

Ein Visual Analytics Ansatz zur Datenqualitäts-Beurteilung mithilfe von Zeit, Metriken, Unsicherheiten und Provenienz

DISSERTATION

zur Erlangung des akademischen Grades

Doktor der Technischen Wissenschaften

eingereicht von

Christian Bors, MSc.

Matrikelnummer 00626189

an der Fakultät für Informatik
der Technischen Universität Wien

Betreuung: Univ.Prof. Mag.rer.soc.oec. Dr.rer.soc.oec. Silvia Miksch

Diese Dissertation haben begutachtet:

Kai Xu

Axel Polleres

Wien, 5. November 2019

Christian Bors

Facilitating Data Quality Assessment Utilizing Visual Analytics: Tackling Time, Metrics, Uncertainty, and Provenance

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doktor der Technischen Wissenschaften

by

Christian Bors, MSc.

Registration Number 00626189

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Prof. Mag.rer.soc.oec. Dr.rer.soc.oec. Silvia Miksch

The dissertation has been reviewed by:

Kai Xu

Axel Polleres

Vienna, 5th November, 2019

Christian Bors

Erklärung zur Verfassung der Arbeit

Christian Bors, MSc.

Hiermit erkläre ich, dass ich diese Arbeit selbständig 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, 5. November 2019

Christian Bors

Danksagung

Für meine Verlobte, Familie und Freunde.

Acknowledgements

To my fiancé, my family, and my friends.

I want to thank all my current and former colleagues from the CVASt research group, Silvia Miksch, Alessio Arleo, Markus Bögl, Davide Ceneda, Velitchko Filipov, Theresia Gschwandtner, Roger A. Leite, Nikolaus Piccolotto, Victor Schetinger, Bilal Alsallakh, Albert Amor-Amorós, Paolo Federico, Tim Lammarsch. I thank my collaborators from the Human-Computer Interaction Group IGW at TU Wien – Simone Kriglstein and Margit Pohl –, my collaborators from FH St. Pölten – Wolfgang Aigner, Christina Niederer, Alexander Rind, and Markus Wagner –, and my international collaborators from TU Darmstadt – Jürgen Bernard and Jörn Kohlhammer – and Universität Rostock – Christian Eichner, Christian Tominski, and Heidrun Schumann.

To all researchers and industry experts interested in Information Visualization and Visual Analytics, who are passionate to share their interests and knowledge and have given me valuable feedback on my work at conferences, at workshops, and also later in the pub.

Lastly, I want to thank all reviewers who spent time giving valuable and constructive feedback on my works.

Kurzfassung

Die visuelle und interaktive Datenanalyse stellt ein großes Forschungsgebiet dar, das sich erfolgreich in kommerziellen Anwendungen und Systemen durchgesetzt hat um Analytikern zu ermöglichen Schlüsse und Erkenntnisse aus ihren Daten zu ziehen. Im Normalfall befinden sich in den Daten Fehler, die es meistens erschweren oder gar unmöglich machen mit einer bestehenden Analyse zu beginnen ohne zuvor Vorverarbeitungsschritte durchzuführen. Mit Visual Analytics Methoden ist es möglich Fehler zu identifizieren oder zu korrigieren und die Daten in ein nutzbares Format zu überführen. Jedoch müssen dabei unterschiedliche Aspekte berücksichtigt werden: (1) welche Operationen angewandt wurden um Fehler zu beheben, (2) welche Auswirkungen sie auf den Datensatz hatten und (3) ob die verwendeten Routinen die Fehler auch auf angemessene Weise behoben haben. In dieser Dissertation werden Datenqualitätsmetriken und Unsicherheitsmaße berechnet um Provenienz aus Bearbeitungs-Operationen und Vorverarbeitungs-Pipelines zu erstellen. Im Kontext dieser Arbeit werden Qualitätsmetriken als Maße definiert, die die Prävalenz von Fehlern in einem Datensatz beschreiben, Unsicherheiten werden eingesetzt um das Ausmaß von Unschärfe, die durch Verarbeitungsschritte entsteht, zu quantifizieren. Werden solche Maße als Provenienz über die Zeit gespeichert, ist es Analysten möglich abzuschätzen wie Vorverarbeitungsschritte einen Datensatz verändert haben und ob Probleme, die zu Beginn entdeckt wurden, mit Operationen gelöst wurden, die die restlichen Daten kaum verändert haben. Das stellt sicher dass der veränderte Datensatz möglichst dem originalen Datensatz entspricht.

Im Zuge dieser Dissertation wurde eine Benutzer-zentrierte Design Methodologie verwendet um Visual Analytics Prototypen und Visualisierungs-Techniken zu entwickeln. Die Designs haben das Ziel die Forschungsfelder Datenqualität, Provenienz und Unsicherheiten in einem interaktiven Ansatz zu vereinen. In dieser Arbeit präsentiere ich (1) eine neue Methode Datenqualitätsmetriken zu erstellen und anzupassen um Qualitätsprobleme in tabellaren und zeitorientierten Datensätzen zu analysieren; (2) ein Provenienz-Modell basierend auf Bearbeitungs-Operationen, welches auf Datenqualitätsmetriken aufbaut um die Qualitätsentwicklung über alle Verarbeitungs-Schritte hinweg zu verfolgen; und (3) Methoden welche es ermöglichen Unsicherheiten in univariaten und multivariaten Zeitreihen zu quantifizieren und zu visualisieren. Diese Unsicherheiten ermöglichen es das Ausmaß an Unschärfe auf eine Zeitreihe durch eine Bearbeitungs-Operation abzuschätzen und so eine angemessene Parameter-Einstellung zu finden, die Fehler und Rauschen aus der Zeitreihe entfernt ohne unnötig viel Information zu verlieren. All diese Anwendungen

und Methoden wurden basierend auf angewandten Echtzeit-Nutzungsszenarien entwickelt und wurden mittels qualitativen und quantitativen Studien evaluiert um sicherzustellen, dass es Analysten erlaubt ist die Probleme angemessen zu lösen.

Die Ergebnisse des iterativen Designs und der Evaluierung zeigen dass Datenqualitätsmetriken und quantifizierte Unsicherheiten angemessen sind, um den allgemeinen Qualitätszustand eines gegebenen Datensatzes abzuschätzen. Weiters konnte ich finden dass Datenqualitätsinformationen ebenfalls effektiv dafür verwendet werden können, um einen Provenienz-Graphen aus Datenverarbeitungs-Prozessen zu annotieren damit die Veränderung der Qualität über die Zeit bestimmt werden kann. Unsicherheiten, die aus Vorverarbeitungs-Pipelines generiert werden, können verwendet werden um die Auswirkungen der Operationen genauer zu betrachten und so dem Analysten helfen, eine Balance zwischen notwendiger und exzessiver Vorverarbeitungsschritte zu finden.

Abstract

Visual and interactive data analysis is a large field of research that is successfully used in commercial tools and systems to allow analysts make sense of their data. Data is often riddled with issues, which makes analysis difficult or even not feasible. Pre-processing data for downstream analysis also involves resolving these issues. We may employ Visual Analytics methods to identify and correct issues and eventually wrangle the data into a usable format. Various aspects are critical during issue correction: (1) how are the issues resolved, (2) to what extent did this affect the dataset, and (3) did the used routines actually resolve the issues appropriately. In this thesis I employ data quality metrics and uncertainty to capture provenance from pre-processing operations and pipelines. Data quality metrics are used to show the prevalence of errors in a dataset, and uncertainty can quantify the changes applied to a data values and entries during processing. Capturing such measures as provenance and visualizing it in an exploratory environment can allow analysts to determine how pre-processing steps affected a dataset, and if the issues, that were initially discovered, could be resolved in a minimal way, so the data is representative of the original dataset.

Within the course of this thesis I employed a user-centered design methodology to develop Visual Analytics prototypes and visualization techniques that combine techniques from data quality, provenance, and uncertainty research. This work presents (1) a novel method to create and customize data quality metrics that can be employed to explore quality issues in tabular and time-oriented datasets, (2) a provenance model for capturing provenance from data pre-processing, leveraging data quality metrics, and using visualization to show the development of quality throughout a pre-processing workflow, and (3) methods for quantifying and visualizing uncertainty in univariate and multivariate time series to analyze the influence of pre-processing operations on the time series. These approaches were developed using real-world use cases and scenarios and were evaluated using qualitative and quantitative user studies to validate the appropriateness of my approaches. The results of the iterative design and evaluation shows that data quality metrics and uncertainty quantified from data pre-processing can be used to assess the overall quality of a dataset. The data quality can furthermore be used to annotate provenance captured during data wrangling, which allows analysts to understand and track the development of quality in a dataset. Uncertainty quantified from pre-processing can be used to assess the impact that pre-processing operations have on datasets and thus support analysts find a balance between necessary and excessive pre-processing.

Contents

Kurzfassung	xi
Abstract	xiii
Contents	xv
I The Problem	1
1 Introduction	3
1.1 Motivation	3
1.2 A Visual Analytics Approach	4
1.3 Problem Analysis	6
1.4 Research Questions	14
1.5 Structure	14
2 Related Work	17
2.1 Foundation and Definitions	17
2.2 Visual Encodings of Data Quality	38
2.3 Interactive Methods for Data Quality Assessment	49
2.4 Visualizing Uncertainty	54
2.5 The Role of Uncertainty in Visualization and Visual Analytics	59
2.6 Methods for Visualizing Provenance	64
2.7 Visual Analytics Methods Leveraging and Analyzing Provenance	67
2.8 Connecting Data Quality, Uncertainty, and Provenance	70
II The Proposed Solution	73
3 Conceptualizations	75
3.1 Defining Data Quality Metrics	75
3.2 Uncertainty in Time Series Pre-Processing	81
3.3 Data and Insight Provenance from Data Quality	85
	xv

4	Visual-Interactive Customization of Data Quality Metrics	89
4.1	Requirements Analysis	90
4.2	Design Rationales	91
4.3	Visualization Design	93
5	Capturing Provenance from Data Wrangling	99
5.1	Provenance Model Implementation	100
5.2	Requirements Analysis	101
5.3	Task Considerations	102
5.4	Usability Inspection Study	103
5.5	Design Rationales	105
5.6	Visualization Design	107
6	Quantifying Uncertainty in Time Series Pre-Processing	111
6.1	Quantifying Uncertainty from Rastering	111
6.2	Quantifying Uncertainty from General Pre-Processing	116
III The Evaluation		119
7	Case Studies	121
7.1	Case Study – Analyzing ISP Connectivity Data	121
7.2	Case Study – Analyzing Provenance from Wrangling Operations	124
7.3	Usage Scenario – Quantifying Uncertainty from Pre-processing Weather Experiment Data	127
7.4	Lessons Learned	129
8	Iterative Design Process and Evaluation	131
8.1	Iteration One – Conceptual Design	133
8.2	Iteration Two – Design Evaluation	134
8.3	Iteration Three – Focus Group Evaluation	135
8.4	Iteration Four – Final Development and Inspection	136
8.5	Results	137
8.6	Discussion & Lessons Learned	138
9	Qualitative User Experience Evaluation	141
9.1	Procedure	141
9.2	Results	143
9.3	Discussion & Lessons Learned	143
10	Visualizing Uncertainty of Segmented Time Series	145
10.1	Design Goals	146
10.2	Visualization Design	147
10.3	Study Design	149
10.4	Results	152

10.5 Discussion & Lessons Learned	156
IV The Conclusion	159
11 Conclusions & Limitations	161
11.1 Summary of Contributions	161
11.2 Answering My Research Questions	163
11.3 Publications and Dissemination	164
11.4 Future Directions	166
V Appendix	169
12 DQProv Explorer – Qualitative User Study	171
12.1 Evaluation Structure	171
12.2 Summarized Results	173
13 DQProv Explorer – Usability Inspection	185
13.1 Evaluation Structure	185
14 Visualizing Uncertainty in Time Series Processing	191
14.1 Questions and Results per Question	191
14.2 User Study Results - Uncertainty in Time Series Segmentation Results	195
Glossary	209
Acronyms	211
Bibliography	213

Part I

The Problem

Introduction

1.1 Motivation

Data analysis is likely preceded by a pre-processing workflow to bring the data into usable form. People spend countless hours trying to prepare data in a particular way, changing the structure, transforming values, and cleansing data entries. To make data analysis possible, the data often need to be cleaned, aggregated, transformed into a particular format. Pre-processing data is an iterative process in which transformations are applied consecutively. It fosters awareness of Data Quality (DQ) for the next steps in the data analysis process and for the interpretations of the results and insights. Analysts often find themselves in a dilemma of DQ: are the data sufficiently clean so that they can commence downstream analysis?

Performing data cleansing, i.e., ridding the dataset of quality problems and inconsistencies, is necessary to correctly identify different types of errors in the dataset and appropriately resolve them. Different cleansing operations will change the effects on the data. For instance, if an entry is missing individual cells, an analyst might remove these entries if they contained vital information, and in turn continue with a smaller dataset. Alternatively, if non-critical information is missing but can be imputed (e.g., by interpolating values), the analyst might try to keep these entries and retain the original dataset's size. Downstream analysis often requires data to be available in a particular format and structure, which demands re-formatting the original data. For recurring analysis with updated or streaming input data, automatic scripting can be set up to automatically pre-process the new datasets. However, auditing the results of these automatic transformations is rarely possible without requiring detailed inspection of the original and pre-processed data. This is exacerbated by large data sizes: Detailed inspection of the raw data to look for unresolved issues is impossible without assistance.

Attempting different methods to prepare a dataset can be valuable, analysts become familiar with the attributes of the data, the value and application domain, and the

dataset in general, but they often lead to dead ends. Frequently, analysts identify errors in their pre-processing routines and need to revert the data processing, resulting in tediously re-tracing their own actions. These operations and transformations are often scattered across multiple applications, self-written scripts, or online processing tools. Without providing detailed logging of the applied operations, parameter settings, and their order, it can be pointless to reproduce the actions taken. On the other side, an already pre-processed dataset without further information on what changes were applied and how they affected the dataset can make analysts unfamiliar with the data doubt their usefulness and trustworthiness. Trust is increasingly important in data analysis, and the provenance of a dataset, i.e., storing any change applied to data, can be useful for improving analysts' confidence in the data.

According to Kandel et al., data wrangling is defined as “a process of iterative data exploration and transformation that enables analysis” [KHP⁺11, p. 272]. Data cleansing can be understood as the process of “detecting and removing errors and inconsistencies from data in order to improve the quality of data” [RD00, p. 1]. Visual Interactive methods can facilitate both data wrangling and cleansing approaches. Combining automated techniques for detecting issues with interactive visualizations, analysts can explore the dataset, identify issues and subsequently more appropriately resolve them. By computing measures of quality and allowing analysts to investigate the prevalence of problems in the dataset, they can make an informed decision if the data are usable in their current state, or if it is necessary to resolve these issues. This depends on the downstream analysis task, so analysts need to use their expertise and domain knowledge to assess quality.

Current data wrangling and cleansing allow visually profiling data and give a good overview of the raw data. However, quality issues are often domain-specific, and analyst unfamiliar with the data could overlook issues quite easily. By employing automatically generated measures of quality to familiarize the analyst with the dataset, data wrangling and cleansing can be facilitated by signaling if and where quality issues still remain. In this work I will present new methods for detecting and exploring quality issues in data wrangling and cleansing. I leverage DQ metrics to show the distribution of errors and inconsistencies in the dataset, and employ this information to show the development of quality across multiple transformation steps and wrangling branches, as well as utilize the gathered information to quantify uncertainty that is inevitably introduced by wrangling and cleansing a dataset. This will be illustrated on the concrete scenario of multivariate time series data, a data domain where pre-processing is vital due to the often big scale of the data.

