal or ordinal data attribute, e.g., a political district. The categories directly define the partitions. Nominal categories are ordered descendingly by their frequency, while ordinal data preserves the order. If the number of categories exceeds the possible number of partitions for a given column width, nominal data attributes group the tail as one partition called “rest” while ordinal data combine consecutive categories.

For visualizing the indication distribution, each table row represents the partitions as equally-sized segments, except for when a focal range is defined (see Figure 5.4). Hovering a segment with the mouse cursor highlights the corresponding partition in the column header. Each segment linearly encodes the percentage of contained data records with indications by the area of a centered rectangle. A minimal area of 2x2 pixels ensures the visibility of small percentages. The main reason for this design was that we found centered rectangles as most suitable for supporting comparisons along rows and columns at the same time. Moreover they generate a disconnected pattern that stresses the discontinuous occurrence of indications more intuitively than for example continuous bar charts.
If all indications per segment share a single class, the rectangle is drawn in the hue of that class. Otherwise, the color for "multiple classes" is applied. In order to increase the perceptual discrimination of segments with and without indications, a desaturated background is drawn in partitions with at least one indication. Hovering a class in the color legend of the Data Quality Overview temporarily filters each cell to display the percentage of contained check indications for the hovered class. This allows for a quick overview of the distribution of check indications of a specific class.

### 5.2.3 Indication Overlap

For table rows comprising multiple check results, the overlap of check indications is often relevant to assess on a summary level already if indications originate from a single, multiple, or all underlying check results (see Figures 5.1e and 5.5). The idea of this column is to represent different degrees of overlap as partitions, e.g., for records with indications by a single check result (no overlap), by two results, by three results, and so on. The number of partitions is determined independently for each row, as it depends on the number of check results that lie in a specific row, where each partition comprises data records with indications overlapping in the corresponding number of check results. The horizontal space of each row is then equally subdivided by the respective partitions. The first partition, which contains indications with no overlap in other check results, and the last partition, which contains indications that overlap in all check results of the row, may be of special importance to the user: indications with no overlap may indicate false positives of a check, as they are unique over all check results, whereas indications that overlap in all check results may confirm an apparent data quality problem.

The visual representation of the Indication Overlap column is analogous to the Indication Distribution columns (Section 5.2.2), however, each row of the table is partitioned depending on the number of underlying checks, and therefore may result in a different number of partitions for each row. The column width again determines the level of detail: While the partitions for no overlap (i.e., indication by a single check) and full overlap are always preserved, consecutive intermediate partitions may be combined (e.g., "2 - 5 checks"). A centric rectangle encodes the percentage of indications per partition by its relative area size, the rectangles of all partitions per row thus accumulate to 100%. Additionally, the background supports to discriminate partitions with and without

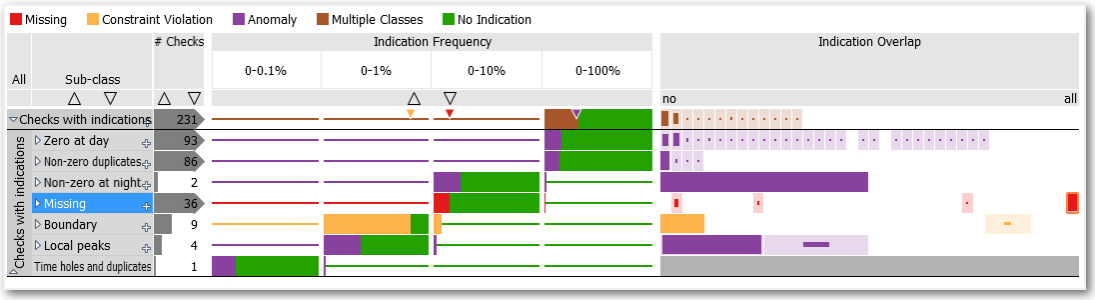


Figure 5.5: Example of a Data Quality Overview including an Indication Overlap column.

indications, whereas a gray background indicates that the column is not applicable because the row contains only a single check result.

Let us have a look at the example as seen in Figure 5.5. If we consider the node for all plausibility checks, we can see that for most check indications, there is no overlap with check indications of other check results in the node, or an overlap in a small number of check results. There are no data records which are marked as check indications in all checks. However, looking at the node including only check results for "Missing", we can see that there exist some missing entries overlapping in nearly all underlying check results.

### 5.2.4 Check Property Coverage

A discrimination by check properties such as the class, target data attributes, or severity levels may either be accomplished via the hierarchical definition of the table rows (Section 5.1), or via dedicated columns. A row-based drill-down provides more details, while a column-based discrimination offers a compact overview of the coverage per check property and table row.

For visualization, the space is split into one partition per specific property, e.g., per severity level (Figure 5.1f). The idea is to represent the distribution of data records with indications over these partitions. Each partition of a cell therefore encodes the relative number of indications over all check indications in the underlying row using the area of a centered rectangle (as in the Indication Distribution columns, see Section 5.2.2). The background again discriminates partitions with and without indications.

### 5.2.5 Check Subset Characteristics

Additional columns characterize the subset of checks per table row in terms of the numbers of checks, targeted data attributes, and involved data attributes. For simplicity, the visual encoding uses bar charts (see, e.g., Figure 5.1b). The scaling ensures that steps of one remain perceptible, while the right part is cropped for bars exceeding the column width. Based on user feedback, these columns also display the numbers as text.

## 5.3 Configuration

The Data Quality Overview offers multiple options for configuring the visualization to specific needs and target user groups (*Requirement 9*).

### 5.3.1 Layout

All columns of the Data Quality Overview can be reordered and resized by drag and drop. Based on user requests, we support two layout modes. Per default, the shown columns fully cover the width of the view and are resized proportionally upon, e.g., the addition of new columns. However, as the view has limited horizontal space, which particularly is an issue if a large number of columns are displayed, the view alternatively supports horizontal scrolling. In this case, a user-defined number of left-hand columns is kept visible to preserve context for scrolled columns.

### 5.3.2 Filtering

**Check Filter** Checks can be *filtered* in terms of check properties. For example, checks may be restricted to those with indications, which is the default, or to certain data attributes, check classes or severity levels. Figure 5.6 shows an example application of a check filter. First, the Data Quality Overview shows all checks which have at least one indication (Figure 5.6a). As the yellow triangle in the first row of the Indication Frequency column indicates the presence of checks of the class "Constraint Violation", the user may want to investigate indications of this class, and therefore applies a filter for this class. Applying the filter reveals all data attributes with constraint violations, where "Temperature 17" stands out in particular (Figure 5.6b). Selecting the indications of the respective row immediately shows the corresponding data records in the linked time series view for further investigation (Figure 5.6c). Hover-triggered buttons support modifications by either *hiding* or *keeping* selected properties, i.e., filtering the selected or all but the selected ones. These buttons also serve as drop targets to which check properties can be dragged from, e.g., the hierarchy or the color legend. Additionally, hovering a class within the color legend applies a temporary filter to the visualization, overlaying bars for that class in the Indication Frequency column, and restricting all other columns to display only the indications of the hovered class. This temporarily reduces the visual complexity, by showing the result of a possible applied filtering.

**Data Record Filter** Another possibility is a global *data record filter*, where all data records are restricted to one or more categories of a categorical data attribute. An example would be to consider only data records of weekends. The filter is applied when computing the plausibility check results, where only the entries matching the defined filter may result in check indications. The definition of a data record filter provides even more flexibility in configuring the Data Quality Overview for the exploration of the plausibility check results.

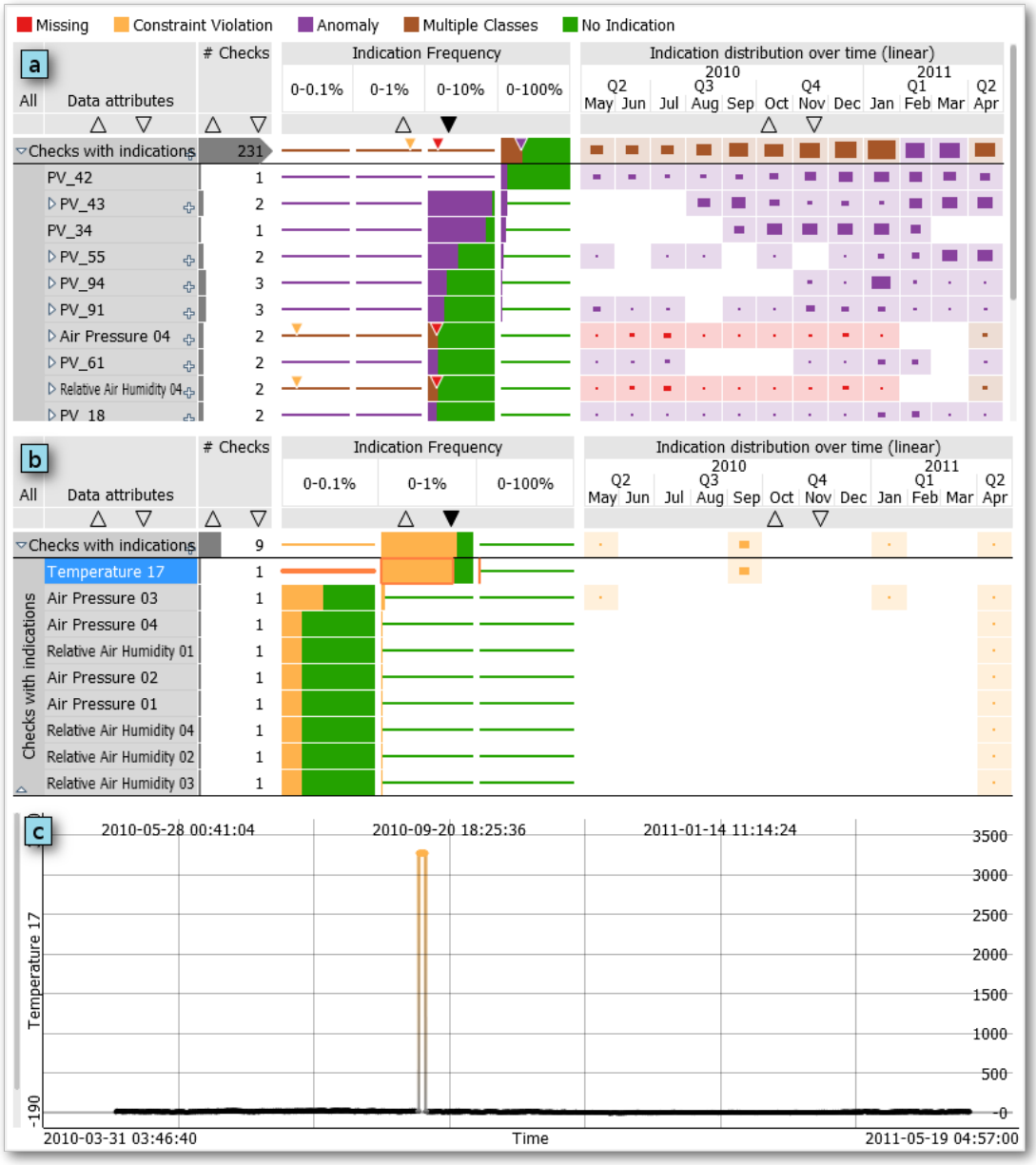


Figure 5.6: Filtering checks in the Data Quality Overview by the class "Constraint Violation". (a) Shows the view containing all checks with indications, in (b) the filter is applied, which reveals "Temperature 17" with most indications. (c) Shows the selected violations in a linked time series view.

### 5.3.3 Hover-Triggered Controls

A frequently used concept in the Data Quality Overview are hover-triggered control elements. In order to reduce the number of simultaneously visible control elements, the controls for a specific part of the visualization are only shown when the respective area is hovered with the mouse cursor. Figure 5.7 shows the concept in context of a temporal Indication Distribution column. Controls are provided for removing (Figure 5.7a), replacing (Figure 5.7b) or adding columns (Figure 5.7c), as well as for the configuration of parameters of the hovered column (Figure 5.7d), which in this example comprises a combo-box for switching between linear and cyclic partitioning as well as a slider [ARP14] for controlling the target range and the granularity of partitions in a defined focal range [LA94]. The controls for configuring parameters of the column usually take a lot of screen space, but don't need to be visible at all times. Therefore, these controls are hidden by default, but can be accessed by the hover-triggered button as seen in Figure 5.7e.
Hover-triggered controls are also provided for each segment in the hierarchy. Hovering such a segment overlays controls for removing the segment or replacing the underlying check property, and expanding or collapsing all visible nodes of the respective hierarchy level.

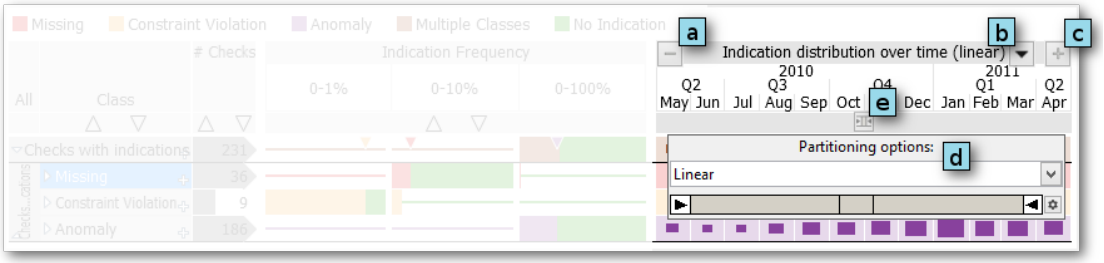


Figure 5.7: Hover-triggered control elements in the context of a temporal Indication Distribution column. Each column provides controls for (a) removing the column, (b) replacing it with another one or (c) adding a new column right of the current column. Some columns may provide controls for parameters of the view, which can be seen in (d) and (e) can be accessed by a button.

### 5.3.4 Sorting

In order to sort table rows, most columns offer upward and downward arrow buttons in their header. Columns in the hierarchy part of the visualization, for example, sort the nodes in the respective hierarchy level in alphabetical order, while columns, which display aspects of check indications, use the underlying value such as the indication frequency as sorting criterion. For partitioned columns (i.e., Indication Distribution, Indication Overlap, and Check Property Coverage), the arrows also serve as drop targets for sorting

rows by particular partitions via drag and drop, e.g., the indication frequency within a particular month. The order of rows still preserves their hierarchical structure, i.e., each node sorts its child nodes separately.

## 5.4 Selection

The Data Quality Overview supports the selection of checks, data attributes, and data records (*Requirement 4*). This is important for integrating the Data Quality Overview on a system- and workflow-level (see Chapter 6) and enables access to details (*Requirement 5*), a validation of indications (*Requirement 6*), and the export for downstream steps (*Requirement 7*).

Clicking on a table row selects all contained checks, including all child-rows, a multi-selection of table rows is possible via common modifier keys. A blue background in the hierarchy representation marks selected table rows. Selecting checks also selects the involved target data attributes, e.g., for assignment to linked views (Chapter 6). Hovering a row temporarily desaturates all but the selected row and its child rows, in order to emphasize the currently hovered plausibility checks.

Most columns support a selection of data records. Within the Indication Frequency column, a click on a bar selects data records with or without indications in the underlying plausibility check results (see Figure 5.6b), while triangles select indications of the corresponding class. For partition-based columns, clicking a partition selects data records with indications in that partition. Hovering any of these trigger areas temporarily highlights the data subset in linked views [BP10].

A global selection mode defines if selected data records replace, extend, refine, or are subtracted from any previous selection. This concept applies to selections within the Data Quality Overview as well as to linked views. See Sections 2.5 and 6.2 for more details of this concept.

CHAPTER 6

# Integration in Visplore

The Data Quality Overview, as described in the previous section, focuses on a scalable summary and selection of check results, which complies with some of the defined requirements in Section 3.3 (*Requirements 1, 2, 3, 4, 8 and 9*), but does not attempt to fully cover the other requirements. Instead, the widely-accepted concept of linked views [WBWK00] (see Section 2.2.3) is used for access to details (*Requirement 5*), as well as a visual validation and identification of checks (*Requirement 6, Requirement 10*).

In order to fulfill the remaining requirements, the implementation of the Data Quality Overview is based on an integration in the visual analytics system Visplore. Section 2.5 describes Visplore in more detail, including available visualization techniques and used concepts. In short, Visplore provides the needed tools to fulfill all defined requirements, which are not covered by the Data Quality Overview itself: data import and export (*Requirement 7*), the management of checks (*Requirement 10*), a multitude of visualization techniques, the support of scripting languages and tools, the definition of statistical models and selections on data attributes and data records.

## 6.1 Linked Views

For the purpose of accessing details of affected data records and their underlying data attributes (*Requirement 5*) as well as the validation of check results (*Requirement 6*), expert users may create and parametrize new linked views of any type at any time. In order to support non-expert users (*Requirement 9*), pre-defined configurations of views have proven necessary for acceptance. In case of energy sensor data, for example, a common configuration includes a time series view and a table for details as seen in Figure 6.1. To minimize the interaction effort, most views support an automated assignment of selected data attributes to visual attributes. In the case of the previously mentioned configuration, the data attributes targeted by a selected set of check results are automatically assigned to the Y axis of the linked time series view, which enables

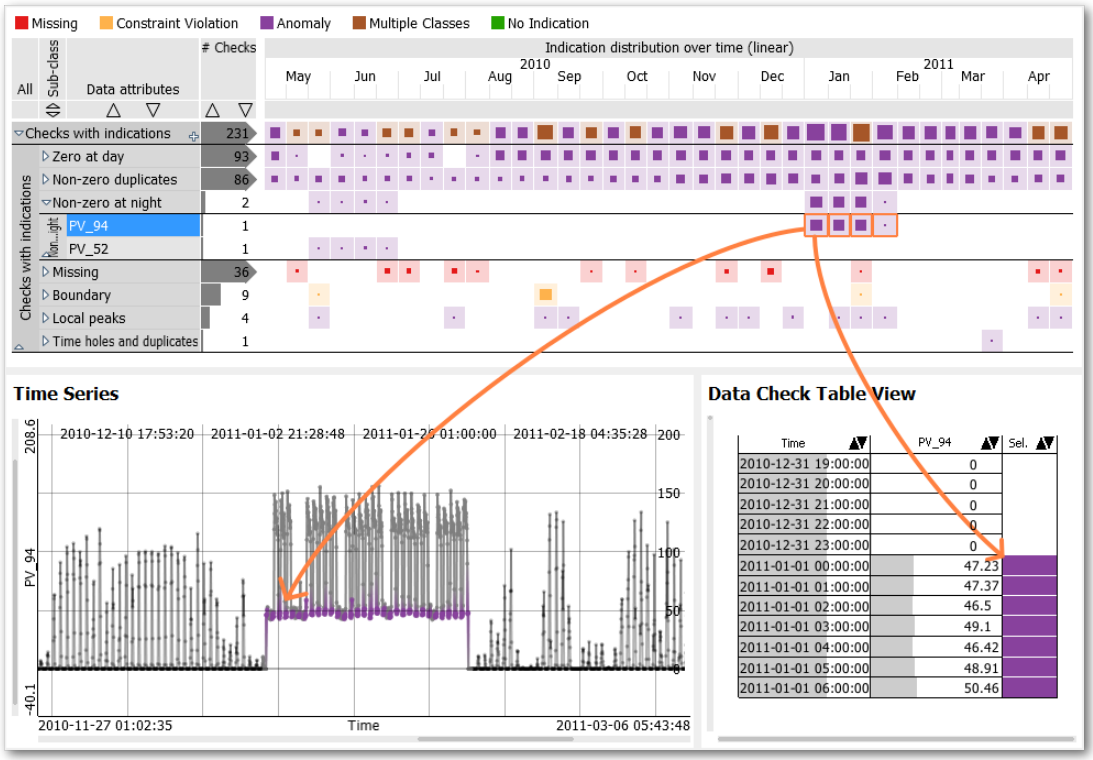


Figure 6.1: The concept of linked views in an example including a data quality view with a linked time-series and data table view. Selecting check indications in the Data Quality Overview highlights the corresponding data records in the linked time-series and shows the corresponding values in the table view. The time-series view is further configured so that the view zooms to the selected records of the time-series.

users to quickly access details of the selected check results. Subsequently, the selection of check indications in the Data Quality Overview immediately highlights those records in the linked time series view (see Section 6.2). The user instantly can investigate whether the selected plausibility check indications pose real data quality problems, or just indicate false positives or false negatives. A configuration including such a time series view may be helpful when assessing the quality of time series data, but it may prove unusable regarding other data types. The variety of views provided by VISPLORE enables configurations for many data types and scenarios, as well as the possibility to look at data quality problems from many different points of view.

## 6.2 Selection

In VISPLORE, the supported Focus+Context system (see Sections 2.2.4 and 2.5) allows for linking views through interactive brushing. Selecting or hovering plausibility check

indications in the Data Quality Overview highlights the corresponding set of data records in the hue of the class of the detected data quality problem in most of the linked views. The Focus+Context system defines layers such as: a temporary subset of hovered data records [BP10], persistently selected data via brushing, and an optional context as an extension of the persistent selection. In case of time series data, the context may contain all data records within a certain time-period of the selected records, such as an hour or a day. It is useful for preserving the context of data quality problems, e.g., when listing affected values in a table, or for automatically zooming to the selected data in the time series view. Figure 6.1 shows an example of this system in combination with the Data Quality Overview. The selected data records with indications are highlighted in the hue of the selected class, in this case purple for "anomaly", while the context of the selection is highlighted in gray, using an increased point size.

## 6.3 Export

The data matrix as defined by the selection of data attributes and data records can be exported (*Requirement 7*) to a data base, a CSV file, or via the clipboard to spreadsheet software such as Excel. A common use-case is the export of the set of data records without plausibility check indications, and is supported by performing an export after selecting the green bar of a row in the indication frequency column (see Use case 3 in Section 7.1).
Visplore also supports exporting views as images, where either a single view or the set of visible views as a whole can be exported. This may be useful for reporting found issues in the data to other users or the management.

## 6.4 Plausibility Check Management

In order to manage the created plausibility checks, a GUI component in Visplore (subsequently called "Check Manager") lists all defined checks (see Figure 6.2). It displays a scrollable table containing all checks (Figure 6.2a), which may be filtered using the text entries above the table (Figure 6.2b). The visible set of checks is reduced to checks containing the specified filter-text in one of its attributes (e.g., the name of the data attribute as in the figure). Selecting one or more checks in the table displays controls for implementation-specific parameters of the selected checks (Figure 6.2c). Independent of the implementation-specific parameters, the controls additionally include means to define user-defined tags (Figure 6.2d), which, in a second step, may be used for an additional hierarchy level in the Data Quality Overview, allowing to partition the available checks by the defined tags. Modifying parameters of the selected plausibility checks immediately triggers a re-computation of the checks, which also updates the visualization. As checks may share the same parameter, the GUI always displays the controls of parameters shared by all checks of the current selection. The selection of checks itself is linked between the Data Quality Overview and the Check Manager, allowing for a selection of checks in the view and immediately providing means to modify their parameters in the respective GUI

component. The Check Manager further provides means to create new checks, and clone or delete existing checks (Figure 6.2e).

In combination with linked views and the Focus+Context system, the plausibility check management system enables the user to identify new plausibility checks as well as to optimize the checks regarding their parameters (*Requirement 10*).