1.2 A Visual Analytics Approach

Visual Analytics (VA) is defined as “science of analytical reasoning facilitated by visual interactive interfaces”, according to Thomas and Cook [TC05, p. 4]. This field aims to (i) enable analysts to obtain deep insights that directly support assessment, planning, and decision-making, (ii) leverage human perception and the ability to understand large

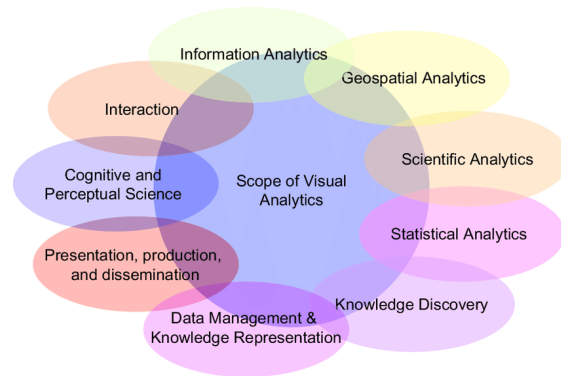


Figure 1.1: The scope of VA is an interdisciplinary field of different scientific research disciplines [KMS⁺08].

amounts of information, (iii) transform and represent data to support visualization and analysis, and (iv) support production, presentation, and dissemination of analysis results. More specifically, Keim et al. [KKEM10] state that VA techniques enable analysts to synthesize information and derive insight, detect the expected and discover the unexpected, provide timely, defensible, and understandable assessments, and communicate these assessment effectively for action. Thomas and Cook described the analytical reasoning process to be iterative and support users in the sense-making task, which can also be understood as knowledge crystallization. There have been various definitions of the VA process and sensemaking in VA. Keim et al. [KAF⁺08] characterized the process of VA as shown in Figure 1.2.

VA is an interdisciplinary field of research involving multiple disciplines, trying to leverage human cognition as well as automated processing. As such, it combines data management and processing, knowledge discovery, cognition and perception, statistical analysis, and visualization research, among others (see Figure 1.1). The goal is to leverage the combined strength of the different disciplines to emphasize human cognition and facilitate insight generation and decision-making, while making the process traceable and the results comprehensible by others.

Sense-making, as part of the iterative analysis process [TC05], is based on the iterative sense-making loop for analysis by Pirolli and Card [Pir05]. They identified the process to consist of two major loops, the *foraging loop* where users aim for seeking and extracting information, and the *sense-making loop* that supports the user with building a mental model that is based on the evidence found in the *foraging loop*. Users can iterate these loops in bottom-up or top-down processes. In cognitive science, Klein et al. [KMH06] presented a data/frame theory of sense-making, positing a closed-loop transition sequence of mental model formation (backward-looking and explanatory) and mental simulation (forward-looking and anticipatory). The sense-making models have been under continued refinement, e.g., Sacha et al. derived a knowledge generation model [SSS⁺14], Federico et al. [FWR⁺17] proposed to use explicitly and implicitly generated knowledge to assist

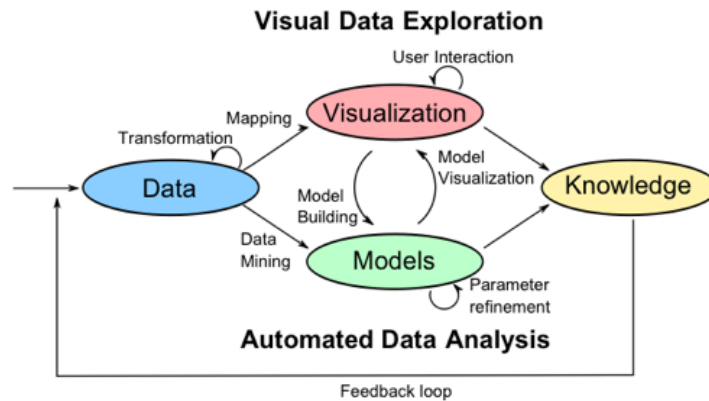


Figure 1.2: The VA process combines automated analysis and visual data exploration. To discover insights and gain knowledge, there is a tight coupling between data, visualization, models and the user [KAF⁺08].

VA and sense-making processes. Both approaches, however, made a clear distinction in the knowledge generation models between the computational/machine and human/user parts. Both approaches show that knowledge is a central element that can be fed back into the model, but ultimately is the goal of an analysis process.

1.3 Problem Analysis

Visual-interactive data wrangling and cleansing approaches employ data profiling techniques to gain insights into the currently analyzed dataset. Summary visualizations are used to give analysts an overview of the data. In data cleansing these profiling techniques are also used for detecting quality issues or inconsistencies that signal problems within the dataset. Usually, particular application domains require specific anomaly detection methods. The role of context for determining issues within a dataset is important, but rarely considered in general data wrangling and cleansing tools. Hence, analysts resort to specific tools that allow them to determine contextual problems in the dataset. In general, current approaches in scientific literature lack the ability to detect and communicate different types of quality issues, so that analysts are able to explore the issues of an entire dataset consistently.

I propose to introduce customizable quality measures that detect domain-specific issues. These quality measure can then be leveraged to support analysts during various data wrangling or cleansing tasks: (i) Analysts can explore the distribution of quality issues throughout the dataset. (ii) The quality measures can be computed throughout the entire wrangling and cleansing process to log the development of quality over time. (iii) Saving the changes in the dataset between transformation enables tracking how much the data were changed, which can be used to quantify uncertainty from the wrangling/cleansing process.

Combining the use of quality measures with a VA approach for analyzing the prevalence of quality issues and the qualitative state of the dataset will allow analysts to make informed decisions on the usability of a dataset for downstream analysis. By employing Shneiderman’s Visual Information Seeking Mantra [Shn96], I plan to provide users with an overview first methodology, and provide means for filtering the data and getting details on demand. Computing quality measures throughout the entire wrangling and cleansing process and storing it as data provenance is expected to enable analysts explore the development of quality over time. Provenance can also be explored by analysts to make sense of preceding wrangling and cleansing efforts.

When wrangling and cleansing data, the original data is inevitably altered. Specifically, in time series analysis, the data must be pre-processed, e.g., to smooth out sensor noise, or reduce the data resolution by sampling. These changes reduce faithfulness in the data and could cover up patterns in the data and analysts should be aware of these intrinsic uncertainty. By quantifying uncertainty from data wrangling and cleansing operations, it can be externalized and communicated to the analyst. This allows decision-making under the awareness of uncertainty.

1.3.1 Research Methodology

VA solutions need to satisfy both the integration of automated methods by employing visual interfaces and other input techniques while providing visual methods for interactive exploration. The applicability of these methods needs to be evaluated to validate if the employed methods are appropriate for the data, the users, and the tasks. Miksch and Aigner [MA14] derived a design triangle for designing VA of time-oriented data, which I will use for deriving the design requirements of my VA solutions. They determined that requirements towards developing VA solutions center around what kinds of *data* is used, who the target *users* are, and what *tasks* these users want to conduct. These three considerations are used to develop visual encodings, analytical and interaction methods that satisfy three characteristics [MA14]: (1) *Expressiveness* refers to the requirement of showing exactly the information contained in the data; nothing more and nothing less must be visualized [Wij06], (2) *Effectiveness* addresses that the visual encoding used leverages cognitive abilities to detect contextually relevant information [Wij06], (3) *Appropriateness* involves the cost-value ratio for users to benefit from the employed VA methods for a particular task [Mac86]. For visualization design and development, Munzner’s Nested Model [Mun09] is employed. This model splits the visualization design process into four levels and suggests evaluation methodologies for each level to ensure these steps are explicitly followed and successfully passed before proceeding to the next design step. The four levels of design consist of (i) characterizing problems and the data, (ii) mapping them to abstract operations and data types, (iii) design using visual encodings and interaction techniques, and (iv) creating algorithms for execution. In particular, Munzner also lists the threats and possible validation methods of these nested levels, as can be seen in Figure 1.4. This are considered to avoid pitfalls during design and development, and choose an appropriate evaluation method for the individual levels.

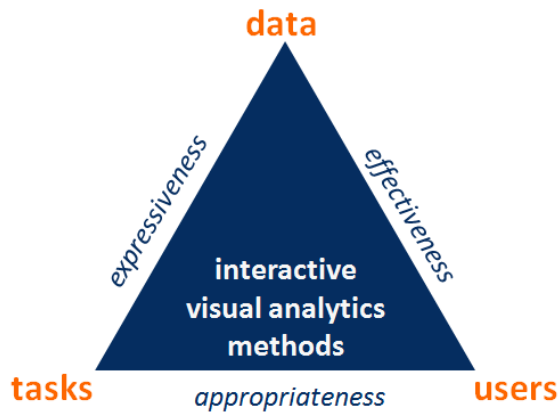


Figure 1.3: The design triangle for designing VA of time-oriented data by Miksch and Aigner [MA14].

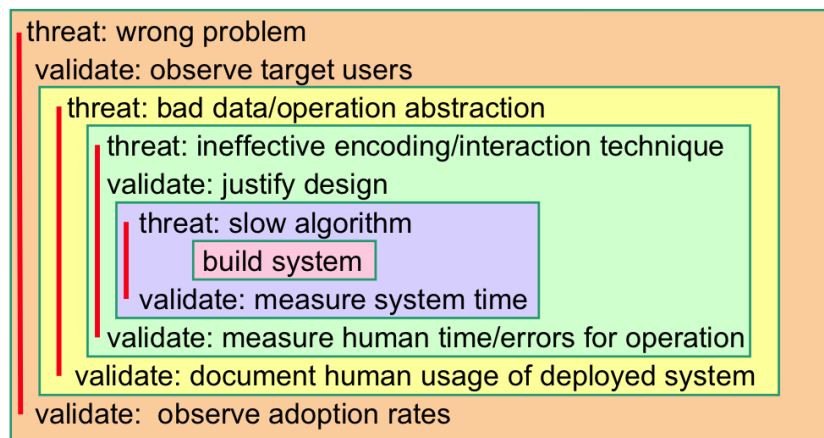


Figure 1.4: A four-level nested model for designing and evaluating visualizations [Mun09].

Van Wijk presented a model of visualization [Wij06] and proposed an iterative design cycle: (i) Setting up requirements, (ii) generating multiple possible solutions, and (iii) evaluating these solutions towards the requirements to determine the best solution. He furthermore associated a cost with developing and using visualizations, and considered it to be measured for its utility by evaluating these costs. They range from initial development cost, cost per user and per session – these might be computational costs – to perception and exploration costs – time the user has to spend learning and getting used to the visualization and possible interactions.

Iterating upon the nested model for visualization design and development, Federico et al. [FAAM16] presented a nested workflow model for VA design and validation. They generalized the design triangle for VA design of time-oriented data and introduced nesting design levels into the three components of data, users, and tasks. They described tasks

to be the central component of the nested workflow. For this matter they nest different aspects within task characterization: (i) problem domain, (ii) operation abstraction, (iii) visual encodings and interaction techniques, and (iv) algorithms. The data can be seen as input and output of a VA solution and as such requires problem domain and abstraction, whereas knowledge gained throughout an analysis should also be externalized. The users need to be considered in the nested workflow, being affected by the problem domain and being part of the interactive system. The workflow model is supposed to evaluate whether users, given data or knowledge as input, can fulfill a particular task on different levels of the nested workflow.

Brehmer and Munzner [BM13] engaged in abstracting and generalizing visualization tasks by presenting a multi-level typology of abstract visualization tasks. They distinguish this typology by three main questions: *why*, *how*, and *what*. The abstract task typology can be used to describe existing visualization and VA tools. But it is also possible to use it to prescribe and inform the process of designing new visualization or VA solutions. These three main questions are also incorporated in Federico et al.'s nested workflow model to decompose the characterized tasks on different levels of nesting. They showed how their nested workflow model can be applied “to understand users’ environments, work practices, and visual data analysis reasoning” [FAAM16, p. 6]. In Section 1.2 we discussed the influence of sense-making and knowledge on VA. Matching these abstract tasks to high-level decision making and validating if these tasks correspond to users’ mental models, Lam et al. [LTM18] attempted to derive tasks from analysis goals. This should allow designing visualization and VA solutions that are more appropriate and effective for the users and the tasks they are targeted at.

Pirolli and Card’s [Pir05] sense-making loop (see Figure 1.5) shows cycles and iterative processes within the model indicate that foraging and sense-making is invoked in bottom-up and top-down in an opportunistic way. Thus, if new insights suggest forward (bottom-up) or backward (top-down) actions, users will iterate the model accordingly. The sense-making loop can be utilized in VA solution development to determine which actions should be supported. However, the model does not distinguish processes to be executed by the human or the computer within a VA approach.

The knowledge generation model for VA presented by Sacha et al. [SSS⁺14] builds on the exploration and the sense-making loop and separates tasks performed by the computer system and the human component. Furthermore, the human reasoning process is extended by a verification loop. The particularity of VA approaches allows a close connection between the human and the computer through interaction and feedback observation, as well as taking actions depending on the findings or the analysis goal (see Figure 1.6). The goal of tightly coupling computer and human is to minimize the gulf of execution and evaluation, as defined by Norman [Nor88, pp. 38]. They mention the role of trust in this loop, which depends on the knowledge gained from confirming hypotheses. Sacha et al. further iterated on this model to include uncertainty propagation and human trust building in [SSK⁺16]. Different sources of uncertainty are inherent to multiple stages of the knowledge generation model, and Sacha et al. discuss their propagation and possible

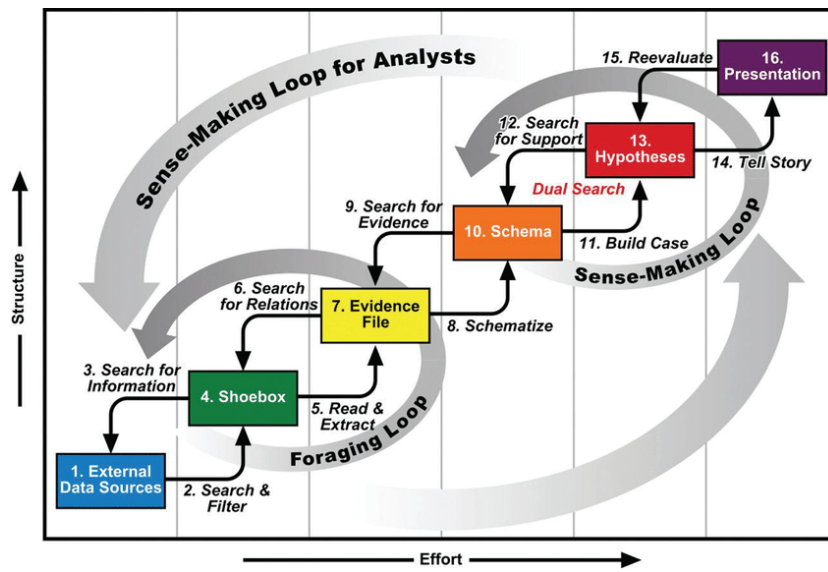


Figure 1.5: Pirolli and Card's [Pir05].