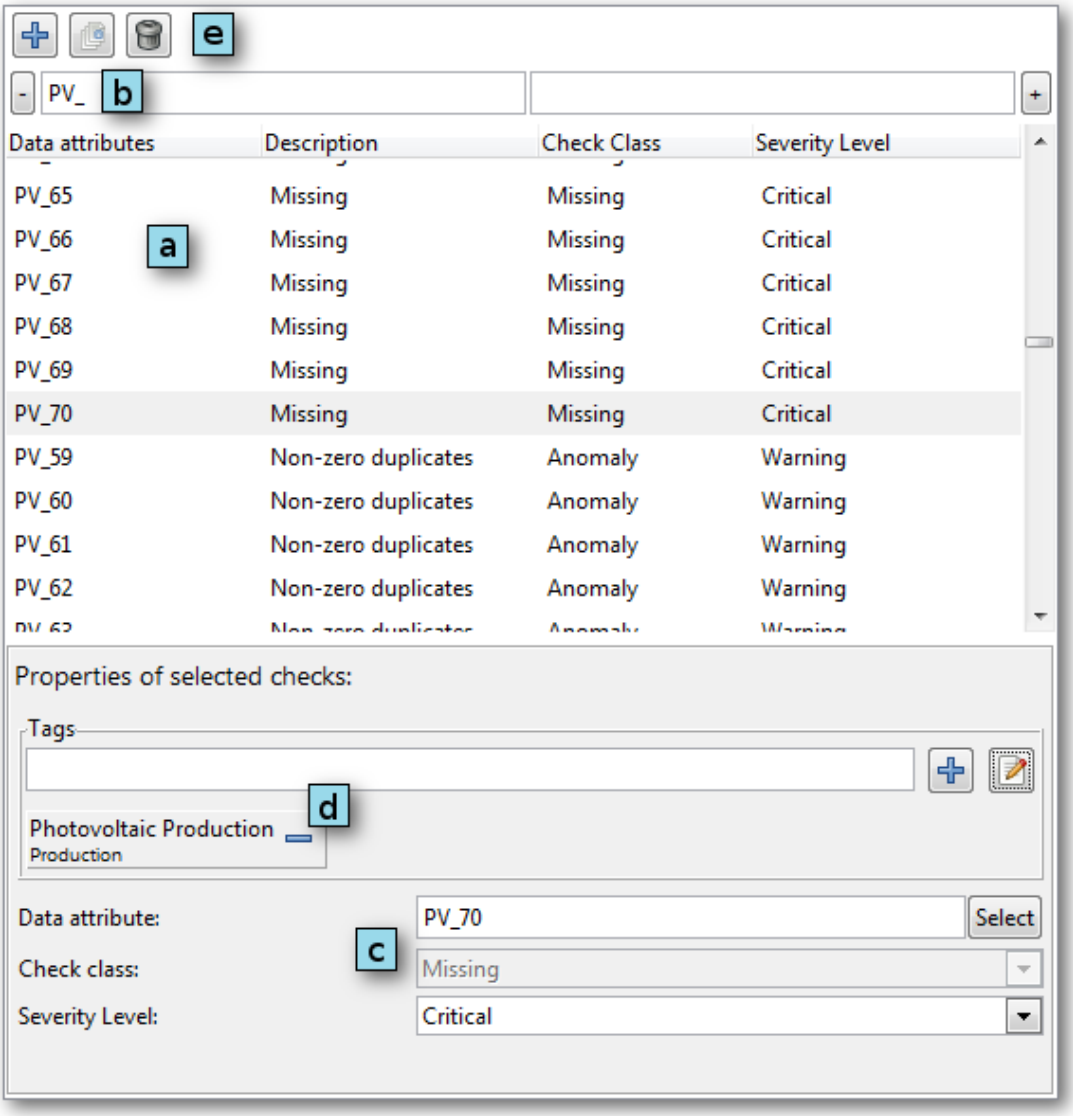


Figure 6.2: The user interface of the Plausibility Check Manager, providing (a) the list of current plausibility checks, (b) means to filter the list using substrings of attributes of the checks, (c) controls for changing parameters for the currently selected checks, (d) control elements for user-defined tags and (e) means to create, clone or delete existing checks.

CHAPTER 7

# Case Study and User Feedback

This chapter contains a set of use cases which are derived from tasks as identified in the Task Analysis (see Section 3.2). The given examples are based on the example photovoltaic production data set (Section 4.1). The chapter further includes a summary of user feedback from deployments at project partners from the energy and healthcare sector.

## 7.1 Case Study

Section 3.2 identified a comprehensive set of tasks for data quality assessment based on insights from collaborations with project partners - during interviews, joint data analysis sessions, and contextual inquiries with domain experts. In this section, the Data Quality Overview is applied to the example data set as described in Section 4.1. All use cases are based on a pre-defined set of checks for detecting *missing values*, *boundary violations*, *duplicates*, *gaps of time stamps*, and various *anomalies*, i.e.,

- successive identical non-zero values,
- local peaks,
- outliers exceeding some standard deviations from the mean,
- and a photovoltaic production of zero at daytime or non-zero at nighttime.

**Use case 1** In the first use case the goal is to assess the quality of a data set for data-driven tasks like modeling or clustering. Figures 5.1 and 5.2 depict the Data Quality Overview for this use case. Initially, the aggregation of all checks with indications shows that $\sim 30\%$ of data records has indications in one or more data attributes (Figure 5.2a). A drill-down by data attributes reveals that four photovoltaic power plants (abbreviated PVs) have anomalies or missing values in more than 4% of their data (Figure 5.2b)

and can be inspected in more detail by further drill-downs (Figure 5.2c). Swapping the hierarchy levels shows zero values at daytime as most frequent problem overall, and non-zero duplicates rank second (Figure 5.2d). An additional column displays the temporal distribution of the indicated data quality problems (Figure 5.1d), where the user is able to determine months with a high concentration of plausibility check indications. It shows that most check indications occur during the colder months of the year (December to February). Additionally, missing values occur in all months except February and March, while most boundary violations occur during September.

As another drill-down, two PVs have non-zero production at nighttime. Selecting the check indications of PV_94 for January, for example (see Figure 6.1), shows the respective data records in the linked time series view. In this case, the user is able to validate the found indications, as they present a set of collective anomalies in the time series. Furthermore, the user is able to view the raw values of selected indications in a linked data table view. Concluding use case 1, the data quality seems sufficient overall, but some parts of the data must be excluded for certain downstream tasks.

**Use case 2** A drill-down shows that the checks for zero photovoltaic production values at daytime reveal indications in roughly 15 of the data records (Figure 7.1a). The goal is now to generate hypotheses about possible causes of the detected data quality problems. The linked "Rank by Feature view" provided by VISPLORE (see Section 2.5 for further details) may reveal if any meteorological quantity can explain this data quality problem. After selecting the indications of all checks of "Zero at day" in the Indication Frequency Column of the overview, the linked view ranks all meteorological data attributes by their relevance for this selection as quantified by mutual information. Gust Speed of weather

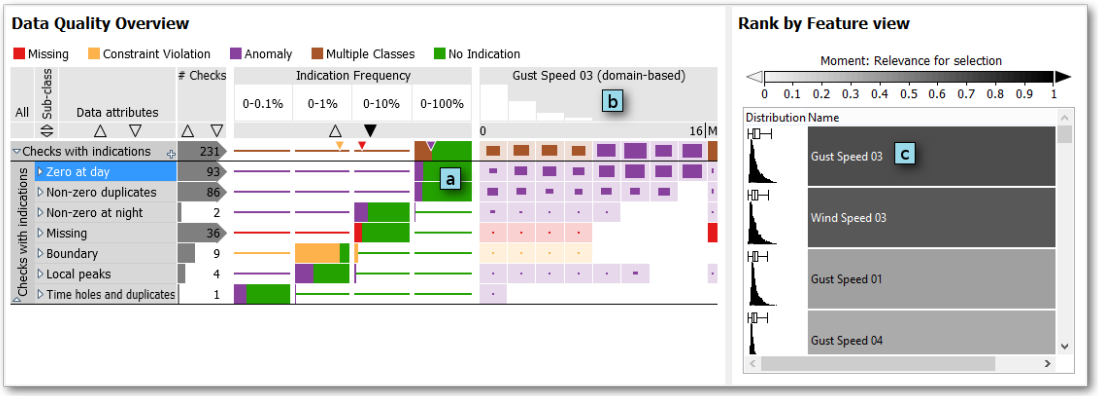


Figure 7.1: *Use case 2*: Selecting detected indications for the sub-class "Zero at day" (a) reveals the gust speed of weather station 3 as possible cause in a linked "Rank by Feature view" (c), which ranks meteorological quantities by their relevance for the current selection of data records. (b) Confirms that the indications mainly occur at times with high gust speed.

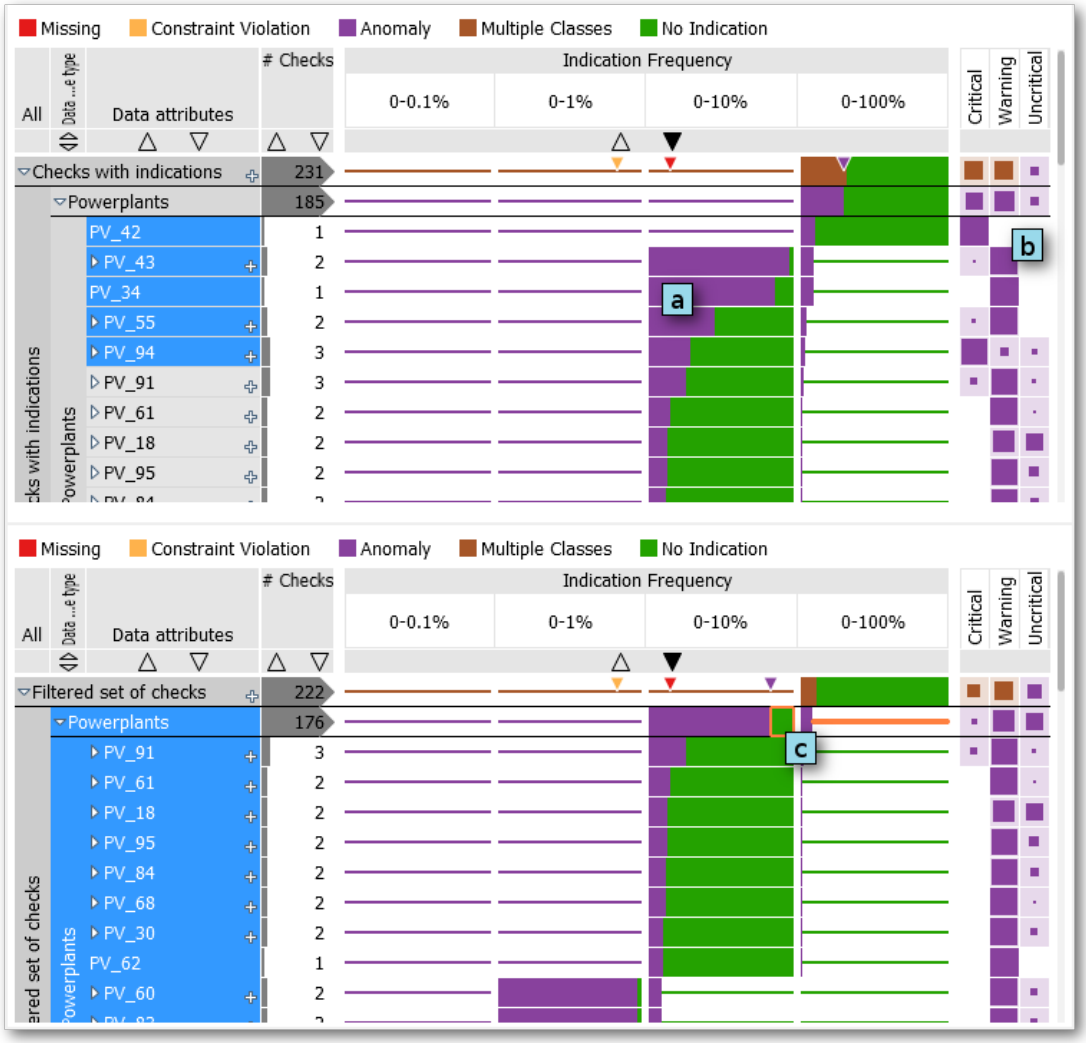


Figure 7.2: *Use case 3*: Excluding photovoltaic power plants with too many data quality problems (a) and high severity (b). The selection of data records without check indications (c) supports a quick definition of clean data subsets for further processing.

station 3 ranks first (Figure 7.1c). Adding it to the Data Quality Overview in the form of a quantitative distribution column, enables the user to have a look at the indication distribution via domain-uniform partitioning (Figure 7.1b). Apparently, zero values at daytime of photovoltaic power production occur more frequently at times of high gust speed. The visualization supports communicating the found problems and their possible causes to the data providers.

**Use case 3** The goal is to select data subsets with a sufficient quality for a subsequent clustering of photovoltaic power plants. A user-defined tag "Powerplants" enables to drill-down on all photovoltaic production data attributes (Figure 7.2). After sorting the table rows by indication frequency, the user is able to select the plants with most check indications, in this case a selection of PVs with indications in roughly more than 3% of their data (Figure 7.2a). Another column of the Data Quality Overview indicates that the severity of most of their data quality problems was classified as warning or critical (Figure 7.2b). The user decides to exclude the selected power plants from further processing steps by applying a plausibility check filter. The clean data subset is then selected by a click on the green bar in the aggregation of the remaining power plants (Figure 7.2c). The resulting data matrix can then be exported, e.g., via the clipboard to Excel.