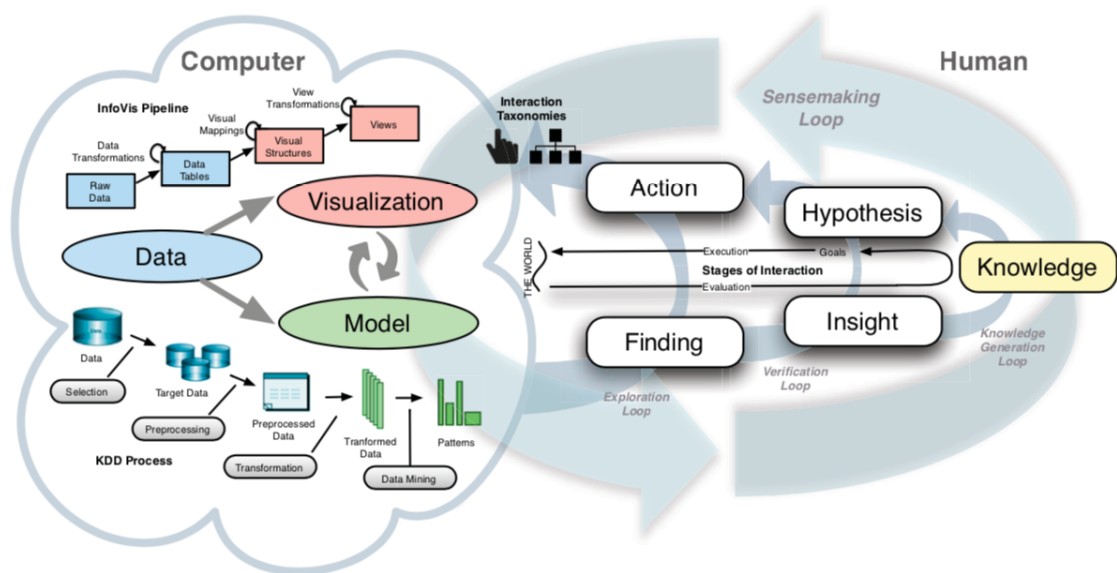


Figure 1.6: Knowledge generation model for VA by Sacha et al. [SSK⁺16].

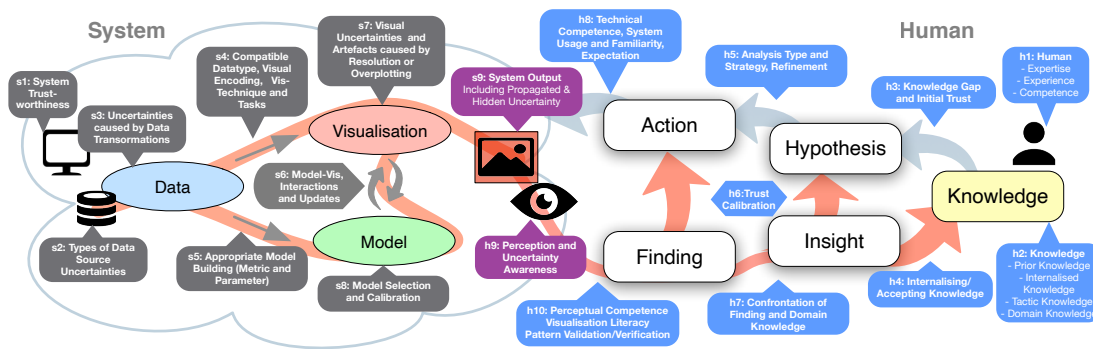


Figure 1.7: Knowledge generation model for VA including uncertainty propagation and human trust building by Sacha et al. [SSK⁺16].

influences on steering and user intent. One of their propositions was to “analyze human behavior in order to derive hints on problems and biases” [SSK⁺16, p. 7].

1.3.2 Evaluation

One vital topic in VA and visualization research is evaluating the outcome of the design and implementation. Evaluating visualization and VA methods helps us understand how users interact with the visualization/VA solution, and if the developed methods could improve analysis or reasoning over existing approaches. Lam et al. [LBI⁺12] presented seven types of evaluation used for evaluating visualization techniques or systems. They differentiate between:

Understanding data analysis: These types of evaluation scenarios focus on trying to externalize and quantify the mental models, user methodologies, and how well visualizations and interactions suit these. (1) *Understanding Environments and Work Practices (UWP)*: Eliciting formal requirements for design by understanding the work, analysis, or information processing practices. (2) *Evaluating Visual Data Analysis and Reasoning (VDAR)*: Assessing a visualization tool’s ability to support visual analysis and reasoning about data. This is done for a tool as a whole, as opposed to certain functionality (compare *Evaluating User Performance*). However, it must be defined how reasoning can be done within such tool, and how it is captured in an evaluation environment. (3) *Evaluating Communication through Visualization (CTV)*: Studying how communication can be supported with visualization. Evaluation measures how effectively messages are delivered and acquired, often through qualitative or quantitative metrics, e.g., learning rate, or participants’ learning approaches. (4) *Evaluating Collaborative Data Analysis (CDA)*: Evaluating how a visualization tool supports collaborative analysis and/or decision-making processes.

Understanding visualizations: Opposed to the first set of scenarios, where evaluation tries to encapsulate how data analysis is conducted. This set of evaluation scenarios focuses on understanding how the visualizations themselves are perceived and used. (1) *Evaluating*

User Performance (UP): Quantifying measurable metrics of user performance often analyzed with descriptive statistics. (2) *Evaluating User Experience (UE)*: Determining users' experience and reaction to a visualization technique or system, which is often a subjectively observed, collected, and measured result. (3) *Evaluating Visualization Algorithms (VA)*: Scoring visualization algorithms against existing solutions, in terms of performance or quality.

The types of evaluation can be matched to Munzner's nested model for visualization design and development. Furthermore, Isenberg et al. [IIC⁺13] added another category to these categories: *Qualitative results inspection (QRI)*, which describes descriptive statements by the author which should be implicitly assessed by the reader. They also categorized the types of users demonstrated in case studies for *UWP/VDAR* scenarios: (1) Domain experts, (2) Close collaborations between visualization researchers and domain experts, (3) descriptions by visualization researchers, and (4) usage scenarios described by visualization researchers. They also discussed considerations for future evaluation of visualization design and systems: The authors should provide a clear statement on the scientific contribution and application in real-world scenarios, with an indication towards reporting *UWP*, *VDAR*, *CTV*, and *CDA* evaluation scenarios. Evaluation and reporting on evaluation should be done rigorously, including "who participated", "collaboration details", "how many people participated", "the study protocol", "controlling experiments with rigor", "reporting qualitative results inspection with rigor."

While [IIC⁺13, LBI⁺12] described different types of evaluation types, Sedlmair et al. [SMM12] determined a design study methodology framework. It consists of three phases, *precondition*, *core*, and *analysis*. The framework should be followed linearly, however the process is dynamically iterative, indicated by the backwards-facing arrows in Figure [SMM12]. For example, the analysis phase could also be started early in the process, which could in turn require re-thinking the chosen abstractions to more clearly articulate them. In the discussion, they stress one important aspect for evaluating and interpreting results from a design study: "transferability is the goal, not reproducibility" [SMM12, p. 2438]. So far, the methods for evaluating visualization and VA approaches is done by formalizing user studies and testing hypotheses. Andrienko et al. [ALA⁺18] presented a conceptual framework that defines VA as goal-oriented workflow that produces a model in the end, reflected in the data. Knowledge externalization and representation to "capture, store, and reuse the knowledge generated throughout the entire analytic process" [TC05, p.42] has continued to be an important concept, and for many of the presented models (e.g., [SSS⁺14, FWR⁺17]) it is considered as a central element that is generated from – and directed back into – the visual analysis process. Simultaneously, the mental model should be externalized for later recall and collaboration. Andrienko et al. advocate for collecting model provenance (also defined as analytic provenance by Ragan et al. [RESC16]) to record the analysis process. They propose to validate a developed VA solution against their proposed representation of the VA workflow, as can be seen in Figure 1.9.

I intend to leverage existing taxonomies on provenance, DQ, and uncertainty analysis

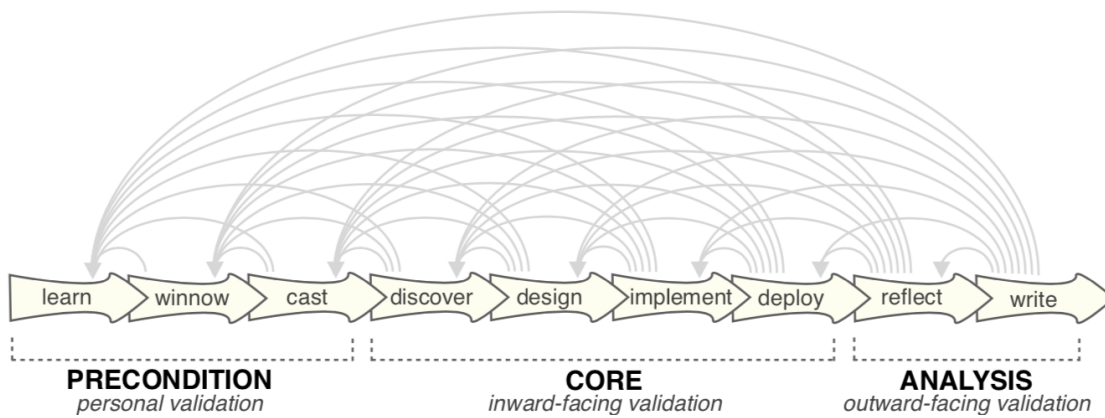


Figure 1.8: Nine-stage design study methodology framework by Sedlmair et al. [SMM12].

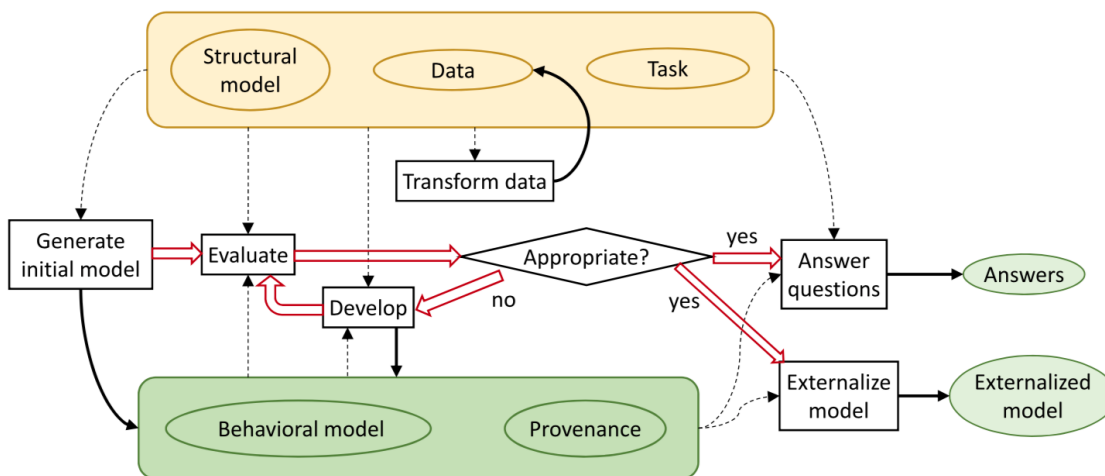


Figure 1.9: Representation of the VA workflow, by Andrienko et al. [ALA⁺18]: The primary results are the behavioral model of the subject and the provenance of the model. The secondary results are the externalized model representing the behavioral model and the answers to the subject's questions.

to develop visualizations and VA systems that support knowledge generation and sense-making. To validate if the requirements defined in the design triangle [MA14] are met, I will conduct user studies and design evaluations to ensure the developed techniques are appropriate, effective, and expressive.

1.4 Research Questions

From the above covered problem description and definition of VA I now formulate my research questions that I aim to answer within the course of this thesis.

Main Question. Which VA methods can be found as appropriate to explore and identify DQ issues in time-oriented data leveraging metrics, provenance, and uncertainty?

I want to further specify sub-questions to investigate aspects that have not been resolved so far in scientific literature.

Sub-Question 1. Can DQ metrics be utilized in a data wrangling and cleansing application as **measures of quality for various types of data** to give a visual overview of the **overall** amount of issues as well as a **detailed information** about the errors in the dataset? And how can VA methods be utilized to support **identifying, understanding,** and **correcting quality issues**?

Sub-Question 2. How can **uncertainty** be **quantified from data wrangling and cleansing** and how can it be visualized to assess the influence of the pre-processing steps on downstream analysis?

Sub-Question 3. What kind of **DQ information** can be **stored as data provenance** and used by analysts to **comprehend the history of data wrangling and cleansing steps** and assess the qualitative condition of the dataset to judge the data's usability?

1.5 Structure

The first part of this thesis defines **the Problem**. In this chapter, I introduced the use of visual-interactive systems and motivated the problem statement. This was followed up by discussing the employed research methodology and formulating my research questions. To complement the problem descriptions and research methodology, the Related Work chapter exhaustively covers the state of the art on VA and information visualization research as well as visualization and interactive analysis of DQ assessment, uncertainty, and provenance.

In the second part of the thesis I present **the Proposed Solution**, initially defining the conceptualizations used in the developed approaches, defining notions of DQ, uncertainty in time series pre-processing, and data and insight provenance from DQ. Chapter 4 presents MetricDoc, an approach for creating and customizing DQ metrics to visually explore quality issues of tabular datasets. Chapter 5 leverages these DQ metrics and

captures them alongside pre-processing operations to let users analyze the development of quality during data wrangling. In Chapter 6 I present a new method for quantifying uncertainty from pre-processing operations in multivariate time series (MVTS).

In **the Evaluation & Results** part, the developed designs and techniques are evaluated using different evaluation methods: Chapter 7 shows case studies of applying the developed systems in real-world scenarios, in Chapter 8 an iterative design methodology is presented to iteratively validate visualization design during development of MetricDoc. Chapter 9 presents a user experience evaluation of the provenance analysis approach presented in Chapter 5. Lastly, in Chapter 10 the uncertainty quantification methodology is used to develop and evaluate a visualization design for time series segmentation results.

In **the Conclusion** I discuss the outcome of the proposed solutions and determine if and how the research questions were addressed based on the implications found in the evaluation results. I conclude my thesis by motivating future research directions.

Related Work

In this chapter, I will first introduce the foundations and definitions that the thesis is based on, followed by the state-of-the-art in the relevant fields of DQ, uncertainty, provenance, and time-oriented data analysis.

2.1 Foundation and Definitions

We base our fundamental understanding of quality on DQ research, which includes DQ management and database research. Quality needs to be adequately controlled, either in an organizational, architectural, or a computational level. In the course of this thesis I will cover computational solutions for dealing with DQ. However, the goal is that the presented VA solutions will help analysts better understand and leverage information in the organizational and architectural levels. According to Sadiq [Sad13], the computational solution space covers data record linkage, lineage and provenance, data uncertainty, semantic integrity constraints (which will be referred to as DQ checks), as well as trust and credibility. The emphasis on such diverse methodologies to detect, assess, and improve DQ has historically been covered in visualization and VA research (e.g., data management [CFS⁺06], interactive data wrangling [KPHH11] and cleansing [GAM⁺14]), and until today illustrates that DQ continues to be an interdisciplinary field [LAW⁺18].

Most visualization and VA approaches assume that the input data contains a clean and perfect set of entries. This can not be further from the truth, according to Dasu and Johnson [DJ03] 80 % of the time spent analyzing a dataset constitutes data cleansing. Hence, assessing DQ, and deciding when a dataset is “fit for use”, i.e., free of defects to allow analysis and decision-making, is vital to reduce the time spent with data cleansing.

Pipino et al. [PLW02] also distinguished between **subjective** and **objective assessment**. Hence, defining a general measure to use in a metric is only possible for objective assessment. For subjective assessment, analysts may use custom measures.