**Use case 4** A visual exploration of the data attributes as overlaid curves per day reveals an unusual value at 8 pm for global radiation at weather station 1 (Figure 7.3a). Selecting all check indications in the Data Quality Overview confirms that this anomaly is not yet covered by any existing check. The goal is thus to create and parametrize an appropriate check.

The system VISPLORE enables to define a new rule called "Radiation after Sunset" based on a script. For demonstration purposes, we provide an intentionally simple Python script that selects non-zero values after 7 pm for classification by multiple thresholds (Figure 7.3b). Applying this rule to "Global Radiation 01" creates an according check, which appears instantly in the Data Quality Overview, and can be seen in Figure 7.3c. Selecting the initial plausiblility check indications allows for a validation of the indications in the linked view, which reveals many false positives (Figure 7.3d). For check optimization, a property window enables to modify the thresholds (Figure 7.3e), which updates the results in the Data Quality Overview (Figure 7.3f). After identifying an appropriate threshold and validating it again in the linked view (Figure 7.3g), the rule can be applied to the global radiation of all other weather stations. The Data Quality Overview displays similar anomalies for multiple weather stations at multiple days. In the future, the identified checks can be applied to newly acquired data.

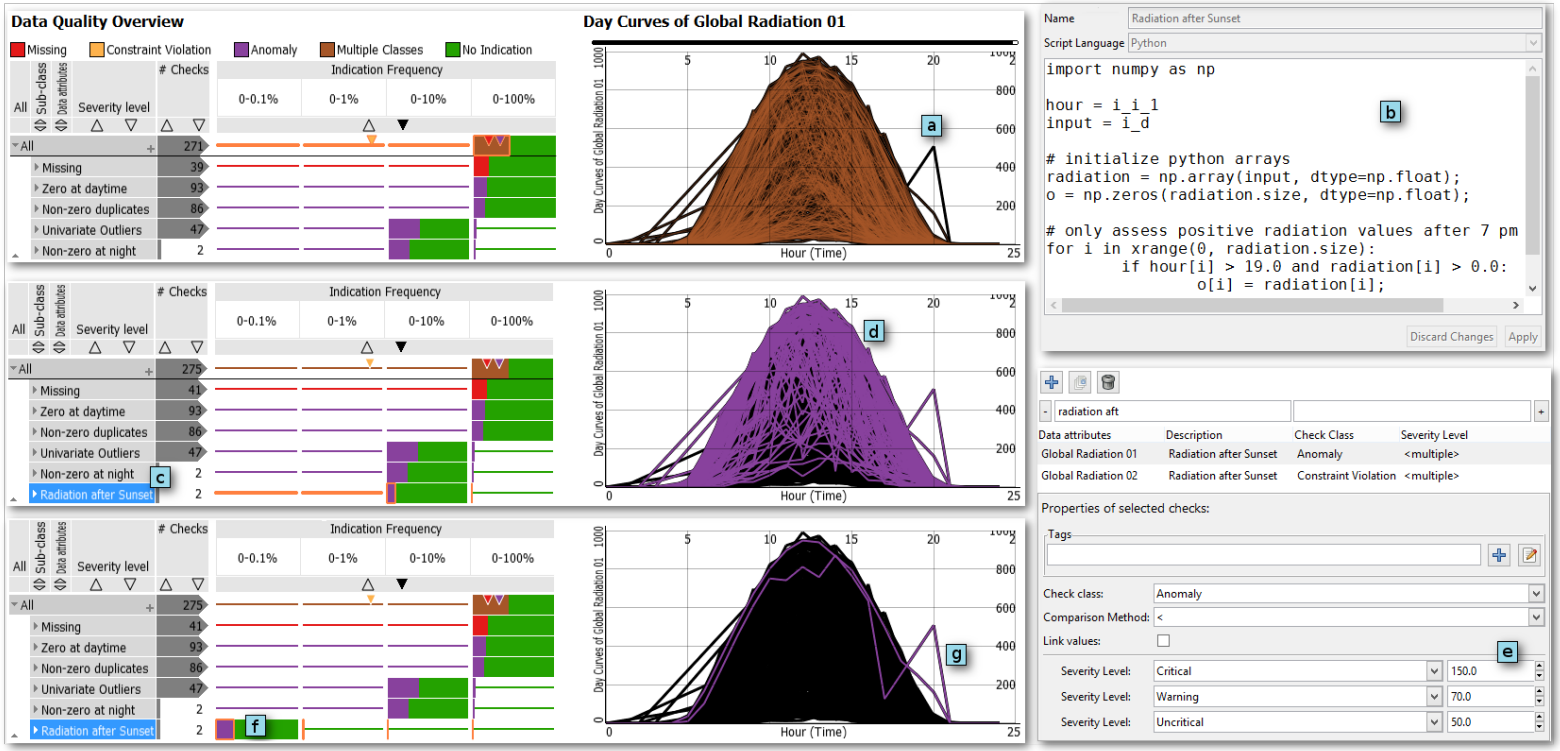


Figure 7.3: *Use case 4*: Parameter tuning of a user-defined plausibility check for global radiation after sunset. (a) Shows an anomaly which is not yet covered by existing checks, (b) allows the user to define a plausibility check based on a python script, (c) shows the newly created check in the overview, which, however, suffers from false positives (d). (e) The Check Manager allows the user to change parameters of the check, which is then updated (f, g).

## 7.2 User Feedback

From the beginning of the development of the Data Quality Overview, domain experts have been involved in the design process for a continuous improvement and validation of the design choices. Additionally, a close-to-final iteration of the view has been deployed for one month to five domain experts from two companies, one being a power grid operator, which provided the example data set (see Section 4.1) and the other one being an IT-solution provider. Currently, the Data Quality Overview is deployed to a company in the industry, in order to monitor quality issues in their data of the production process of their products.

The experts from the energy sector confirmed that data quality assessment is one of their most time-consuming tasks, as observed by Kandel et al. in previous work [KHP+11]. Previously, in order to assess the data quality of a data set, the workflow of the users consisted of inspecting raw data tables or static graphs of single time series, in addition to an inspection of sums of indications by user-defined plausibility checks in tools like Excel. In some cases, violations were detected by formulae similar to our plausibility checks, and for each time series the sum of violations was computed and inspected. However, validating the detected indications was a cumbersome task, as the tools lacked the possibilities to quickly examine single indications in the context of time or other attributes. Additionally, it proved difficult and time-consuming to improve these formulae for a more accurate performance (i.e., to reduce false positives and/or false negatives), as the results of each update of a formula could not be validated in a short time. As a result, in several cases, the high effort for a detailed data quality assessment was not considered to be justified and therefore was only carried out for the most important and critical analyses.

All five domain experts evaluating the deployed close-to-final iteration had previous experience with Visplore. This was essential for the validation, as the users were already familiar with the framework and the Visual Analytics concepts such as multiple coordinated views, therefore reducing distortions of the results by difficulties in using the framework and its basic interactions. After two hours of initial training for the Data Quality Overview, which also included the training for the definition of plausibility checks, the experts used the Data Quality Overview as part of Visplore in their day-to-day workflows, for example, prior to statistical modeling. Using the Data Quality Overview, the involved experts were soon able to perform a thorough data quality assessment within 1/4 to 1/5 of the time previously necessary. After initially defining and optimizing a suitable set of plausibility checks, consulting the Data Quality Overview was fast enough to precede any analysis. This enabled the user to assess the quality of data sets, which previously would not have been inspected in detail due to the amount of time needed for the investigation, ultimately increasing the confidence in the data quality and the following analyses. One expert even pointed out that, for the first time, the Data Quality Overview enabled him an efficient data quality assessment of five million data records and dozens of data attributes.

Even though the experts initially were not familiar with the concept of scale-stacked

bars in the case of the Indication Frequency column (see Section 5.2.1), they quickly saw the advantages of the visualization and claimed that it is easier to interpret as frequently used logarithmic scalings. Overall, the experts consider the visualization as easy to interpret and suitable for presenting data quality problems to the management and other stakeholders. Especially the efficient drill-down possibilities for checks in the hierarchy in combination with columns for displaying the distribution of indications over time, quantitative, or categorical attributes allowed the experts to communicate detected data quality problems to data providers.

The experts particularly liked the selection of data subsets with indications as well as "clean" subsets without indications for downstream processing, which was not as easily possible with their previous tools. Surprisingly, the users soon used the Data Quality Overview not only for inspecting data quality problems, but also for overviews of script-based indicators in contexts outside of data quality assessment. Some experts tuned the selection of representative training data by an interactive sensitivity analysis of thresholds before feeding the resulting subset to a forecasting tool. According to these experts, the tight integration of check modification and result inspection means a speed-up by a factor of ten over previous approaches for this task.

In addition to the deployment of the Data Quality Overview to companies, feedback was also collected from a broader audience by showcasing the Data Quality Overview for three days at "E-World 2015", Europe's premier energy fair ($\approx$ 24,000 visitors). Around 50 experts from 12 companies including managers and technical directors, gave feedback after private demonstrations of 20-30 minutes. During these short demonstrations, many of them could immediately compare the Data Quality Overview with respect to their own needs, and even asked about its commercial availability. Although most of the experts had no prior experience with concepts such as linked views, they claimed to be able to follow our demonstration well. They were fascinated by the drill-down possibilities, however, they also imagined using fixed configurations of the Data Quality Overview as a monitoring tool for everyday use.

CHAPTER 8

# Design Process

The design of the proposed visualization approach was an iterative process. After an initial task analysis (see Chapter 3), we first used parallel prototyping [DGK+12] of hand-drawn sketches, before iteratively refining software prototypes. In meetings at irregular intervals, experts from both energy and healthcare domains were prompted to give positive, negative, and prospective feedback using the rose-bud-thorn method [LUM14]. This refined our understanding of tasks and requirements, and inspired new iterations. The following sections summarize some key iterations from the initial sketch to the final design of the Data Quality Overview, comparing intermediate prototype implementations with the final result by stating drawbacks of rejected implementations and advantages of kept design concepts. At the end a summary of "lessons learned" from the intermediate software prototypes is given.

## 8.1 Initial Sketch

The hand-drawn sketch as seen in Figure 8.1 served as starting point for the implementation of the Data Quality Overview. The final implementation still resembles the early design concept, however, multiple concepts proved as impractical during development or are considered as future work of this diploma thesis (see Chapter 10).

### 8.1.1 Hierarchy

The initial design of the Data Quality Overview already structured the set of plausibility check results in a hierarchy, similar to an icicle plot [MR10], where only the leaf-nodes of the hierarchy are displayed (see Figure 8.1). This concept, in which the root node and intermediate nodes are replaced upon expansion, was, however, rejected after the implementation and evaluation of a first prototype, as it turned out to have several drawbacks (see Section 8.2.1).
Additionally, the initial design sketch includes first ideas for columns to define hierarchy

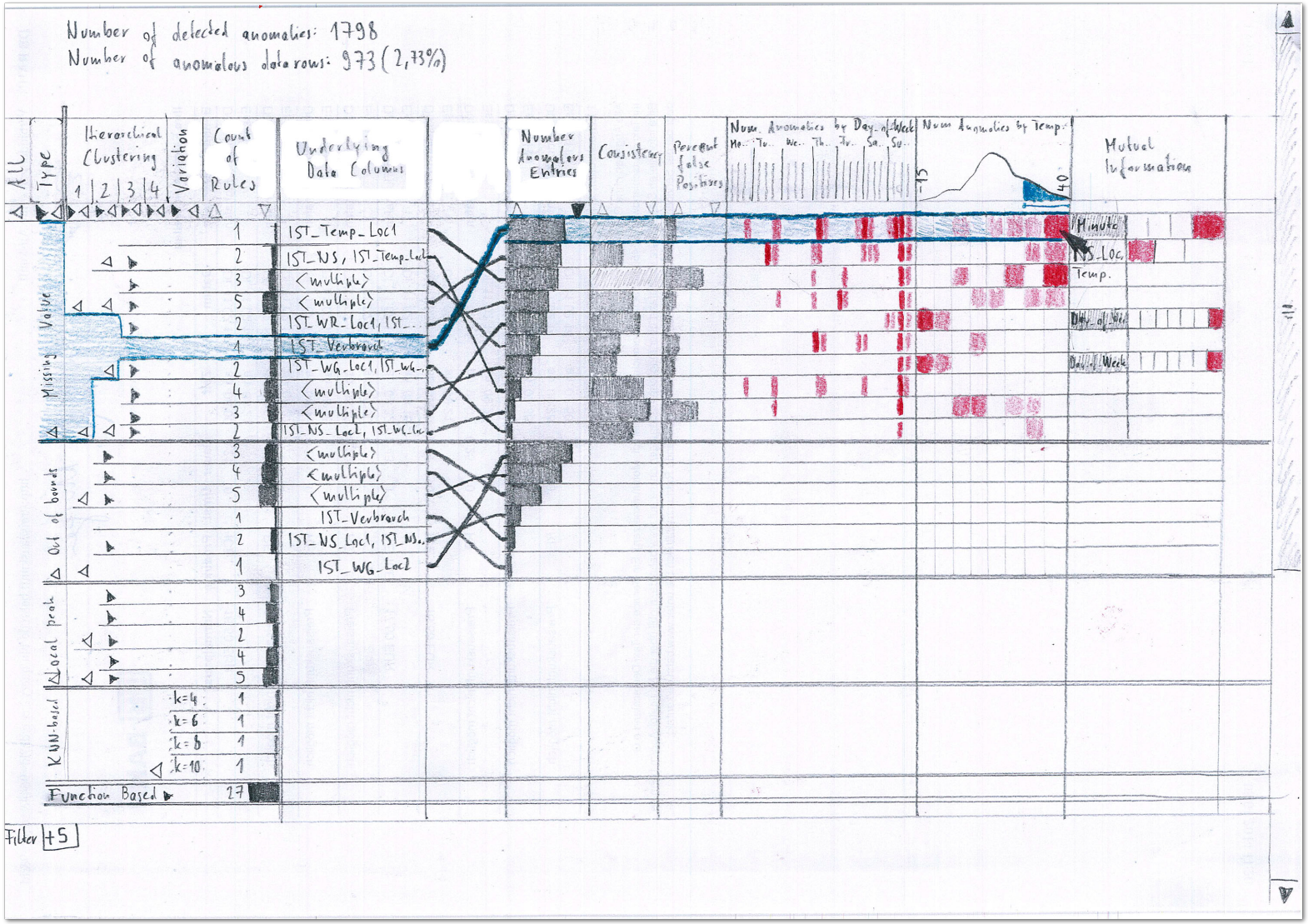


Figure 8.1: Initial hand-drawn sketch of the design of the Data Quality Overview [Pir14].

levels, such as “Type”, “Hierarchical Clustering” and “Variation”. The “Type” column was designed to partition check results depending on the type of the detected data quality problem. The idea of a “Hierarchical Clustering” column was to group check-results based on similar indication patterns, in order to find checks whose results may correlate. The “Variation” column was intended to discriminate between check results only differing in the value of a check parameter, such as a threshold or, as in the example of the sketch, the number of neighbors of a K-nearest-neighbor-based plausibility check.

### 8.1.2 Columns: Aspects of Check Indications

The initial sketch contained a comprehensive set of columns to provide information about check indications. This includes columns to display properties of each row such as the number of underlying check results and data columns, but also columns to indicate the frequency of indications, their consistency or distribution.

**Number of Anomalous Entries** In order to provide information about the indication frequency for the set of plausibility check results in each row, a column named “Number of Anomalous Entries” was contemplated. For each row, the column was designed to show a bar indicating the percentage of detected data quality problems.

**Consistency** The “Consistency” column was intended to give the user an idea of how consistent detected data quality problems are over all checks of a row. Again, for each row, the column shows a bar, indicating the percent of detected entries which are consistent in the underlying check results.

**Percent false Positives** This column, provided that the underlying data set has information about the validity of each data record, gives information about the presence of data records with indications which in reality are valid values (*false positives*). Similarly, another column was designed to provide information about *false negatives*, which are data records without indications in one of the underlying plausibility checks, but in reality would be invalid. This information can be useful to optimize plausibility checks in order to minimize an incorrect classification of data records. Currently, pre-defined labels (Section 2.3.2) are not yet supported by the Data Quality Overview, and therefore this column remains future work.

**Indication distribution** The initial design sketch also contemplated columns to provide information about the distribution of data records with indications. Looking at Figure 8.1, two columns were designed for this task. The column “Number of Anomalies by Day of Week” was designed to display the temporal distribution of indicated data quality problems, by partitioning the underlying temporal space into units such as years, months, or in this case the day of the week. In order to display the maximum detail, each partition ultimately covers the range defined by one pixel. For each of the partitions, the percent of data records with indication is encoded with a colored quad, using transparency

to indicate the percentage. The column "Num. Anomalies by Temp." is an example for displaying the distribution of indications with respect to a given data attribute (in this case temperature). The domain space of the underlying data attribute is divided into equally-sized partitions, similar to the temporal distribution column, also using the same visual encoding.

**Mutual Information** The goal of the "Mutual Information" column was to suggest data attributes which correlate to the data indications of the underlying checks of a row in one way or another. The idea is similar to the concept of mutual information introduced by Kandel et al. in *Profiler* [KPP+12] (see Section 2.4.3), where a set of visualizations are automatically suggested which may explain a data quality problem at hand.

### 8.1.3 Column Group Separators

The separators visible in the initial design sketch were intended to provide further flexibility in the configuration of the visualization. Using this concept, the user is provided with the possibility to split the table into multiple parts (*column groups*), where each corresponding row is connected by a line (*column group separators*). The idea behind such a scheme is to enable to sort each part independently by one of the contained columns, enabling a comparison of the ranks of rows in different columns. However, this feature was later rejected, as users considered it overly complex in the application context.

## 8.2 First Prototype

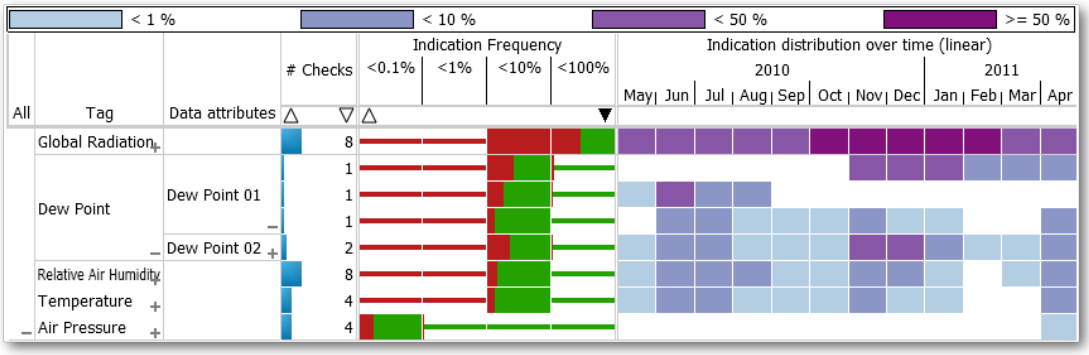


Figure 8.2: A first prototype of the Data Quality Overview.

### 8.2.1 Hierarchy

The implementation of a first prototype of the Data Quality Overview included the hierarchy as designed in the initial sketch (see Figure 8.2). However, own evaluations of this prototype as well as user feedback allowed us to identify several drawbacks of this concept of replacing nodes upon expansion, ultimately resulting in our decision to abandon it. First of all, the replacement of intermediate nodes resulted in a loss of overview of the overall data quality, as the root node, containing a summary of the results of all checks, was replaced in order to drill down to additional hierarchy levels or single checks. This information is important and should to be visible at all times. It is defined as the first requirement of our system (*Requirement 1*). Secondly, hiding intermediate nodes upon expansion made it difficult to compare visible nodes with their parent and sibling nodes. This replacement of information violates the "Rule of Complementary" [WBWK00], which states that a visual comparison is easier to accomplish than memory-based comparison (see Section 2.2.3). User feedback also suggested that expanding nodes instead of replacing them was more familiar to users, as it is a frequently used concept, for instance in graphical user interfaces of file systems. With the first prototype, the possibility to create user-defined *tags* to a plausibility check was added, in order to be able to compose user-defined groups of checks. Subsequently, the hierarchy could now partition the set of check results by their tags. This provided even more flexibility in defining the hierarchy of check results, in order to configure the visualization for specific tasks.

### 8.2.2 Indication Frequency

The Indication Frequency Column (see Section 5.2.1) is one of the central components of the visualization, as it is used to fulfill one of the main requirements of our approach (*Requirement 1*). The appearance of the column underwent many iterations, caused by difficulties arising due to the varying scale of percentages, as well as design decisions such as a global encoding of the semantic class with hue, which immediately affected the design of this column.
Initially, each cell displayed a simple bar to show the percentage of data records with indications. Even if this approach was simple and easy to read, it proved impractical when dealing with very small values. To deal with this issue, the concept of scale-stacked bar charts [HSBW13] was adopted (see Section 5.2.1 for more details).
After user feedback suggested that the semantic class of data problems was important information to be displayed at all times, we decided to encode the semantic class as hue in the entire visualization. This decision posed a challenge for the design of the Indication Frequency Column. In a first iteration, we simply stacked the bars of each check class to encode this additional information, however, it was possible that small percentage values could not be perceived.

### 8.2.3 Indication Distribution

In the first prototype, each partition of an Indication Distribution Column displayed the percentage of data records with indications encoded by hue, using a different color for defined magnitudes of the percentage. Figure 8.2 shows this initial encoding, which has several drawbacks. One of the main drawbacks of the approach was that differences in the percentage values were only comparable in the specified orders of magnitudes such as $< 1\%$, $< 10\%$, $< 50\%$ and $>= 50\%$. The idea behind such a logarithmic coloring scheme was to ensure the visibility of small percentage values, as well as, a perception of higher orders of magnitude. However, the differences between the magnitudes were not comprehensible without a second glance at the color legend. Additionally, after we decided to encode the check class by hue in all parts of the visualization, we finally decided to switch to another encoding, as we did not want to overuse color as visual attribute. Encoding the percentages using glyphs of different sizes turned out to be the attribute to go for. The percentage values were directly mapped to this attribute, allowing for an easy comparison of the values – horizontally and vertically (see Section 5.2.2).

## 8.3 Final Iterations

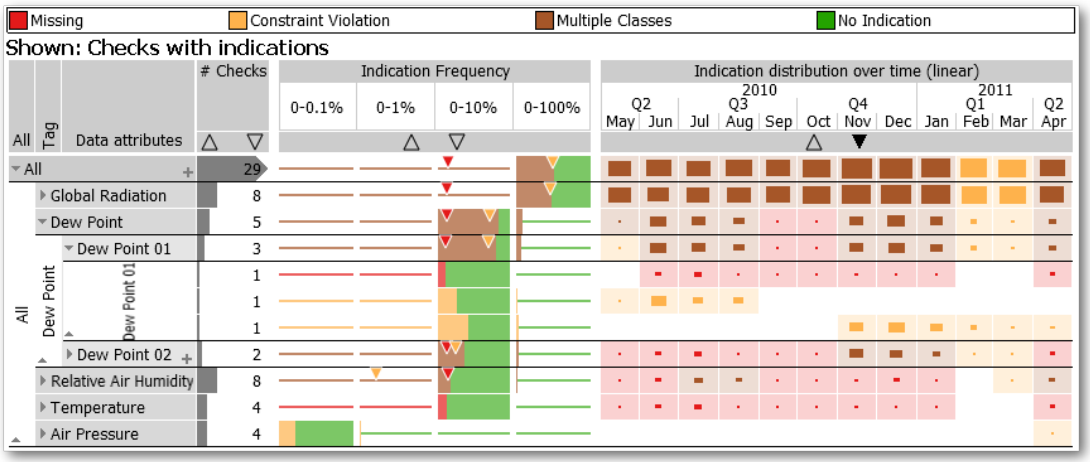


Figure 8.3: Close-to-final Prototype of the Data Quality Overview.

### 8.3.1 Indication Frequency

Using stacked bars to encode the frequency of check indications for each class again introduced the problem of the visibility of small percentage values, especially in the $0 - 100\%$ sub-column, as the perception of each class could not be guaranteed when multiple classes had smaller orders of magnitudes than their sum. The solution was the introduction of a color for "multiple classes", while a glyph was added to display the

percentage of indications for each class individually. Figure 8.4 shows several intermediate iterations to encode the percentage of data records with their semantic class as hue.