2.1.1 Data Quality

According to Dasu [Das13], a measure of DQ is always derived from a data matrix $D = \{d_{ij}\}$, with $i = 1, \dots, N$ representing a row of an entity, and $j = 1, \dots, d$ corresponding to an attribute. While D is the data generated by recording the real-world process P , it might not be faithful to what the analyst's expectation D^* (e.g., due to recording issues, or perception differences). The disparity between D and D^* can be seen as a measure of quality: The less similar D and D^* are, the lower the quality of the dataset. Consequently, if we are able to make D and D^* more similar by cleansing the data, we can improve quality. In a controlled experiment we might know D^* , but in real-world scenarios, it is unknown. Hence, it is not possible to determine whether D comes closer to D^* and still representing P faithfully. It is necessary to make a trade-off between making D similar to D^* and distorting the representation of P . Dasu [Das13] describes data cleansing to be an iterative process that is divided into four stages: (1) *Define and Identify*: establishing a notion of what constitutes a quality issue/data glitch, (2) *Detect and Quantify*: defining functions to detect quality issues, (3) *Clean and Rectify*: selecting adequate cleansing methods, and (4) *Measure and Verify*: quantifying the impact of the applied cleansing methods.

The meaning of DQ is often depending on the context and application domain [DJ03]. Also, Batini et al. [BCFM09] described a *DQ methodology* that bases its measures on input information and the application context. They generalized the methodology into three phases, (1) state reconstruction (optional), (2) assessment/measurement, and (3) improvement. For the *assessment* phase – i.e. measuring quality along relevant quality dimensions, and comparing them to reference values to enable diagnosing – they defined different steps to be followed, namely data analysis, DQ, identification of critical areas, process modeling, and measurement of quality. In the following sections, I will more closely look at formalizations of DQ dimensions, DQ errors, and how DQ metrics and checks can be used to combine generic DQ dimensions with domain-specific constraint validation and error identification.

Data Quality Dimensions

Both Dasu [Das13] and Batini et al. [BCFM09] describe stages or phases where a notion of quality is to be defined prior to being able to assess the quality of a dataset. Initial work by Wang and Strong [WS96] described different dimensions of DQ. Batini et al. condensed a basic set of DQ dimensions from previous literature, accuracy, completeness, consistency, timeliness. According to Redman [Red12], DQ dimensions help find a tangible, formal specification which can be associated with the data model, values, and recording. A DQ dimension captures a specific aspect of quality, it refers to properties of data and their attributes [BS16]. However, such dimension is not intended to be a quantitative measure, but more a qualitative one to describe DQ. Within the thesis I will use the following definitions of DQ dimensions, based on [Das13, BCFM09, BS16, PLW02], and are concerned mostly with data values, as opposed to presenting the data or the conceptual data structure:

Completeness [BCFM09, BS16, Das13]: Completeness defines the degree to which the data values correspond to the real world object, in terms of values, tuples, attributes, or relation. Missing data can correspond to lost values or tuples, due to issues in data collection, processing, or storage.

Uniqueness [Das13, DJ03]: Uniqueness of an entry is violated if there are duplicate key value vectors. The uniqueness constraints are highly context dependent, hence they could be defined by duplicate, distinct, or non-unique values. Specifically, it is highly likely that multiple constraints need to be defined to adequately ensure the uniqueness of a tuple.

Consistency [BCFM09, Das13, TZHH18]: This notion of quality covers violations of pre-defined semantic rules for data values or tuples. If data are created or entered incorrectly, such rules could prevent spurious tuples that would otherwise not be detected. It can be distinguished between intra-relation constraints cover valid ranges for value specific requirements, while inter-relation constraints use information across the data tuple/entry for validation.

Accuracy [BCFM09, TZHH18]: In early publications, accuracy corresponded to the notion of data quality as a whole, e.g., “the extent to which data are correct, reliable, and certified free of error” [WS96, p. 14, Table 1]. In this thesis the quality dimension accuracy is a data value oriented definition, distinguishing between **semantic accuracy**, i.e., the closeness of value v to the true value v_0 , and **syntactic accuracy**, i.e. the closeness of value v to the corresponding value domain’s specific format requirements.

Some other DQ dimensions are specifically defined for certain data domains, which makes them difficult to generalize [BS16]. As a result, it is necessary to not only look at notions or dimensions of quality, but also more generally look at how errors in the data affect these dimensions.

Data Quality Errors – Glitches and Issues

As with different notions, different terms for DQ errors are used throughout literature interchangeably: data glitches [Das13, DJ03], data errors [KCH⁺03, GGAM12, WS96, BCFM09], and (data or information) quality issues [Red12, BS16], data quality problems [ORH05, BG05]. For the remainder of this thesis I will use the term DQ error. According to Dasu and Johnson [DJ03, p. 103], “a data glitch is any change introduced in the data by causes external to the process that generates the data and is different from the usual random noise present in most data sets.” Opposed to random noise, errors introduce systematic changes. According to Kim et al. [KCH⁺03], an error causes a wrong result or does not allow deriving any result due to inherent problems in the data.

Figure 2.1 shows an example of how a an error can occur in the data, and illustrates that we need to make a distinction between the cause of an error and the manifestation of an error in the data (detected in different notion of DQ). What can be seen is that the error was likely caused by the polling interval that had one record being pushed to the next

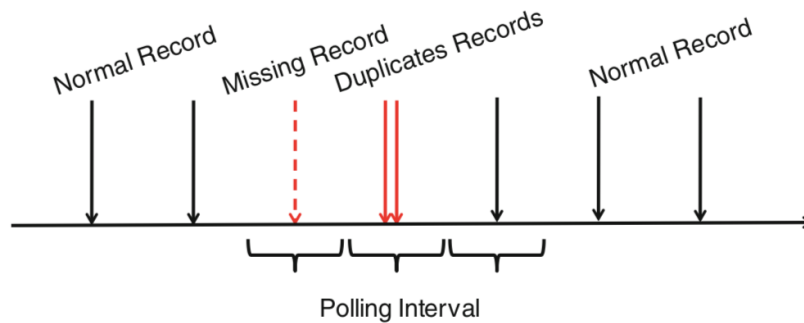


Figure 2.1: Illustration of a quality error that creates a missing record, and a duplicate record in the data, affecting different notions of quality, i.e., completeness and uniqueness.

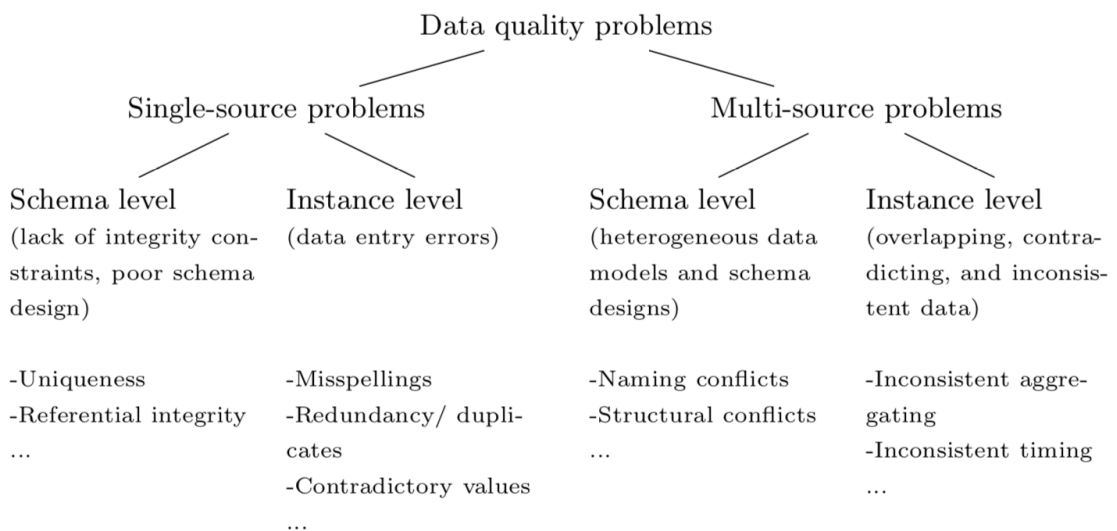


Figure 2.2: Classification of DQ problems based on error source and cause [RD00] (obtained from Gschwandtner et al. [GGAM12]).

interval. An analyst might not know this, and only see the effects in the data, but the goal is not to treat the effects in the data but to gain insights into how to react to the data error appropriately. In this case, the analyst could impute the missing value with the duplicate value stored in the successive polling interval.

There have been considerable efforts for categorizing the types of DQ errors. Rahm and Do [RD00] classified data quality problems based on their sources and their causes (compare Figure 2.2) While some errors are observable easily, others can only be noticed through comprehensive analysis because they might be concealed in local data phenomena. Kim et al. [KCH⁺03] and Li et al. [LPK10] presented general non-formal taxonomies of dirty data to facilitate the correct identification of errors (compare Figure 2.3). Based on a formal definition of DQ errors, Oliveira et al.'s [ORH05] also presented a taxonomy

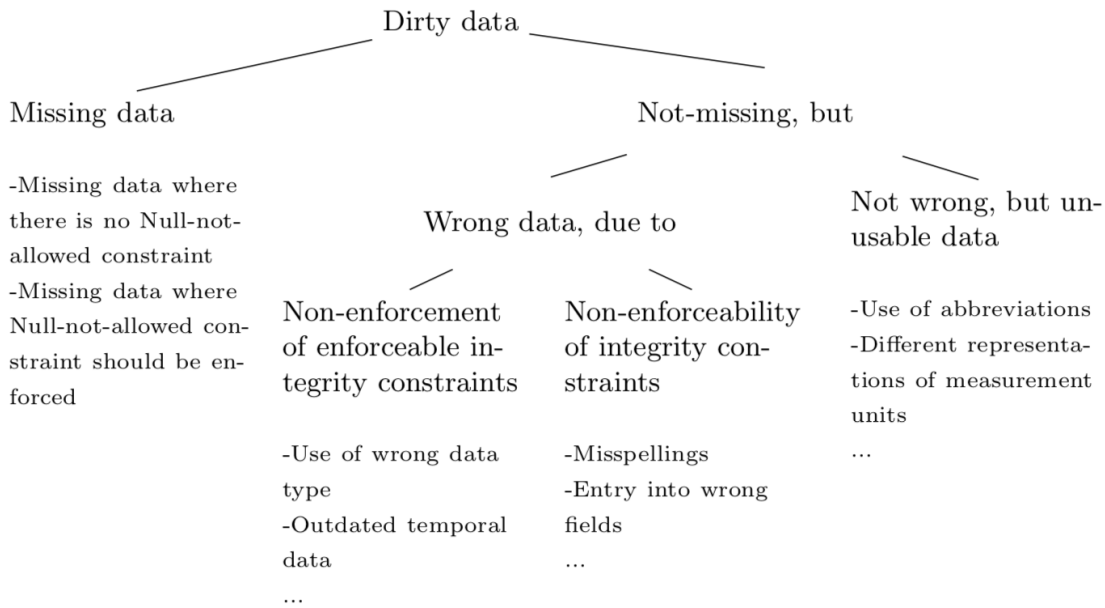


Figure 2.3: Classification of dirty data by type of error [KCH⁺03] (courtesy of Gschwandtner et al. [GGAM12]).

of *DQ problems*, described partly with definitions and partly through natural language. The taxonomy characterizes DQ errors according to levels of the hierarchical data model (compare Figure 2.5). The problem categories distinguish between DQ errors at the lowest (1) attribute/tuple level, including (1a) single attribute of a single tuple, (1b) single attribute in multiple tuples, and (1c) multiple attributes of a single tuple, (2) the level of a single relation, (3) the level of multiple relations, (4) the level of multiple data sources. In other works, Müller and Freytag [MF03] classified *quality criteria*, i.e., DQ dimensions according to our notion, and associated potential *data anomalies*, i.e., DQ errors. Li et al. [LPK10] juxtaposed the taxonomy of *DQ problems* by Oliveira et al. [ORH05] with established DQ rules [AMA05] (see Table 2.1b) and dimensions (see Table 2.1a).

Gschwandtner et al. [GGAM12] made an effort to compare the various DQ error taxonomies for completeness and error specificity. Figure 2.4 shows the comprehensiveness of the taxonomies, grouped by source and type. It shows that the taxonomies cover the problem space extensively, but all taxonomies fall short of encompassing all DQ errors. In their paper, Gschwandtner et al. furthermore extensively characterized different types of time-oriented data (extending definitions from Aigner et al. [AMST11]) and present a taxonomy for dirty data for time-oriented data, which accounts for the specificity of this data and application domain.

2. RELATED WORK

	Rahm & Do, 2000 [1]	Kim et al., 2003 [2]	Müller & Freytag, 2003 [3]	Oliveira et al., 2005 [4]	Barateiro et al. 2005 [5]
Single source					
Missing data	●	●	●	●	●
Missing value	●	●	●	●	●
Missing tuple	○	○	○	○	○
Semi-empty tuple	○	○	○	○	○
Dummy entry (e.g., -999)				●	
Syntax violation / wrong data type	●	●	●	●	●
Duplicates	●	●	●	●	●
Inconsistent duplicates / Contradicting records	●	●	●	●	●
Approximate duplicates	○	○	○	○	○
Unique value violation	●	●	●	●	●
Incorrect values	●	●	●	●	●
Misspellings	●	●	●	●	●
Domain violation (outside domain range)	●	●	●	●	●
Violation of functional dependency (e.g., age-birth)	●	●	●	●	●
Circularity in a self-relationship	○	○	○	○	○
Incorrect derived values (error in computing data)	○	●	○	○	○
Unexpected low/high values			●		
Misfielded values	●	●	●	●	●
Invalid substring / Embedded values	●	●	●	●	●
Ambiguous data; imprecise, cryptic values, abbreviations	●	●	●	●	●
Outdated temporal data		●		●	
Inconsistent spatial data (e.g., incomplete shape)		●			
Multiple sources					
References	●	●	●	●	●
Referential integrity violation / dangling data	●	●	●	●	●
Incorrect references	●	●	●	●	●
Heterogeneity of representations	●	●	●	●	●
Naming conflicts	●	●	●	●	●
Synonyms	●	●	●	●	●
Homonyms	●	●	●	●	●
Heterogeneity of syntaxes	●	●	●	●	●
Different word orderings	●	●	●	●	●
Uses of special characters	○	●	○	○	○
Heterogeneity of semantics	●	●	●	●	●
Heterogeneity of measure units (EUR vs. \$)	●	●	●	●	●
Heterogeneity of aggregation/abstraction	●	●	●	●	●
Information refers to different points in time	●	●	●	●	●
Heterogeneity of encoding formats (ASCII, EBCDIC, etc.)		●			●

Figure 2.4: Comparison of taxonomies of general data quality problems by Gschwandtner et al. [GGAM12] (●... included in taxonomy, ○... further refinement covered in parent problem).

Data Quality Metrics

So far, we have defined DQ dimensions to be a qualitative measure of quality and DQ errors to be the cause of undesired effects in the data. However, the dimensions only serve as guidelines to what kinds of error are relevant to be detected in a particular dataset. As to how the quality is interpreted for a specific use case depends on context and domain knowledge for correctly identifying errors [DJ03, BCFM09]. For tabular and structured datasets, metrics are used as probes that validate properties of a DQ dimension to provide a quantitative measure of particular error characteristics [BS16]. As such, they allow measuring the quality of a dataset. Dasu [Das13] distinguishes DQ metrics between **constraint-based metrics**, i.e., validation schemata based on data properties and functional dependencies, and **quantitative metrics**, based on statistical measures and data mining methods. Pipino et al. [PLW02] also differentiate between **task-independent** and **task-dependent metrics**: (1) task-independent metrics reflect data quality without contextual specificity and can be applied to any datasets, (2) task-dependent metrics include specific application contexts and domain constraints, to ensure format rules or other conventions. Particularly for tabular datasets, metrics can be calculated **column-wise**, **tuple-wise**, and **entry-wise**.

How a metric is measured, depends on the application scenario they are used in. First

Rule No.	Dirty data type No.
R1.1	N/A
R1.2	DT.21
R1.3	N/A
R1.4	DT.1, DT.15
R2.1	DT.4
R2.2	DT.5
R2.3	DT.11, DT.14, DT.17, DT.20, DT.26, DT.35
R2.4	N/A
R2.5	DT.19, DT.34
R3.1	DT.16, DT.24, DT.25
R3.2	DT.3, DT.22
R4.1	DT.8

(a) Adelman et al.'s DQ rules associated with DQ problem taxonomy from Oliveira et al.

DQ dimension	Dirty data type No.
Accuracy	DT.2, DT.4~DT.9, DT.11, DT.14, DT.16, DT.17, DT.19, DT.20, DT.23~DT.26, DT.34, DT.35
Completeness	DT.1, DT.15, DT.21
Currentness	DT.3, DT.22
Consistency	DT.10, DT.13, DT.23, DT.27~DT.32
Uniqueness	DT.12, DT.18, DT.33

(b) DQ dimensions associated with Oliveira et al.'s DQ problem taxonomy.

DQ dimension	Rule No.
Accuracy	R2.1~R2.5, R3.1, R4.1 R4.5
Completeness	R1.2, R1.4
Currentness	R3.2
Consistency	R5.5, R6.1, R6.2
Uniqueness	R5.1, R5.2

(c) Adelman et al.'s DQ rules associated with DQ dimensions.

Table 2.1: Li et al.'s [LPK10] taxonomy of dirty data associated DQ rules [AMA05], DQ problems [ORH05] and DQ dimensions. The rules are constituted of: R1 *business entity rules*, R2 *business attribute rules*, R3 *data dependency rules*, and R4 *data validity rules*, R5 *duplicate record validity*. The used DQ dimensions are accuracy, completeness, currentness, consistency, and uniqueness.

and foremost, generic measures that are independent of dataset dimensions help develop comparable results for assessing quality across datasets. Pipino et al. [PLW02] proposed different measures for quality metrics: (1) **Simple Ratio**, the number of *undesirable outcomes* divided by *total outcomes* subtracted from 1. An *undesirable outcome* is defined by a set of defined criteria or validation functions to accurately describe a DQ dimension. (2) **Min or max operators**, a measure for aggregating multiple dimensions, e.g., using a minimum or maximum measure to represent the best or worst DQ metric from a group of metrics representing a DQ dimension. (3) **Weighted average** measure, i.e., adding weighting factors to DQ metrics.

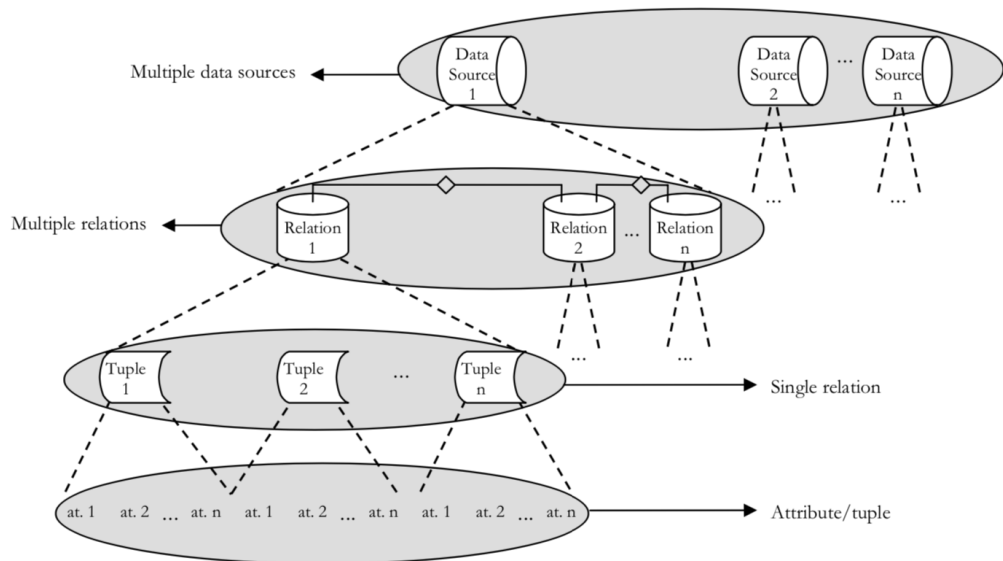


Figure 2.5: Organizational data model that illustrates the different sources of DQ errors, according to Oliveira et al. [ORH05].

Developing task-dependent DQ metrics requires awareness of multiple characteristics and domain-specific properties of the data. Analysts can refer to various DQ error taxonomies, with formal or semi-formal definitions of such errors could be formalized to serve as DQ metric validation functions. These taxonomies help determining appropriate types of metrics for identifying specific DQ error patterns. Developing metrics for such specific error types requires knowledge of the application domain. Li et al.’s [LPK10] juxtaposition of DQ errors, dimensions, and rules can help build a set of compound metrics to have an accurate qualitative measure of a DQ dimension. Data quality checks can be added to metrics to integrate constraints. Knowing (1) which DQ rules need to be satisfied, (2) which errors occur and need to be accounted for in downstream analysis, (3) which quantification method to use for the employed metrics, and (4) which DQ dimension they adhere to, lets the analyst develop a comprehensive set of DQ dimensions to assess quality.

In the following section, I will investigate how the classifications of DQ dimensions, errors, and metrics can be used for identifying, correcting and annotating data.

2.1.2 Data Cleansing, Profiling, and Wrangling

Initially, efforts for improving DQ were coined under the term of data cleansing/cleansing. Rahm and Do described data cleansing as “detecting and removing errors and inconsistencies from data in order to improve the quality of data. (...) to provide access to accurate and consistent data (...)” [RD00, p. 1]. Müller and Freytag [MF03] described problems, methods, and challenges in data cleansing, which was founded on many impor-

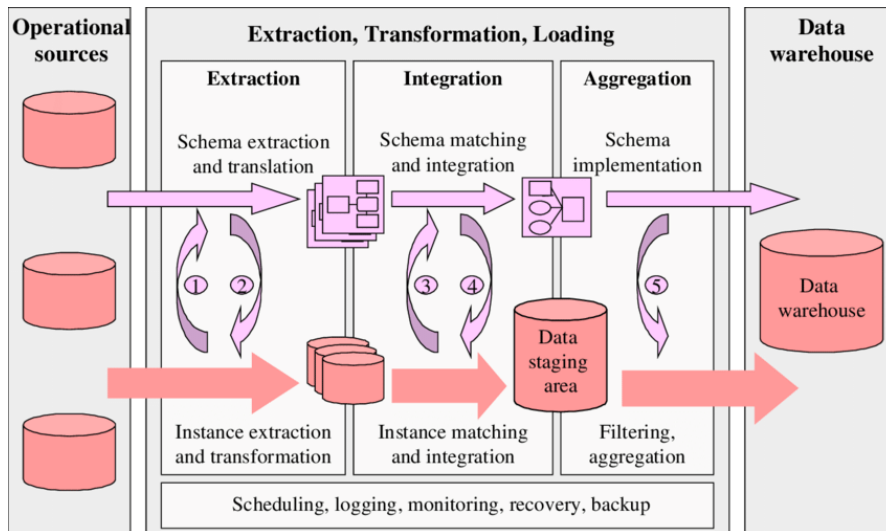


Figure 2.6: Steps of building a data warehouse: the ETL process.

tant preliminary works [Red98, RH01, GFSS00]. These early approaches were mainly concerned with data warehousing and database management, and attempted to provide integration of DQ assessment into the entire data processing workflow. Figure 2.6 exhibits Rahm and Do’s *ETL* workflow, i.e., extraction, transformation, loading. In the extraction and integration stage, DQ assessment is performed, in the *data staging area* in between the integration and aggregation stages correspond to the process where data cleansing is performed. Van den Broeck et al. [BCEH05] defined data cleansing as the iterative process of screening, diagnosing, and editing data. Dasu and Johnson’s separated these processes and described the notions of exploratory data mining and data cleansing as automated methods to help understand the data, by applying exploratory data mining, and help ensure DQ, by conducting automatic data cleansing and DQ metric computation. They propose multiple techniques and algorithms for cleansing data on an entry level (compare Figure 2.5) e.g., missing value imputation, outlier detection, and on the data source level for relational databases, e.g., approximate matching, functional dependencies, field value classification.

A first overview of DQ applications was conducted by Barateiro and Galhardas [BG05], who identified six different categories of tools: (i) data profiling; (ii) data analysis; (iii) data transformation; (iv) data cleansing; (v) duplicate elimination and (vi) data enrichment. These categories were further condensed to represent the troika of DQ assessment: (i) Data Cleansing, (ii) Data Profiling, and (iii) Data Wrangling.

Data cleansing, profiling, and wrangling are iterative processes, with the intention to produce a usable dataset without exacerbating the time spent on pre-processing. In the past sections I have extensively grounded DQ assessment in the fields of data warehousing and database management. However, in this section we can see that interest has increased to develop visual and interactive methods for assessing DQ. This especially holds true

for data cleansing, profiling, and wrangling. However, it is first necessary to define and clearly distinguish these disciplines:

Data Cleansing [MF03, GAM⁺14, RD00, BCEH05, Hel08] is the process of identifying and correcting DQ errors in data. Often, domain specific algorithms and methods are developed to improve DQ. Cleansing is performed on different levels of the organizational data model to identify and resolve errors on single-source and multi-source data instances, as well as on a schema/metadata level.

Data Profiling [Das13, ORH05, BG05, SNHS17] describes the identification and communication of DQ errors. Means of measuring DQ can be used to get a conceptual model of quality. On the one hand, DQ metrics can be utilized to detect errors (compare 2.1.1). Descriptive statistics or other means of overview allow users to identify errors in the data, or assess the overall usability of a dataset, e.g., using a measure of confidence in the data [Gol13]. It is particularly important in conjunction with cleansing, to validate if errors could be resolved, and if DQ improved. On the other hand, DQ metrics are employed as overview measures that give analysts simple measures of quality without requiring detailed inspection.

Data Wrangling [KHP⁺11, KPHH11, BG05] – often only referred to as data transformation – is defined as an iterative data exploration and transformation process to enable analysis. Wrangling represents the effort to make sure the data are *credible*, *usable*, and *useful* [KHP⁺11]. While in data cleansing processes the analyst ensures that DQ is improved, the emphasis during data wrangling is that entries can be used in a meaningful way in downstream analysis. This usually consists of a continuous refinement loop of transforming data and using analytical methods (e.g., data profiling) to ensure the data has not been skewed and the data is still representative after wrangling has been performed.

These processes are clearly distinct in their defined goals and the methods used. However, in real-world scenarios, analysts often find it necessary to swap between tools to perform dedicated operations (compare Figure 2.7). The result is that tools start to intermingle functionality from data cleansing, profiling, and wrangling for particular application domains, to form a general DQ assessment workflow, e.g., Ajax by Galhardas et al. [GFSS00], TimeCleanser from Gschwandtner et al. [GAM⁺14]. What can be observed from these works is that already in early stages of research interactive methods were used to facilitate detecting and cleansing errors and inconsistencies.

2.1.3 Uncertainty

Despite the rapid growth of data generated and new means for generating data, uncertainty is still prevalent in multiple fields, e.g., medical imaging, geo-sciences, weather and climate research, due to measurement inaccuracy, or sensor deficiencies. Generally, **uncertainty** is defined as **the lack of information** [BHJ⁺14], more specifically the “degree to which the lack of knowledge about the amount of error is responsible for hesitancy in accepting results and observations without caution” [HGR94, p. 368]. Data analysis is influenced

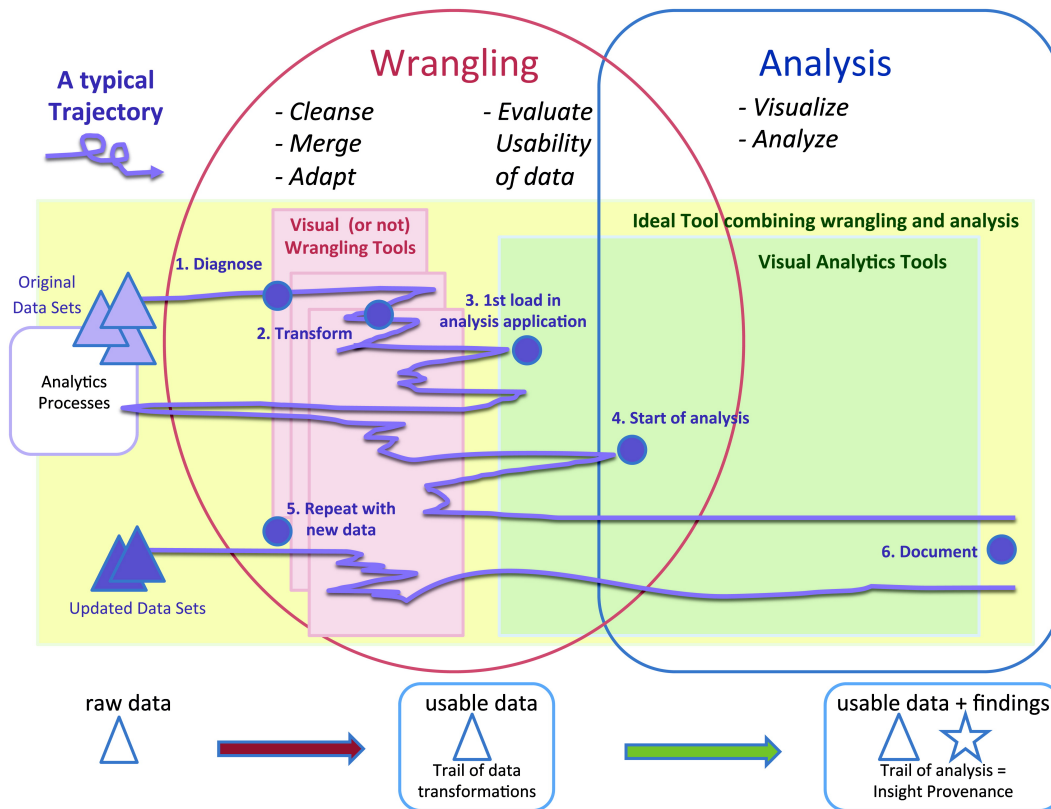


Figure 2.7: Data wrangling and analysis is seen as an iterative process. Feedback loops can lead analysts back and forth between analysis and wrangling. This illustration shows the close connection between data wrangling, data profiling, and actual analysis. [KHP⁺11]

by uncertainty, albeit implicitly, i.e., there is no uncertainty encoded in the data, but it cannot be assumed that the data are absolutely true, or explicitly, i.e., uncertainty is explicitly considered in the analysis scenario in any form. Analysts have to consider the confidence and trust in the available information, their own judgment, and experience for subsequent decision-making. We can distinguish different phenomena that cause that concrete lack of information [BHJ⁺14, AK06]: (1) aleatoric/aleatory and (2) epistemic causes of uncertainty. Aleatoric/aleatory uncertainty describes results being created by change due to actual real-world phenomena. These results can never be measured more accurately and can be modeled using probabilistic approaches. Epistemic uncertainty describes that in principle could be known, but in practice is not. Such results are affected by errors that cannot be controlled and as such can be described by non-probabilistic modeling. Measures of uncertainty can be represented in different ways, ranging from single numeric values annotating the original data, or being an integral descriptive function of the data. These aspects will be discussed in the upcoming section.