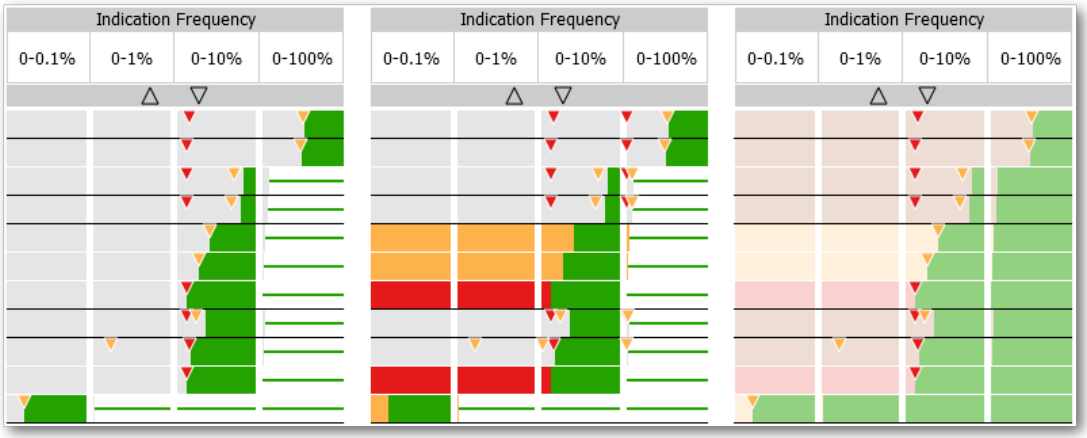


Figure 8.4: Alternate implementations of the Indication Frequency column in later stages of the design process.

### 8.3.2 Indication Distribution

Figure 8.3 already shows the final implementation of the Indication Distribution column. The percentage values of each row are directly mapped to the size of each glyph, which makes it possible to compare each glyph horizontally and vertically.

### 8.3.3 Indication Overlap

The sketch in Figure 8.1 contemplated the representation of the Indication Overlap (Section 5.2.3) as a simple column with a bar for each cell, displaying the consistency of the indications of plausibility checks inside of a row. However, such an approach has several disadvantages. By using a bar for each cell it was only possible to encode the number of data indications overlapping in all checks, without further information whether there were data entries with partial or no overlap in other checks. Additionally, the approach limited the possibilities in selecting the data entries depending on the amount of overlap between the respective plausibility checks. Having this in mind, the column was first changed to display the percentage of data indications with no, partial, and full overlap in the underlying checks of a row. However, this approach proved to be insufficient especially in a cell with a large number of plausibility checks, as there was no distinction between indications overlapping in all but one checks, and indications overlapping in only two checks. Therefore, we decided to partition each cell depending on the number of plausibility checks of the row. Section 5.2.3 describes this final representation in more detail.

## 8.4 Lessons Learned

In this section, some lessons learned from mistakes and user feedback during the design of the Data Quality Overview are summarized. Even if the following findings are only based on reflections on the design process of the proposed approach, some aspects also may be valid in a more general context.

**Confusion of "overview first" with "overwhelm first".** In order to convey much information at one glance, initial designs of the Data Quality Overview already showed a pre-defined hierarchy along with multiple columns at startup. The resulting visual complexity asked too much from most users who had only been familiar with simple diagrams before. Having a simple and highly aggregated starting point (Figure 5.2a) and providing further information only at explicit drill-down stages, received a much better acceptance, e.g., at the demonstrations at the "E-World 2015" event.

**Expand rather than replace at drill-down.** Initially, the table rows were hierarchically structured similar to an icicle plot [MR10], i.e., intermediate nodes had no representations as rows. The users preferred the indented layout as the context in the form of the parent node, which was not lost when expanding a node. Additionally, such a layout is more familiar to well-known concepts such as the folder hierarchy in file browsers. Also, the indented layout required less space than the version similar to an icicle plot.

**Meaningful partitions rather than maximal detail.** For indication distribution columns, initial designs defined partitions as the range covered by one pixel for maximal detail. However, users preferred meaningful partitions instead of maximal detail, allowing for an easier implementation (for example months or days for temporal partitioning and steps of 10 for quantitative partitioning).

**Increased simplicity by a dedicated "multiple class" category.** At first, the scale-stacked bars of the Indication Frequency column was subdivided into one segment for each class. However, this suffered from multiple problems. For example, the perception of each class could not be guaranteed when multiple classes had smaller orders of magnitudes than their sum, and users where overwhelmed by the complexity of the visualization. Using a dedicated category for "multiple class" and relying on a drill-down for disambiguation was seen as the better option by our users.

**Integration as important as visualization.** Despite the positive feedback on the Data Quality Overview, most users stressed that it would mean much less benefit for them without the possibility to define and to re-use own checks in scripting languages such as Python.

CHAPTER 9

# System Architecture and Implementation

The Data Quality Overview was implemented as a plugin into the existing framework VISPLORE (see Section 2.5), which is implemented in C++ and uses OpenGL for rendering. The multithreading architecture of VISPLORE [PTMB09] provides the means to ensure responsiveness of the visualizations implemented in the system, allowing for immediate visual feedback to interactions. The plausibility checks and their results are seamlessly integrated into the multivariate data model of VISPLORE.

## 9.1 Plausibility Check Results

In order to support an efficient hierarchical aggregation of plausibility check results (see Sections 2.2.1 and 4.3), each result in the Data Quality Overview is represented as a *binary mask*, for which the implementation is already provided by the data model of VISPLORE. Each mask simply marks the indications of a check result for each entry of the underlying data attribute(s). VISPLORE supports an efficient combination of such masks, which is frequently needed when aggregating check results and subsequently computing measures for columns in the overview.

When aggregating the check results for a node in the hierarchy (see Section 4.3.1), different combinations of masks have to be computed, depending on the classes of the underlying check results and the used aggregation level. Most columns of the Data Quality Overview display information about the class of data quality problems. In the example of the Indication Frequency column (see Section 5.2.1), the relative frequency of check indications for each class is displayed. Therefore, an aggregation of check results of each class is needed, i.e., a combination of binary masks per check class.

When using record-based aggregation, the binary masks of the check results can be combined with a simple binary OR-combination, resulting in a single binary mask per

check class. Using value-based aggregation, however, is more expensive, as we need to combine the check results for each underlying target data attribute, before being able to compute measures such as the relative frequency of indications (see Section 4.3 for more information).

## 9.2 Multithreading

In addition to the main thread, handling user inputs, visualizations in VISPLORE use an own *visualization thread* for updating their data model and rendering, in order to ensure responsiveness of the system and to provide feedback as much and as soon as possible. Each time data records are selected or parameters such as the point size of a 2D scatterplot are changed, the thread of the affected visualization is restarted, updating the data model and rendering the components of the visualization according to the applied changes.

In order to make full use of the architecture of VISPLORE, the Data Quality Overview maintains multiple threads to update the visualization. A schematic representation can be seen in Figure 9.1. The Data Quality Overview frequently needs to do expensive computations, for example when creating and adding new plausibility checks, changing parameters of existing checks, or expanding nodes in the hierarchy, where the aggregations for subnodes need to be computed. In this case, using one thread for the visualization would not guarantee responsiveness at all times, as an interaction with the visualization would not easily be possible during expensive computations. This led us to the design decision to use two threads, one for updating the data model of the Data Quality Overview, and one for the rendering of the visualization.

### 9.2.1 The Data Model Thread

As previously mentioned, the main reason, the *data model thread* was introduced, was to reduce the load of the visualization thread. It computes all needed costly computations, while the overview still allows to be responsive for interactions of the user. Each time the data model thread finishes its computation, the computed state is forwarded to the visualization thread, which then redraws the relevant parts of the Data Quality Overview. During computation of the data model, the visualization thread still renders the last valid state. The architecture would also allow for an animated transition from the old to the new state, which is however still future work. Currently, in some cases, the visualization simply indicates ongoing expensive computations by providing text messages. An example would be when initially computing the results of all existing checks, which usually takes some time, depending on the number and complexity of the plausibility checks. In this case, a simple text consisting of the name of the currently computed check result along with the progress in percent of overall checks is shown.

After computing all plausibility check results, the hierarchy of the Data Quality Overview needs to be updated. Nodes of the hierarchy are created, deleted or updated, depending on changes in check results or in the hierarchy structure. Changes in the

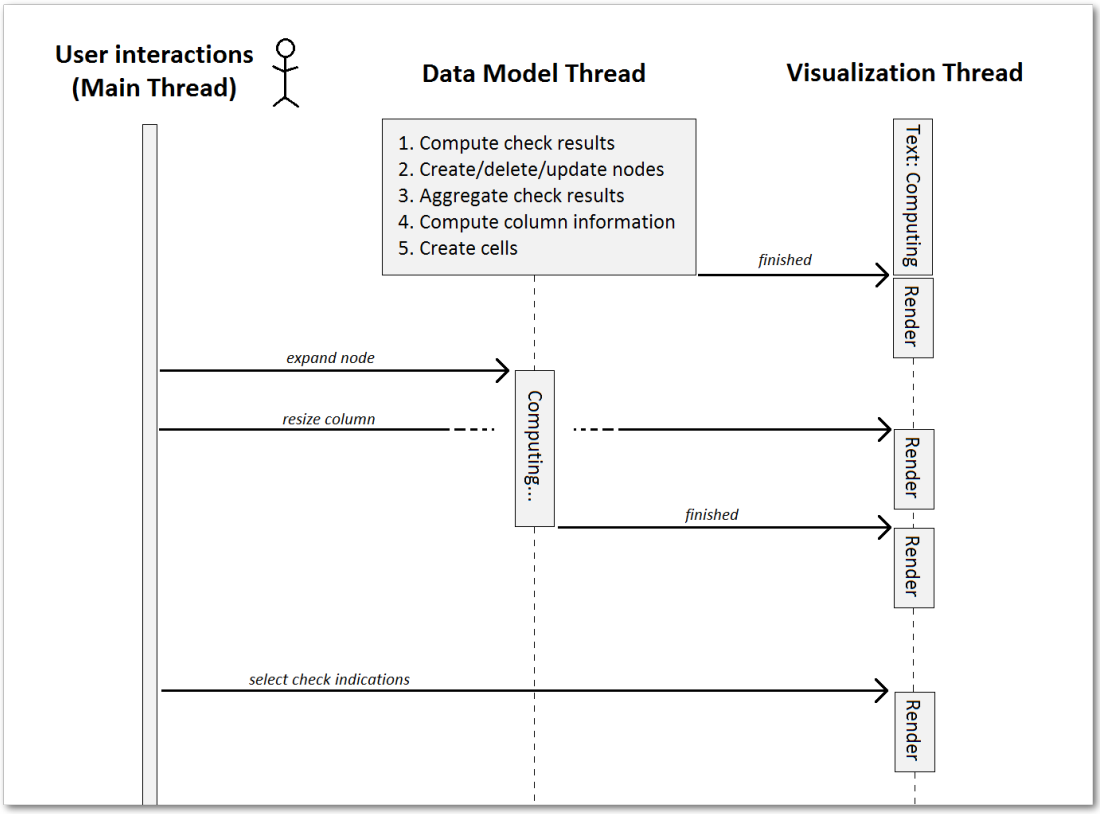


Figure 9.1: Schematic representation of the multithreading architecture of the Data Quality Overview.

hierarchy structure include expanding or collapsing nodes, or adding or removing hierarchy levels. In some cases, the definition of nodes of a hierarchy level may depend on the results of underlying plausibility checks. An example would be a hierarchy level which creates nodes for results with the same or similar indication patterns. An optional user-defined check result filter (Section 2.2.2) is also applied in this stage, and needs to be considered when creating nodes of a hierarchy level.

As a next step, the aggregated binary masks for each node are updated. Depending on the column configuration, different types of check result aggregations are needed, such as aggregations per check class or per severity level, in order to provide the needed information for a column to compute its measures and create its cells.

However, in addition to the binary masks provided by each node, some columns may need additional information before finally being able to create a cell for each node in the hierarchy. This includes simple information such as the number of underlying checks or data attributes of a node, or more complex computations such as the creation of partitions for a temporal partitioning column (see Section 5.2.2). In the case of such a column, the partitioning of the column itself is first computed, depending on the type

of partitioning and considering modifications using the provided hover-triggered slider (Figure 5.7). For each partition a binary mask is created, marking all records of the data table which belong to the respective partition. In order to retrieve the check indications of a node in a given partition, we use a simple binary AND-combination of the aggregated node mask and the partition mask.