Independently, uncertainty is often specific to application domains, hence, the usefulness of

uncertainty for analysis and correspondingly the measures of uncertainty vary depending on the goal of analysis. For example, in geospatial data analysis, the uncertainty is traced back to communicate issues in DQ. Uncertainty can also be utilized for risk analysis, e.g., prediction models [AK06]. In the following, I will demonstrate different sources of uncertainty.

Sources of Uncertainty

It is necessary to distinguish between uncertainty associated with information/data and uncertainty associated with the analysis process itself [THM⁺05], because analysts need to respond to uncertainty they can or cannot control differently. Within the context of visualization and VA, uncertainty introduced during the analytical process itself needs to be communicated accordingly. To accomplish that, we need a typology of uncertainty sources. Griethe and Schumann [GS06a] described that uncertainty is influenced by errors, imprecisions, accuracy (e.g., size of interval containing values), lineage, subjectivity, non-specificity, and noise in the data. Similarly, Thomson et al. [THM⁺05] presented a typology categories of uncertainty for geospatial data. Their quantitative representations of uncertainty show the close relation between uncertainty and DQ. Bonneau et al. [BHJ⁺14] consolidated these different typologies and classifications into the following sources:

1. **Uncertainty inherent in the sampled data.** Similar to what was described in [GS06a], collecting or sampling data can introduce insufficient, superfluous, or spurious information. By applying imputation, or other estimation techniques, the margin of potential error can be quantified, to measure the confidence of a data value. Here we can see that uncertainty from data sampling is closely related to data quality assessment (also noted by [Che13]), which can be leveraged to minimized the error introduced. Additionally, knowing the data source, and associated metadata, could lead the analyst to also assert a certain amount of uncertainty to the data.

2. **Uncertainty generated by models or simulations alongside the data.** Employing computational models in an analysis workflow is another source of uncertainty. This uncertainty is associated with the variability of the modeling, being caused by *variability from simplifying abstractions, in mechanism or magnitude of causality and relationships, potential error in model inputs, incorrect model parameters, and imprecision in tacit knowledge incorporated in the model*. As such, the output of a model can include the estimated error or accuracy of the result, or the confidence in a qualitative/categorical prediction.

3. **Uncertainty introduced by the analysis and visualization process.** In downstream analysis processes, uncertainty affects propagation, magnification, perception, and overall analysis of data. As a result, uncertainty needs to be carefully explored and adequately integrated using appropriate analysis methods and visualization techniques.

These sources of uncertainty require adequate methods to be adequately measured. I will discuss means of quantifying uncertainty in the upcoming section.

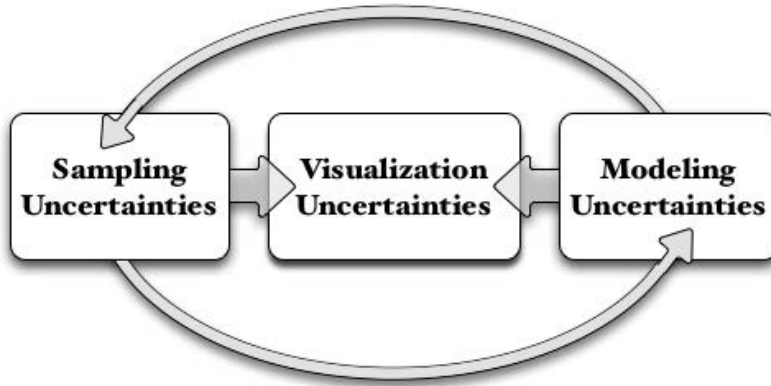


Figure 2.8: Sources of uncertainty and how they affect each other subsequently. [BHJ⁺14]

Quantifying and Measuring Uncertainty

Modeling uncertainty is the basis of identifying uncertainty associated with data. It is necessary to estimate the influence randomness and how it influences the data. Knowing the system and the associated uncertainties, allows analysts to model the effects more appropriately. As such, Ayyub and Klir [AK06] classified uncertainty theories used for modeling: (1) classical probability theory, (2) probability theory based on fuzzy events, (3) classical possibility theory, (4) theory of graded possibilities, (5) Dempster–Shafer theory (DST) of evidence, (6) fuzzified Dempster–Shafer theory of evidence, (7) theory based on feasible interval-valued probability distributions (FIPDs), (8) fuzzified FIPD, and (9) Other uncertainty theories They argue that some theories can be more appropriate for modeling particular uncertainty properties, e.g., set theory can deal with ambiguity, probability and statistical theories can be used for modeling randomness and sampling uncertainty. Furthermore, they define a measure of uncertainty as a function (u) that assigns to each representation of evidence measured as a *classical measure* C (C is a nonempty family of subsets of the universal set of observations X) in the theory a nonnegative real number. Their definition of an uncertainty measure is a function mapping the set U of all uncertainty functions μ to a number R_+ :

$$u : U(\mu) \rightarrow R_+$$

with

$$\mu : C \rightarrow R_+ = [0, \infty]$$

Thus, if multiple uncertainty measures are employed for a system, the uncertainty of the combined set of uncertainty measures should still calculate a real number. This is only true for functions that satisfy the following requirements, according to information theory [KS01]: *subadditivity*, *additivity*, *range*, *continuity*, *expansibility*, *branching/consistency*, and for some theories of uncertainty *monotonocity* and *coordinate invariance*.

Within the extent of this thesis, the uncertainties that will be investigated originate from data quality issues and data pre-processing algorithms. Both these uncertainties can be modeled by classical probability theory. Hence, I adopt the formalization of uncertainty in probability theory from [BHJ⁺14]:. Uncertainty and randomness is examined in the probability space (Ω, F, P) . The probability event space Ω is comprised of all possible outcomes of a random event A . F represents all possible outcomes. The definition of uncertainty is based on the probability measure P with the following principles:

1. $0 \leq P(A) \leq 1$, for any $A \in F$, i.e., the probability of an event A is between 0 and 1,
2. $P(\Omega) = 1$, i.e., the probability of all possible events adds up to 1,
3. For $A_1, A_2, \dots \in F$ and $A_i \cap A_j = \emptyset$, for any $i \neq j$ i.e.,

$$P\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} P(A_i).$$

The resulting uncertainty measure can be represented as a **probability density function (PDF)**, as **multi-value data**, as **bounded data** [BAL12]. Potter et al. [PRJ12] investigated the use of uncertainty measures in visualization and classified them based on the uncertainty dimensionality, differentiating between **scalar**, **vector**, or **tensor field** representations. For visualization and VA the dimensions of the uncertainty representations are combined with the data dimensions. The information encoded in different field types can be conveyed in various ways, depending on the uncertainty measure at hand, and might need to be abstracted further to be effectively combined with the data dimension:

Uncertainty as a **PDF** can be expressed with multiple characteristics (e.g., mean, skew, tail, compare Figure 2.9), to most accurately describe the distribution, or only with single values (statistically grounded like averages and standard deviation, or ungrounded like estimates) that are visually represented. Other usages include changing data values for downstream visualization, e.g., modulating information properties. **Bounded data** represent data values that represent possible intervals instead of actual values in the data dimension, which can be described with means of ambiguity. **PDFs** can be also aggregated to map uncertainty to the data dimension. Multivariate data and uncertainty dimensions can be mapped accordingly, mapping individual uncertainty (e.g., a PDF) to the corresponding data dimension.

2.1.4 Provenance

With computing and data analysis often being enabled by data pre-processing and applying workflow pipelines, data management is vital to make the steps taken and operations applied reproducible. By generating and recording analytic provenance (subsequently referred to as provenance), the analyst is able to maintain the ability to revisit previous stages of the analysis and determine ownership, modification history, applied processes, and their impact thereof [DF08].

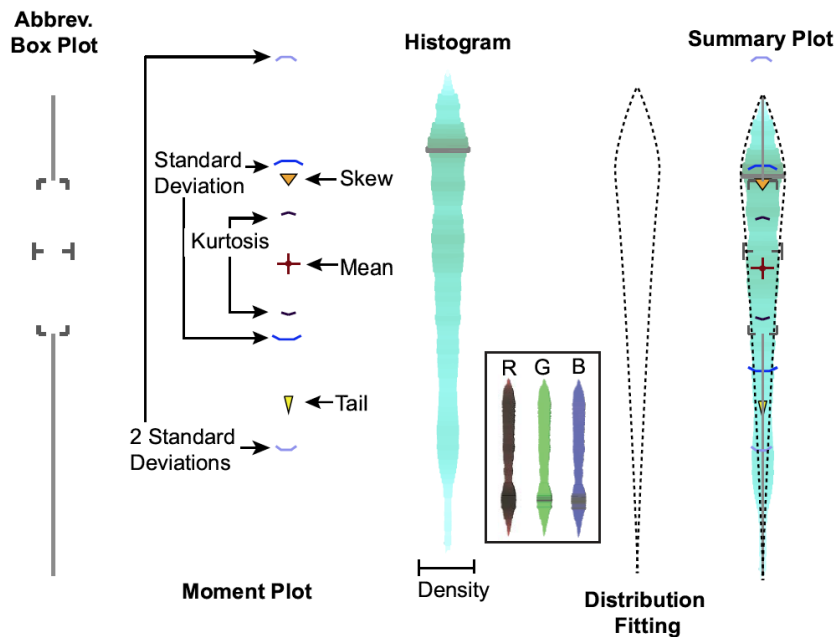


Figure 2.9: Summary plot: Different PDF characteristics are used to describe the distribution [PKRJ10].

Definition of Provenance There are multiple definitions of provenance, depending on the data domain and on the application scenario it is used in. Freire et al. [FKSS08] defined computational provenance as a sequence of steps that led to a result including the chain of reasoning used in its production. It is stored to verify that the sequence of steps used acceptable procedures, inspect the used inputs and parameters, and possibly reproduce the result. I will more closely specify two types of provenance that will be investigated in this thesis: (I) *data provenance* and (II) *analytic provenance*. *Data provenance* is defined as the “description of the origins of a piece of data and the process by which it arrived in a database.” [BKT01, p. 1] *Analytic provenance* captures user’s interactions with a visual interface to retrieve user’s reasoning processes [NCE⁺11]. I use the following general definition from Ragan et al.: Provenance “includes consideration for the history of changes and advances throughout the analysis process” [RESC16, p. 1].

Provenance Types

Provenance is used in different domains and for different purposes that I will go into detail about within the context of visualization and VA research in the upcoming section. Simmhan et al. [SPG05] first created a taxonomy of data provenance in computational science and discussed early methods of visualizing provenance. They investigated the specific uses of data provenance, classifying applications for: Data quality, audit trails, replication recipes, attribution, and general information and context. These early approaches resort to self-developed protocols and solutions for storing and managing

provenance. First popular uses of provenance in visualization and VA research tackled workflow management [CFS⁺06] with VisTrails, proposing a general system for storing such provenance, and interaction histories [GS06b]. For workflow management provenance, Freire et al. [FKSS08] distinguished between *prospective provenance*, i.e., capturing computational tasks' specifications, and *retrospective provenance*, i.e., steps executed and used objects and actors of the system. These leverage terminologies closely related to **data provenance**. Glavic et al. [GDK⁺07] differentiate between the *transformation data provenance*, i.e. provenance of a data item and the processes that lead to its creation, and *source data provenance*, i.e., provenance of the source from which the data item is derived from.

Ragan et al. [RESC16] presented an organizational framework to characterize different types and purposes of provenance. Figure 2.10 shows this framework, differentiating types of provenance between (i) **data**, (ii) **visualization**, (iii) **interaction**, (iv) **insight**, and (v) **rationale**. The purposes of how provenance is used are (i) **recall**, (ii) **replication**, (iii) **action recovery**, (iv) **collaborative communication**, (v) **presentation**, (vi) **meta-analysis**. I will describe these types of provenance in more detail.

Data provenance describes the history of data changes, which includes how the data was captured or sampled, which formats were used, what transformations were applied to them, or if they were derived from other data, which introduces versioning and ownership characteristics. It is also often associated with uncertainty and data quality aspects.

Visualization provenance traces how graphical representations were achieved, which is closely related to data provenance [SSK⁺16]. However, here also the use of visual encodings and interactions is relevant to achieve reconstruction as close as possible.

Interaction provenance records the actions and commands performed by a user, which can be captured on different levels of granularity [GZ09]. It can be distinguished between implicit and explicit interactions.

Insight provenance captures users' hypotheses, insights and analytic findings during exploration and inference. For now, it must be captured outside the grasp of systems because it only observable by prompting users to give extra information.

Rationale provenance describes the reasoning behind decisions, hypotheses, and interactions. It goes beyond intents and its goal is to determine the complete record of reasoning elucidated by user behavior.

Capturing and Storing Provenance

It is necessary to determine how these different types of provenance are captured and stored. As previously mentioned, early works on computational provenance research resorted to proprietary capturing methods and storage models. No open systems were available for using external, generic tools for collecting, representing, storing, and querying provenance. Simmhan et al. [SPG05] motivated efforts to standardize provenance capture

Types of Provenance Information	
Data	The history of changes and movement of data, which can include subsetting, data merging, formatting, transformations, or execution of a simulation to ingest or generate new data
Visualization	The history of graphical views and visualization states
Interaction	The history of user actions and commands with a system
Insight	The history of cognitive outcomes and information derived from the analysis process, including analytic findings and hypotheses
Rationale	The history of reasoning and intentions behind decisions, hypotheses, and interactions
Purposes for Provenance	
Recall	Maintaining or recovering memory and awareness of the current and previous states of the analysis
Replication	Reproducing the steps or workflow of a previous analysis
Action recovery	Maintaining the action history that allows undo/redo operations and branching actions during analysis
Collaborative communication	Communicating and sharing data, information, and ideas with others who are conducting the same analysis
Presentation	Communicating the insights or progression of the analysis with those who are not directly involved with the analysis themselves, such as general public, upper levels of management, or analysts focusing on other areas
Meta-analysis	Reviewing the analytic processes themselves in order to understand and improve aspects of the analysis (such as process efficiency, training efficiency, or analytic strategies)

Figure 2.10: Types and purposes of provenance information, according to Ragan et al. [RESC16]

and storage benefiting interoperability, allowing provenance collection in a centralized way or through middle-ware. Glavic et al. [GDK⁺07] made an effort to generalize a provenance model (compare Figure 2.11) and models for provenance storage, recording, and querying. The model describes functionalities to be supported by provenance systems, and how provenance capture and storage can be achieved. In the model they also include the world model, describing closed or open world provenance, which defines if the system itself controls provenance capture, i.e., *closed world model*, or if the provenance stored is externally generated, i.e., *open world model*. Furthermore, Glavic et al. described strategies for storage and recording (compare Figure 2.12), with the storage strategy describing *no-coupling*, *tight-coupling*, and *loose-coupling* relationships between the provenance data and the source data. The recording strategy also specifies the timing dependency provenance is recorded at: *user controlled recording*, *eager recording*, *no recording* and *system controlled recording*

Moreau et al. [MFF⁺08, MCF⁺11] proposed a standardized provenance storage model,

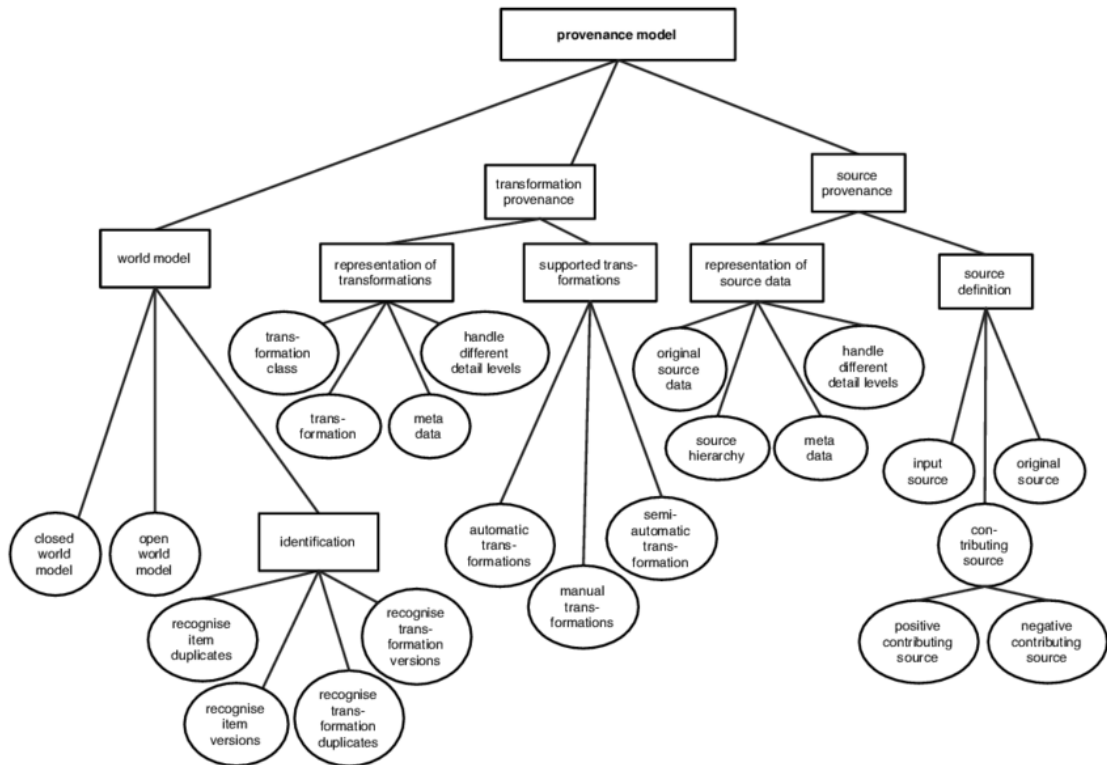


Figure 2.11: The provenance model by Glavic et al. [RESC16] differentiates transformation and data provenance. The world model defines if the provenance system controls the transformation and data items, or if they are externally generated and there is an uncertainty associated if all actions were actually recorded.

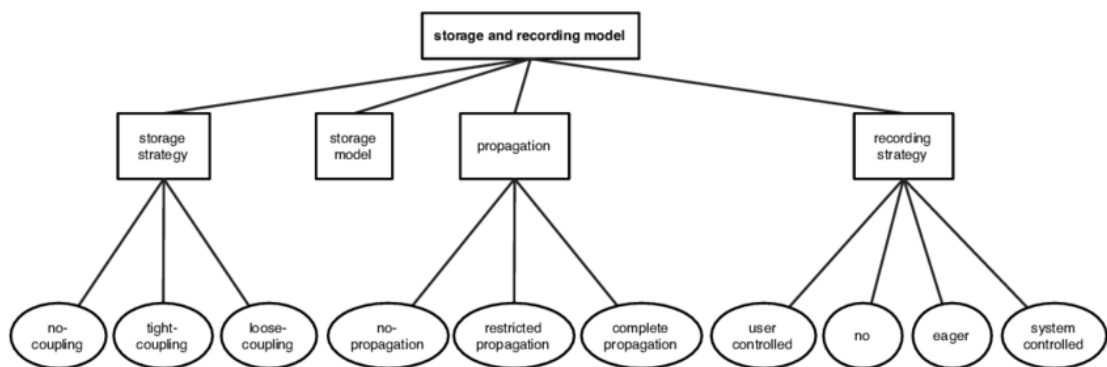


Figure 2.12: Provenance storage and recording model by Glavic et al. [RESC16]

the *Open Provenance Model*. Their intention was to design a model that (1) allows provenance exchange between systems, based on a shared model, (2) can be used to operate tools based on the model, (3) is technology agnostic, (4) can be presented in a generic way, (5) supports multiple layers of information abstraction, and (6) serve as a basis for valid provenance inferences. What can be observed in this model is that the model describes dependencies between *artifacts*, *processes*, and *agents*. *Artifacts* and *processes* can be interpreted as *transformation* and *data provenance* types from Glavic et al. [GDK⁺07], which shows that both types are necessary to adequately describe provenance of a system. Hartig [Har09] proposed a *provenance vocabulary* that exhibits a profile of the generic provenance model, which serves as an example descriptor of provenance for linked data in the web.

In Ragan et al.'s [RESC16] characterization of provenance in visualization and data analysis, they also classify which information is captured in the different types of provenance: **Data provenance** capture is complex, it is supposed to log information on data creation and actions applied to the data (e.g., processing workflows). **Visualization provenance** can be recorded by storing screen shots of the output visualization intermittently, or the states and parameters/settings that led to a particular visual representation. **Interaction provenance** captures user actions taken through system logs. A challenge in capturing interactions is at with granularity the provenance is recorded to most appropriately represent the interactions. **Insight provenance** is often generated by letting users input their experiences while working/interacting with a system. This can be done by annotating different aspects of the system. Another way could be using external protocoling (e.g., thinking-aloud protocols, eye-tracking software). To capture **rationality provenance**, the system designers need to compile different types of provenance to infer the reasoning behind certain actions (e.g., inferring reasoning from interaction logs).

2.1.5 Time and Time-Oriented Data

When capturing and analyzing time and time-oriented data, it is important to appropriately model time depending on the particular task and the problem at hand. Aigner et al. [AMST11] presented important considerations for modeling time (compare Figure 2.13). They distinguish between modeling *time*, the *data*, and the relation of *data and time*. The design aspects of modeling time are **scope**, **scale**, **arrangement**, and **viewpoint**. For appropriate modeling, abstractions of **granularity (and calendars)**, **time primitives**, and **determinacy** must be specified. **Granularities** of time are mappings of time values to larger or smaller conceptual units. If multiple granularities are formed in a hierarchical dependency, they can be categorized into a **calendar**. This includes mapping between pairs of granularities. Aigner et al. describe **time primitives** as an intermediary layer between data elements and the time domain. They distinguish between anchored and unanchored primitives. Combined with instant, interval, and span types, different characteristics of time can be modeled. Lastly, **determinacy** dictates if uncertainty needs to be accounted for. For example, if there is no complete knowledge of all temporal aspects available, e.g., imprecise events, indeterminacy is introduced and

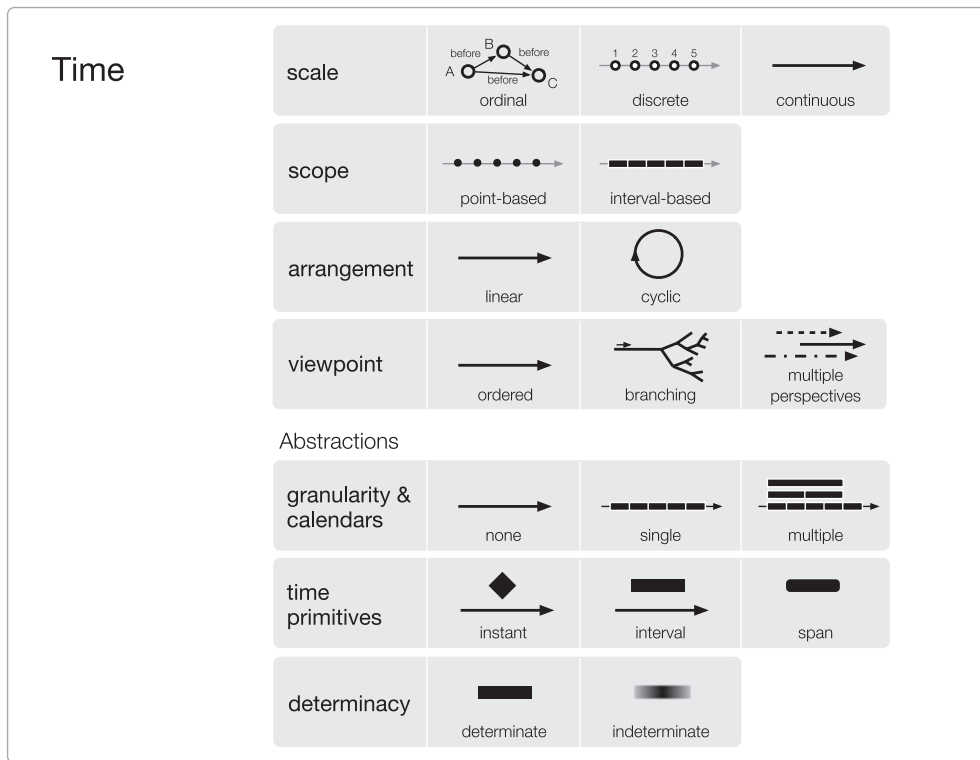


Figure 2.13: Design aspects of modeling time in time-oriented data by Aigner et al. [AMST11]

must be addressed when modeling time. In Figures 2.14, the aspects of data modeling and relating data and time are shown that span the design space available for mapping time to data.

In another effort to defining time similar to time primitives, Gschwandtner et al. [GAM⁺14, GGAM12] added various notions to further describe time. **Rasters** being “a fragmentation of time without gaps consisting of raster intervals,” [GGAM12, p. 65]. **Intervals** are denoted by two points in time, the beginning and end. They classified types of time-oriented data as: (i) **Non-rastered points in time**, (ii) **non-rastered intervals**, (iii) **rastered points in time**, and (iv) **rastered intervals**.

So far, I have discussed definitions and characterizations of the main topics of this thesis: **data quality**, **uncertainty**, **provenance**, and **time-oriented data**. In the upcoming sections, I will deliberate on approaches and research of these fields of research with a particular focus on visualization and VA.

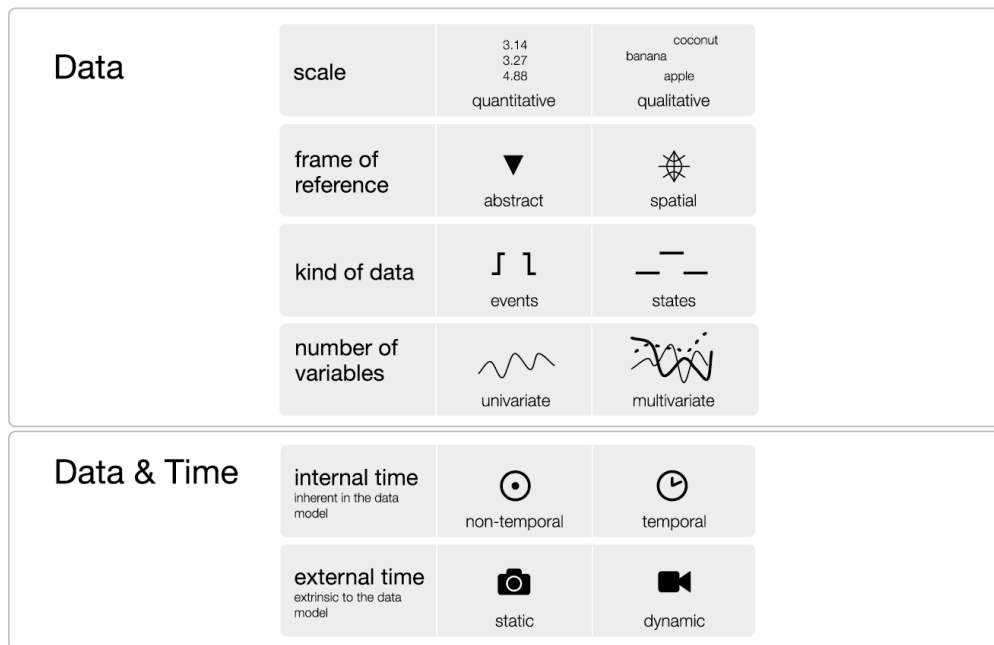


Figure 2.14: Design aspects for modeling data, and time in time-oriented data by Aigner et al. [AMST11].

Visual Analytics Methods for Data Quality and Assessment

In Section 2.1.1 I have mentioned that assessing and improving DQ is an iterative process that requires cleansing, profiling and wrangling new data sets. Ultimately, deciding on the usability of a data set at hand is up to the analyst’s judgment. The proposition is that (I) visual methods and interactive interfaces can help combining automated methods with analysts’ expertise [KHP⁺11]. However, visualization and VA methods often assume that the input data used in the system is pristine and in a perfect condition to perform analysis. It is necessary to (II) communicate to users that potential issues persist in the data, because otherwise they might be conducting analysis and making decisions based on dirty data. Ward et al. [WXYR11] proposed a design methodology for quality-aware visualization, which involves (1) designing and implementing DQ metrics, (2) developing customized display techniques for conveying quality, (3) compare the measures to analysts’ perceived quality, (4) allow interactively changing quality aspects, and (5) developing automatic methods for enhancing DQ. In the upcoming sections I will present works on how to present DQ, interactive methods for data cleansing, profiling, and wrangling, and analytic methods for DQ assessment.

2.2 Visual Encodings of Data Quality

Visualizing DQ varies greatly on the task and the data domain. Correll et al. [CLKS18] proposed visual methods as means for sanity checks of univariate data. Visual-interactive data profiling methods leverage summary visualizations to support assessing detected data anomalies [KPP⁺12]. The used visualization is dictated by the type of data that is subject of analysis. To profile data and assess DQ, **raw**, **aggregated or transformed data**, **DQ measures**, or a combination thereof can be utilized. Visualization methods used for visualizing DQ are predominantly based on simple data representations. Approaches vary in how inspection of DQ is modeled for users, ranging from **visualizing raw data** and **annotating** the raw or aggregated **data visualization with DQ information** to **visualizing DQ information**. Visual encodings of DQ could also be differentiated by type of error: **Incomplete data**, **Outlying data**, and **Erroneous and anomalous data**.

2.2.1 Visualizing Data Quality

The visual encodings are used to amplify outlying or anomalous data, e.g., using color [XHWR06], or using interactive methods for highlighting DQ errors. Closely related to inspecting DQ is visual data mining. Keim [Kei02] proposed visual data exploration to be a central element for getting an overview of the data and analyzing patterns, allowing analysts to identify interesting subsets, which could be data of low or high quality. Keim classified visual data mining techniques into the types of data to be visualized, the visualization techniques, and the interaction and distortion techniques used. When visualizing DQ, we can distinguish between visualizing different types of data: (1) the raw data, (2) data annotated with DQ information, and (3) DQ information. Using these three types of data used, we can visualize particular types of DQ errors: (1) incomplete or missing data, (2) outlying data, or (3) anomalous data.