### 9.2.2 The Visualization Thread

The *visualization thread* uses the state computed by the data model thread to render the Data Quality Overview. This includes drawing the hierarchy nodes, column headers and cell contents of each column, but also the highlighting of selected or hovered nodes, as well as selection rectangles created by the user. This thread is limited to these tasks in order to allow the user to interact with the visualization, even if there are still some ongoing computations in the background.

In the example as seen in Figure 9.1, when creating the Data Quality Overview, the data model thread first has to compute the initial state. During this initial computation, the visualization thread only displays a text showing the current computation process. After the computed state is forwarded to the visualization thread, it renders the corresponding information, and updates it accordingly every time it receives a new state from the data model thread. In the example depicted in Figure 9.1, the first action the user does is to expand the "All" node, which triggers an update in the data model thread. During this computation, the visualization thread still displays the last complete state, and the user is immediately able to do further interactions, and receives feedback from the visualization. After expanding the node, the user resizes one of the columns of the Data Quality Overview, without the need to restart the ongoing computation. The visualization thread updates the visualization accordingly. After the data model thread has finished its computation, the new state is again forwarded and rendered. Afterwards, the user selects a set of check indications in the visualization, and the selection is highlighted in the visualization by the respective thread.

CHAPTER 10

# Reflection and Future Work

Literature and own experience confirmed that the assessment of data quality is an important topic with a high practical relevance. Dasu et al. [DJ03] state that data quality may account for up to 80% of the time and cost in data warehousing projects. Besides, data quality assessment is not just a one-time task, but is a recurring task in most operational settings. In visualization research, dirty and ill-formatted data is still an "elephant in the room", and Kandel et al. [KHP+11] describe that most visualization research simply assumes that the data arrives in a clean state. While research repeatedly mentions visualization as suitable for the detection of data quality problems, not much previous work has been done regarding the routine assessment of data quality problems and their validation for regularly acquired data. The Data Quality Overview, as proposed in this diploma thesis, fills this gap by supporting a variety of user tasks for visual data quality assessment (Section 3.2).

While the visual encoding of the Data Quality Overview is considered as the main contribution of this diploma thesis, the design of the concept complies with the Visual Analytics Mantra as defined by Keim et al. [KMS+08]: The evaluation of plausibility checks enables to *analyze first*, visual summaries of check indications allow users to *show the important*, while the hierarchy and a various set of columns support to *zoom and filter*. Providing means to modify parameters of checks combined with immediate visual feedback enables to *analyze further*. Multiple coordinated views in combination with selecting plausibility check indications allow users to access *details on demand*.

## 10.1 Abstraction

User feedback by domain experts indicated a potential of the Data Quality Overview outside the context of data quality assessment. In fact, the data abstraction of plausibility check results (see Chapter 4) is general and not limited to data quality problems. On an abstract level, the Data Quality Overview provides an aggregated visualization of many

indicators. It relies on meta-information about these indicators like a classification, while specific reasons of indications are abstracted from the visualization. In the context of linked views, the Data Quality Overview thus supports *guidance* to data subsets which are of some interest.

This generalization opens many potential fields of applications. For example, experts from a partner in the healthcare sector suggested to use the Data Quality Overview in the context of business intelligence as an overview of key performance indicators (KPIs). On the other hand, experts in automotive engineering with whom we have collaborated for years [PBK10] imagined the Data Quality Overview to be used in the context of multi-objective optimization as an effective overview of violations of user-defined constraints based on parameter studies. In this case, columns showing the quantitative indication distribution would allow to evaluate the distribution of constraint violations over parameter variations. The investigation of such applications outside the context of data quality will be an important topic for future work.

## 10.2 Scalability

Using a hierarchy for structuring plausibility check results ensures the scalability for a possibly large number of checks and underlying data attributes. The flexible definition of the hierarchy by grouping check results, for example by problem types or data attributes, allows the user to drill down to results important to the current task at hand. However, this could still result in a rather large number of table rows, where the implementation currently allows vertical scrolling of the table when exceeding the available screen space. Even if scrolling is a familiar concept, where feedback revealed that users did not even consider this as a drawback, it violates the *rule of complementarity* as specified by Baldonado et al. [WBWK00], which states that a visual comparison is easier to accomplish than a memory-based comparison. For future work, an implementation based on row-wise focus and context is intended, similar to a table lens visualization [RC94].
Regarding the scalability to a large number of data records, partition-based columns avoid clutter in the visualization, while preserving the distribution of plausibility check indications. Showing the overall frequency of check indications using scale-stacked bars (Section 5.2.1) allows the visualization to show relative numbers of indications of varying scales, and therefore not to lose track of single indications in a row even if the data set has a large number of data records.
As the execution of complex anomaly detection algorithms can be very expensive, the multi-threading implementation of the Data Quality Overview ensures that the visualization remains responsible at all times [PTMB09] (see Section 9.2). However, future work includes the integration of displaying intermediate results in order to support fast visual feedback [MPG+14], for example, when tuning parameters of plausibility checks. Additionally, animated transitions for interactions, like drill-down and filtering, could help to communicate the progress of intermediate results in an understandable way to the user.

## 10.3 Granularity Levels

The Data Quality Overview currently shows check indications only for single data values and data records. In some cases, other granularity levels may be needed, for example when checking for implausible daily patterns in the data. In this case, check indications may be specified in terms of days instead of the granularity level of the records, which may be hours or minutes. Future extensions may include a support to discriminate such granularity levels in the hierarchical structure.

## 10.4 Data Cleansing

Another big topic for future work is the support of data cleansing. Allowing to impute detected check indications to plausible values, such as interpolating missing values, is a logical next step to the data quality assessment. One of the main advantages of the abstraction of plausibility checks (Section 10.1) is that resulting information about data modifications could easily be integrated in the Data Quality Overview. For example, an imputation of data records may add an additional check class (see Section 4.2.1) to the classification of plausibility checks called “Imputed”, and could seamlessly be integrated in the visualization. Additionally, the cleansing of data values could provide a ground truth to be used when evaluating plausibility checks, and further be used to include information about false positives and false negatives of an applied check.

CHAPTER 11

# Conclusion

This diploma thesis describes the Data Quality Overview, a new visualization approach, which addresses a routine data quality assessment based on automated plausibility checks. The overview provides a scalable summary of plausibility check results, with extensive drill-down possibilities regarding checks, data attributes, and data records. The thesis includes a task and requirement analysis based on collaborations with industry partners. It is shown how the Data Quality Overview and its integration within the comprehensive Visual Analytics framework Visplore can support the identified tasks. In addition to the explanation of the visualization and the underlying data model, the thesis also includes a reflection of the design process in order to provide insight into the development process of the thesis, as well as some implementation details.

Additionally to fulfilling the requirements regarding the context of data quality assessment, user feedback suggested an application of the implemented overview also outside of this context for other kinds of indicators. Such an application is left to future work. Another topic for future work is the integration of data cleansing, which was out of scope for this diploma thesis. Motivated by the positive user feedback, the Data Quality Overview can be an important approach for increasing the confidence in the data by enabling an efficient data quality assessment.

# Bibliography


[AMM+07] Wolfgang Aigner, Silvia Miksch, Wolfgang Müller, Heidrun Schumann, and Christian Tominski. Visualizing time-oriented data - a systematic view. *Computers & Graphics*, 31(3):401–409, 2007.

[ARP14] Clemens Arbesser, Oliver Rafelsberger, and Harald Piringer. The Focus-Filter Widget: A Versatile Control for Defining Spatial Focus + Context in 1D. In *Poster Proceedings of IEEE InfoVis*, 2014.

[AS94] Christopher Ahlberg and Ben Shneiderman. Visual information seeking: tight coupling of dynamic query filters with starfield displays. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 313–317. ACM, 1994.

[BB99] Benjamin B Bederson and Angela Boltman. Does animation help users build mental maps of spatial information? In *Proceedings of IEEE Symposium on Information Visualization, 1999 (InfoVis 1999)*, pages 28–35. IEEE, 1999.

[BC87] Richard A Becker and William S Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, 1987.

[BI90] David V Beard and John Q Walker II. Navigational techniques to improve the display of large two-dimensional spaces. *Behaviour & Information Technology*, 9(6):451–466, 1990.

[BP10] Wolfgang Berger and Harald Piringer. Peek brush: A high-speed lightweight ad-hoc selection for multiple coordinated views. In *2010 14th International Conference Information Visualisation*, pages 140–145. IEEE, 2010.

[BS16] Carlo Batini and Monica Scannapieco. Data quality dimensions. In *Data and Information Quality*, pages 21–51. Springer, 2016.

[CBK09] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

[CCM09] Carlos D Correa, Yu-Hsuan Chan, and Kwan-Liu Ma. A framework for uncertainty-aware visual analytics. In *IEEE Symposium on Visual Analytics Science and Technology, 2009 (VAST 2009)*, pages 51–58. IEEE, 2009.

[CKB09] Andy Cockburn, Amy Karlson, and Benjamin B Bederson. A review of overview + detail, zooming, and focus + context interfaces. *ACM Computing Surveys (CSUR)*, 41(1):2, 2009.

[DGK+12] Steven P Dow, Alana Glassco, Jonathan Kass, Melissa Schwarz, Daniel L Schwartz, and Scott R Klemmer. Parallel prototyping leads to better design results, more divergence, and increased self-efficacy. In *Design Thinking Research*, pages 127–153. Springer, 2012.

[Dic15] Oxford English Dictionary. *Visualization.* Oxford University Press, 2015.

[DJ03] Tamraparni Dasu and Theodore Johnson. *Exploratory data mining and data cleaning*, volume 479. John Wiley & Sons, 2003.

[EF10] Niklas Elmqvist and Jean-Daniel Fekete. Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines. *IEEE Transactions on Visualization and Computer Graphics*, 16(3):439–454, 2010.

[EPD05] Cyntrica Eaton, Catherine Plaisant, and Terence Drizd. Visualizing missing data: graph interpretation user study. In *IFIP Conference on Human-Computer Interaction*, pages 861–872. Springer, 2005.

[FB95] George W Furnas and Benjamin B Bederson. Space-scale diagrams: Understanding multiscale interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 234–241. ACM Press/Addison-Wesley Publishing Co., 1995.

[FG14] Sara Johansson Fernstad and Robert C Glen. Visual analysis of missing data - to see what isn't there. In *IEEE Conference on Visual Analytics Science and Technology, 2014 (VAST 2014)*, pages 249–250. IEEE, 2014.

[Fri06] M. Friendly. A brief history of data visualization. In C. Chen, W. Härdle, and A Unwin, editors, *Handbook of Computational Statistics: Data Visualization*, volume III. Springer-Verlag, Heidelberg, 2006.

[GAM+14] Theresia Gschwandtner, Wolfgang Aigner, Silvia Miksch, Johannes Gärtner, Simone Kriglstein, Margit Pohl, and Nik Suchy. Timecleanser: A visual analytics approach for data cleansing of time-oriented data. In *Proceedings of the 14th international conference on knowledge technologies and data-driven business*, page 18. ACM, 2014.

[GGAM12] Theresia Gschwandtner, Johannes Gärtner, Wolfgang Aigner, and Silvia Miksch. A taxonomy of dirty time-oriented data. In *International Conference on Availability, Reliability, and Security*, pages 58–72. Springer, 2012.

[Goo] Google Maps Street View. View of the Upper Belvedere palace in Vienna. `http://maps.google.com/maps`.

[GS06] Henning Griethe and Heidrun Schumann. The visualization of uncertain data: Methods and problems. In *SimVis*, pages 143–156, 2006.

[GTC01] Georges Grinstein, Marjan Trutschl, and Urska Cvek. High-dimensional visualizations. In *Data mining conference KDD workshop 2001*, pages 7–19, 2001.

[HDS+10] Ming C Hao, Umeshwar Dayal, Ratnesh K Sharma, Daniel A Keim, and Halldór Janetzko. Visual analytics of large multidimensional data using variable binned scatter plots. In *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2010.

[Hel08] Joseph M Hellerstein. Quantitative data cleaning for large databases. *United Nations Economic Commission for Europe (UNECE)*, 2008.

[HHSU97] Martin Theus Heike, Heike Hofmann, Bernd Siegl, and Antony Unwin. Manet extensions to interactive statistical graphics for missing values. In *New Techniques and Technologies for Statistics II*, 1997.

[HJ93] Karen Holtzblatt and Sandra Jones. Contextual inquiry: A participatory technique for system design. *Participatory design: Principles and practices*, pages 177–210, 1993.

[HLD02] Helwig Hauser, Florian Ledermann, and Helmut Doleisch. Angular brushing of extended parallel coordinates. In *IEEE Symposium on Information Visualization, 2002 (InfoVis 2002)*, pages 127–130. IEEE, 2002.

[HSBW13] Marcel Hlawatsch, Filip Sadlo, Michael Burch, and Daniel Weiskopf. Scale-stack bar charts. In *Computer Graphics Forum*, volume 32, pages 181–190. Wiley Online Library, 2013.

[IBM] IBM Corporation. IBM Watson Analytics. `http://www.ibm.com/analytics/watson-analytics`. [Online; accessed 15-August-2016].

[ID91] Alfred Inselberg and Bernard Dimsdale. Parallel coordinates. In *Human-Machine Interactive Systems*, pages 199–233. Springer, 1991.

[JSMK14] Halldór Janetzko, Florian Stoffel, Sebastian Mittelstädt, and Daniel A Keim. Anomaly detection for visual analytics of power consumption data. *Computers & Graphics*, 38:27–37, 2014.

[KBFP12] Johannes Kehrer, Roland N Boubela, Peter Filzmoser, and Harald Piringer. A generic model for the integration of interactive visualization and statistical computing using r. In *IEEE Conference on Visual Analytics Science and Technology, 2012 (VAST 2012)*, pages 233–234. IEEE, 2012.

[KCH+03] Won Kim, Byoung-Ju Choi, Eui-Kyeong Hong, Soo-Kyung Kim, and Doheon Lee. A taxonomy of dirty data. *Data mining and knowledge discovery*, 7(1):81–99, 2003.

[Kei02] Daniel A Keim. Information visualization and visual data mining. *IEEE transactions on Visualization and Computer Graphics*, 8(1):1–8, 2002.

[KHG03] Robert Kosara, Helwig Hauser, and Donna L Gresh. An interaction view on information visualization. *State-of-the-Art Report. Proceedings of EUROGRAPHICS*, 2003.

[KHP+11] Sean Kandel, Jeffrey Heer, Catherine Plaisant, Jessie Kennedy, Frank van Ham, Nathalie Henry Riche, Chris Weaver, Bongshin Lee, Dominique Brodbeck, and Paolo Buono. Research directions in data wrangling: Visualizations and transformations for usable and credible data. *Information Visualization*, 10(4):271–288, 2011.

[KKEM10] Daniel A Keim, Jörn Kohlhammer, Geoffrey Ellis, and Florian Mansmann. *Mastering the information age - solving problems with visual analytics*. Florian Mansmann, 2010.

[KMH01] Robert Kosara, Silvia Miksch, and Helwig Hauser. Semantic depth of field. In *IEEE Symposium on Information Visualization 2001 (InfoVis 2001)*, 2001.

[KMS+08] Daniel A Keim, Florian Mansmann, Jörn Schneidewind, Jim Thomas, and Hartmut Ziegler. *Visual analytics: Scope and challenges*. Springer, 2008.

[KPHH11] Sean Kandel, Andreas Paepcke, Joseph Hellerstein, and Jeffrey Heer. Wrangler: Interactive visual specification of data transformation scripts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3363–3372. ACM, 2011.

[KPP+12] Sean Kandel, Ravi Parikh, Andreas Paepcke, Joseph M Hellerstein, and Jeffrey Heer. Profiler: integrated statistical analysis and visualization for data quality assessment. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, pages 547–554. ACM, 2012.

[LA94] Ying K Leung and Mark D Apperley. A review and taxonomy of distortion-oriented presentation techniques. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 1(2):126–160, 1994.

[LGS+14] Alexander Lex, Nils Gehlenborg, Hendrik Strobelt, Romain Vuillemot, and Hanspeter Pfister. Upset: visualization of intersecting sets. *IEEE transactions on Visualization and Computer Graphics*, 20(12):1983–1992, 2014.

[LJH13] Zhicheng Liu, Biye Jiang, and Jeffrey Heer. immens: Real-time visual querying of big data. In *Computer Graphics Forum*, volume 32, pages 421–430. Wiley Online Library, 2013.

[LUM14] LUMA Institute. Vision Statement: A Taxonomy of Innovation, 2014.

[MF05] Heiko Müller and Johann-Christph Freytag. *Problems, methods, and challenges in comprehensive data cleansing.* 2005.

[Moo65] Gordon Moore. Cramming more components onto integrated circuits, 1965.

[MP13] Thomas Mühlbacher and Harald Piringer. A partition-based framework for building and validating regression models. *IEEE transactions on Visualization and Computer Graphics*, 19(12):1962–1971, 2013.

[MPG+14] Thomas Mühlbacher, Harald Piringer, Samuel Gratzl, Michael Sedlmair, and Marc Streit. Opening the black box: Strategies for increased user involvement in existing algorithm implementations. *IEEE transactions on Visualization and Computer Graphics*, 20(12):1643–1652, 2014.

[MR10] Michael J McGuffin and Jean-Marc Robert. Quantifying the space-efficiency of 2d graphical representations of trees. *Information Visualization*, 9(2):115–140, 2010.

[Mun14] Tamara Munzner. *Visualization Analysis and Design.* CRC Press, 2014.

[OM02] Chris Olston and Jock D Mackinlay. Visualizing data with bounded uncertainty. In *IEEE Symposium on Information Visualization, 2002 (InfoVis 2002)*, pages 37–40. IEEE, 2002.

[Ope] Open Source Software. OpenRefine. `http://openrefine.org`. [Online; accessed 15-August-2016].

[PBH08] Harald Piringer, Wolfgang Berger, and Helwig Hauser. Quantifying and comparing features in high-dimensional datasets. In *Proceedings of the 12th International Conference Information Visualisation*, pages 240–245. IEEE, 2008.

[PBK10] Harald Piringer, Wolfgang Berger, and Jürgen Krasser. Hypermoval: Interactive visual validation of regression models for real-time simulation. In *Computer Graphics Forum*, volume 29, pages 983–992. Wiley Online Library, 2010.

[Pie04] Elizabeth M Pierce. Assessing data quality with control matrices. *Communications of the ACM*, 47(2):82–86, 2004.

[Pir11] Harald Piringer. *Large Data Scalability in Interactive Visual Analysis.* PhD thesis, TU Wien, Austria, 2011.

[Pir14] Harald Piringer. Initial hand-drawn sketch of the design of the data quality overview, personal communication, 2014.

[PKH04] Harald Piringer, Robert Kosara, and Helwig Hauser. Interactive focus + context visualization with linked 2d/3d scatterplots. In *Second International Conference on Coordinated and Multiple Views in Exploratory Visualization*, pages 49–60. IEEE, 2004.

[PLW02] Leo L Pipino, Yang W Lee, and Richard Y Wang. Data quality assessment. *Communications of the ACM*, 45(4):211–218, 2002.

[PMZ+13] Niki Popper, Florian Miksch, Günther Zauner, Harald Piringer, Ingrid Wilbacher, and Felix Breitenecker. Ifedh: Solving health system problems using modelling and simulation. *International Journal of Privacy and Health Information Management (IJPHIM)*, 1(2):28–37, 2013.

[PTMB09] Harald Piringer, Christian Tominski, Philipp Muigg, and Wolfgang Berger. A multi-threading architecture to support interactive visual exploration. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1113–1120, 2009.

[PWL97] Alex T Pang, Craig M Wittenbrink, and Suresh K Lodha. Approaches to uncertainty visualization. *The Visual Computer*, 13(8):370–390, 1997.

[RC94] Ramana Rao and Stuart K Card. The table lens: merging graphical and symbolic representations in an interactive focus+ context visualization for tabular information. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 318–322. ACM, 1994.

[RD00] Erhard Rahm and Hong Hai Do. Data cleaning: Problems and current approaches. *IEEE Data Eng. Bull.*, 23(4):3–13, 2000.

[RH01] Vijayshankar Raman and Joseph M Hellerstein. Potter's wheel: An interactive data cleaning system. In *International Conference on Very Large Data Bases (VLDB)*, volume 1, pages 381–390, 2001.

[Rob07] Jonathan C Roberts. State of the art: Coordinated & multiple views in exploratory visualization. In *Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization*, pages 61–71. IEEE, 2007.

[SBM+14] Martin Steiger, Jürgen Bernard, Sebastian Mittelstädt, Hendrik Lücke-Tieke, Daniel Keim, Thorsten May, and Jörn Kohlhammer. Visual analysis of time-series similarities for anomaly detection in sensor networks. In *Computer Graphics Forum*, volume 33, pages 401–410. Wiley Online Library, 2014.

[SEG05] Rajmonda Sulo, Stephen Eick, and Robert Grossman. Davis: a tool for visualizing data quality. *Posters Compendium of InfoVis*, 2005:45–46, 2005.

[Shn96] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pages 336–343. IEEE, 1996.

[Sil86] Bernard W Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.

[SLBC03] Deborah F Swayne, Duncan Temple Lang, Andreas Buja, and Dianne Cook. Ggobi: evolving from xgobi into an extensible framework for interactive data visualization. *Computational Statistics & Data Analysis*, 43(4):423–444, 2003.

[SLH+11] Lei Shi, Qi Liao, Yuan He, Rui Li, Aaron Striegel, and Zhong Su. Save: Sensor anomaly visualization engine. In *IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 201–210. IEEE, 2011.

[Sno55] John Snow. *On the mode of communication of cholera*. John Churchill, 1855.

[SS05] Jinwook Seo and Ben Shneiderman. A rank-by-feature framework for interactive exploration of multidimensional data. *Information visualization*, 4(2):96–113, 2005.

[STH02] Chris Stolte, Diane Tang, and Pat Hanrahan. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):52–65, 2002.

[SYI11] Shazia Sadiq, Naiem Khodabandehloo Yeganeh, and Marta Indulska. 20 years of data quality research: themes, trends and synergies. In *Proceedings of the Twenty-Second Australasian Database Conference-Volume 115*, pages 153–162. Australian Computer Society, Inc., 2011.

[Tab] Tableau Software. Tableau. `http://www.tableau.com`. [Online; accessed 15-August-2016].

[TAF12] Matthias Templ, Andreas Alfons, and Peter Filzmoser. Exploring incomplete data using visualization techniques. *Advances in Data Analysis and Classification*, 6(1):29–47, 2012.

[Tal] Talend Inc. Talend. `https://www.talend.com`. [Online; accessed 15-August-2016].

[Tho05] James J Thomas. *Illuminating the path:[the research and development agenda for visual analytics]*. IEEE Computer Society, 2005.

[TIB] TIBCO Spotfire. Spotfire. `http://spotfire.tibco.com`. [Online; accessed 15-August-2016].

[Tri] Trifacta Inc. Trifacta. http://www.trifacta.com. [Online; accessed 15-August-2016].

[TSDS96] Lisa Tweedie, Robert Spence, Huw Dawkes, and Hus Su. Externalising abstract mathematical models. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 406–ff. ACM, 1996.

[Twi] Twitter Corporation. New Tweets per second record, and how! https://blog.twitter.com/2013/new-tweets-per-second-record-and-how. [Online; accessed 15-August-2016].

[vWN04] Jarke J van Wijk and Wim AA Nuij. A model for smooth viewing and navigation of large 2d information spaces. *IEEE Transactions on Visualization and Computer Graphics*, 10(4):447–458, 2004.

[War12] Colin Ware. *Information visualization: perception for design*. Elsevier, 2012.

[WBWK00] Michelle Q Wang Baldonado, Allison Woodruff, and Allan Kuchinsky. Guidelines for using multiple views in information visualization. In *Proceedings of the working conference on Advanced visual interfaces*, pages 110–119. ACM, 2000.

[WS96] Richard Y Wang and Diane M Strong. Beyond accuracy: What data quality means to data consumers. *Journal of management information systems*, 12(4):5–33, 1996.

[WV12] William Wong and Margaret Varga. Black holes, keyholes and brown worms: Challenges in sense making. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 56, pages 287–291. SAGE Publications, 2012.

[WXYR11] Matthew Ward, Zaixian Xie, Di Yang, and Elke Rundensteiner. Quality-aware visual data analysis. *Computational Statistics*, 26(4):567–584, 2011.