Visualizing raw data: Due to the potentially large size and high dimensionality of datasets during data pre-processing and data quality assessment, techniques employed for visualizing these data require a more sophisticated use of basic visualization principles to maintain the ability to explore large sets of data. The raw data is represented objectively to the analyst, hence he/she must use domain expertise and prior knowledge to identify anomalous data, the visualization will communicate this information implicitly, if the visual encoding is chosen appropriately. Keim [Kei02] describes *geometrically-transformed displays*, *iconic displays*, *dense pixel displays* (see Figure 2.15), and *stacked displays*, e.g., dimensional stacking (see Figure 2.16). For specific data formats, visualizations are used to amplify particular characteristics of the data and allow the identification of poor DQ. Time series data we can employ multiple methods to inspect individual dimensions (for MVTs), or observe patterns over time. For example, in TimeCleanser by Gschwandtner et al. [GAM⁺14] employed heatmaps (see Figure 2.17) or lineplot visualizations of raw and transformed data (see Figure 2.18). In tabular or multivariate data, parallel coordinates views can be used to determine outlying and anomalous entries

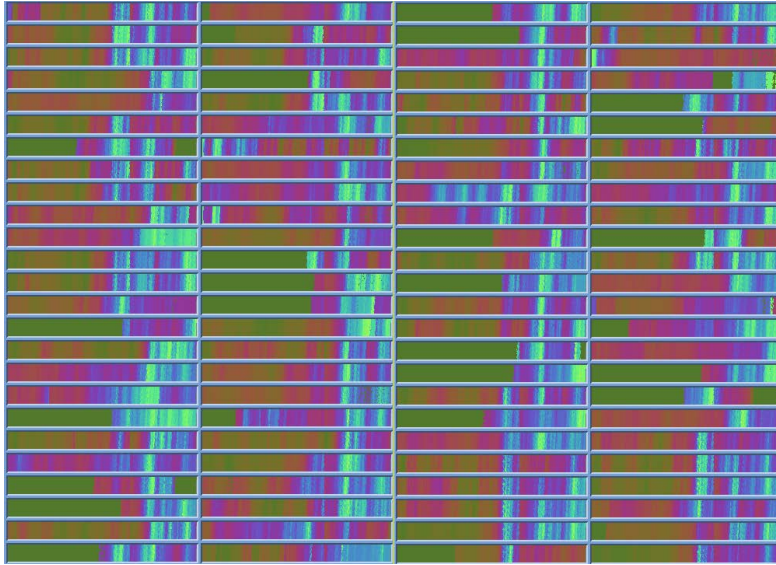


Figure 2.15: Dense Pixel display using a recursive pattern technique [Kei02].

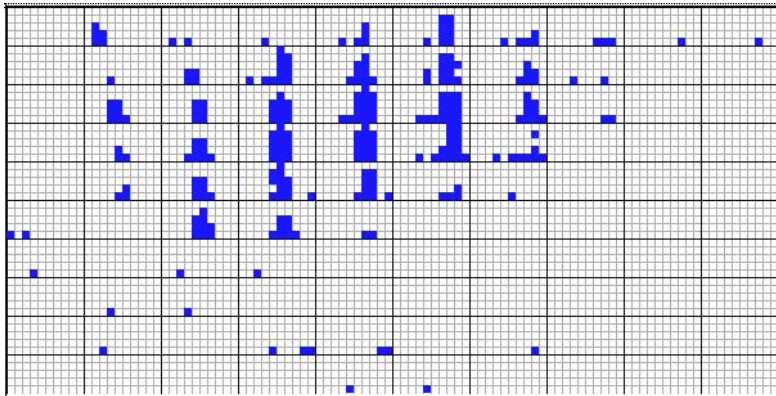
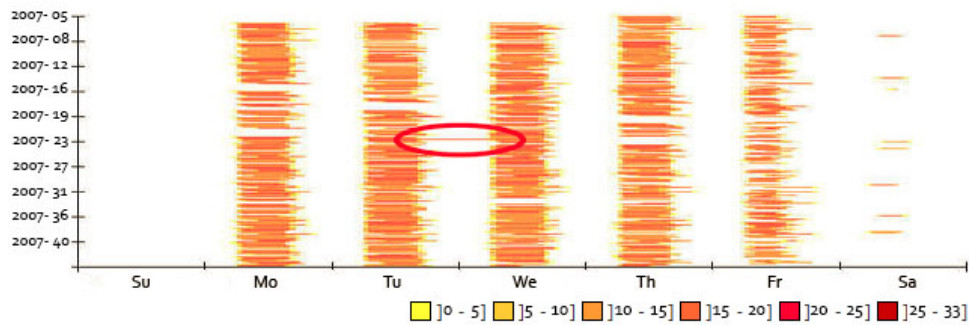
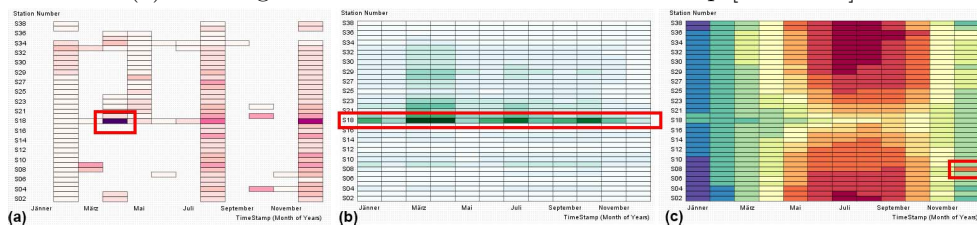


Figure 2.16: Dimensional stacking technique, using oil drilling data to map longitude and latitude onto the outer x- and y- axis, and ore grade and depth to the inner x- and y-axes [Kei02].

2. RELATED WORK



(a) Working hour intervals visualized in a heatmap [GAM⁺14].



(b) Different color mappings of a heatmap visualization of time series data to identify potential DQ errors (compare red rectangles) [GE18]. Figures (a), (b), (c) show different dimensions of a dataset across multiple measuring stations over the course of one year, the color-coding maps (a) the amount of quality problems, (b) the tuple count, and (c) the mean temperature.

Figure 2.17: Heatmap visualizations for exploring time series data. Figures (a) & (b) show different usage of heatmap encodings of periodic time series data.

(e.g., entries deviating from the remaining data). For node-link data, different visual representations and arrangement methods can be used to assess validity (see Figure 2.19). Other issues with scalability of large data can be addressed by using binning, e.g., for scatter plots [CLNL87]. Furthermore, high dimensional data can be reduced in complexity by applying dimensionality reduction, and visualizing the representative dataset. Multivariate data are often represented as tables, With large collections, it is not possible to adequately show them as such. However, visual abstractions of numerical, ordinal, or categorical data allow visualizing an overview of these tabular data, to allow swift exploration [RC94]. Sopan et al. [SFTM⁺13] iterate further on the concept of Table Lens [RC94] to show summary distributions of the column data.

Annotating data visualizations with DQ information: Predominantly, raw data visualizations presented above are adopted for showing data annotated with DQ information. Ward et al. [WXYR11] presented *embedded display* and *quality space display* (compare *Visualizing DQ information*) of quality. *Embedded display* should use visual encodings for DQ based on perception theory to convey quality effectively (compare Figure 2.20b). Similar to raw data visualization, parallel coordinates can be used to show multivariate data annotated with DQ measures (see Figure 2.20), and dimensionality



Figure 2.18: Small multiple lineplots showing the raw time series data, the difference of subsequent values, and the aggregated interval length of the recorded values [GAM⁺14].

reduction can be used to give a visual summary of the data, annotated by DQ measures (see Figure 2.21 [CCM09]). Sulo et al. [SEG05] show abstracted tabular data views with highlighting only incomplete, invalid, or duplicate data. Gschwandtner et al. [GE18] employ quality checks to highlight entries in raw tabular data views, using cell highlighting, and scrollbar annotation. Xie et al. [XWRH07] also employ a brushing technique highlighting entries of low or high quality, for example in multivariate parallel coordinate views. As with only displaying raw or aggregated data, annotating data with visual encodings of DQ information adds further complexity to the visualization. So it is important to consider how much information is encoded and communicated to the analyst. Correll et al. [CGOG11] proposed a *Confidence Fog* technique for indicating the confidence in a color value based on one or multiple measures of data quality or uncertainty. They encoded additional channels in lineplot charts to indicate quality/confidence of virus mutation values with blue and purple *runners* (lines) above and below the rows,

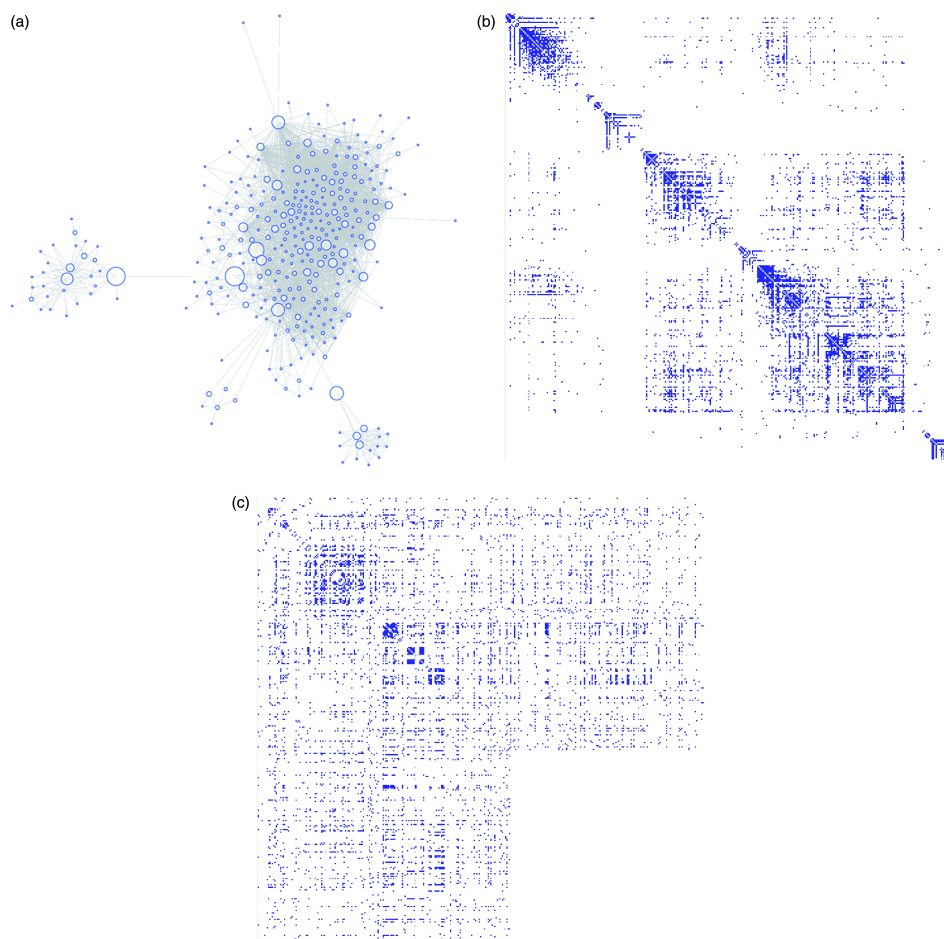
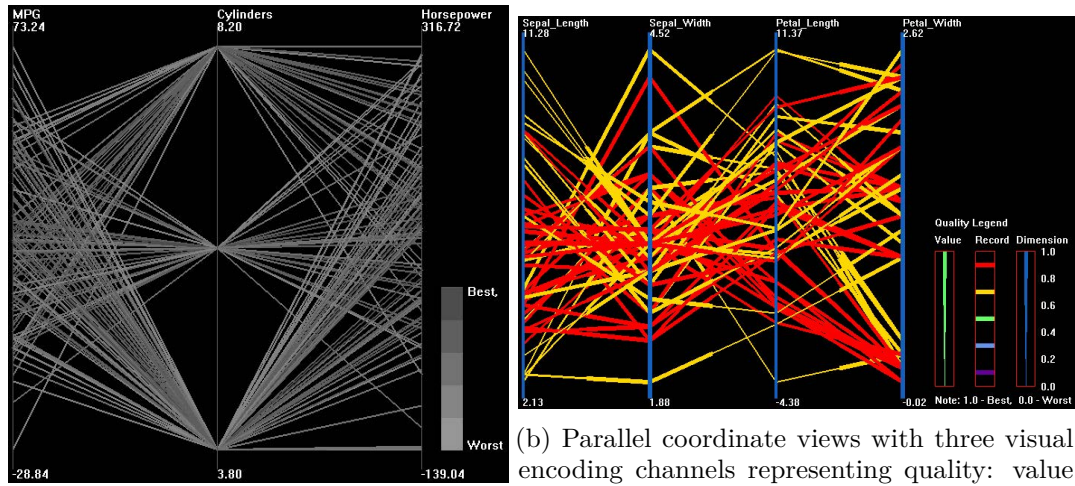
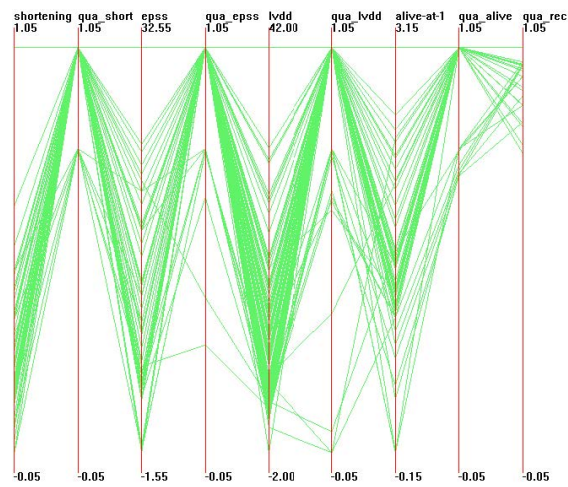


Figure 2.19: Different graph visualizations of the same social network, exemplifying that visual representations often affect how information can be perceived [KHP⁺11]. Figure (a) shows a node-link diagram of the network, Figure (b) shows the same network represented in an adjacency matrix with default sorting, while (c) shows the matrix sorted by ID, which gives insights that the latter portion of the data is missing.



(a) Parallel coordinate view enhanced with DQ metrics. (b) Parallel coordinate views with three visual encoding channels representing quality: value quality – line width, record quality – line hue, and dimension quality – column line width.



(c) Parallel coordinate views showing only DQ metrics for individual data records.

Figure 2.20: Parallel coordinate visualizations for assessing multivariate data [XHWR06].

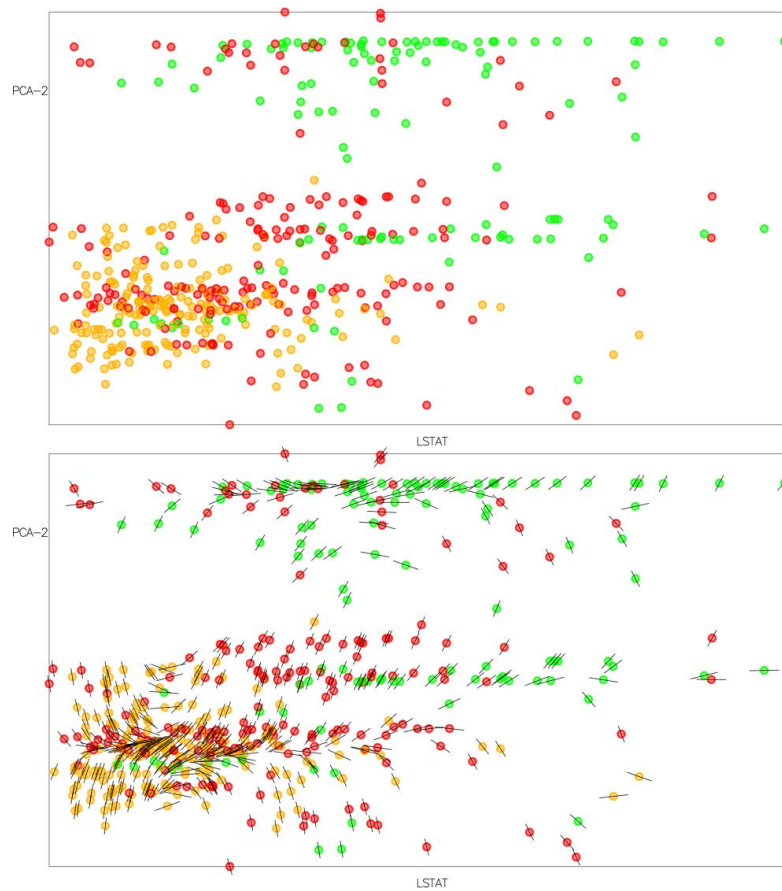


Figure 2.21: Scatterplot visualization showing a principle component representation of the BNHP dataset. The x-axis shows the variable LSTAT, the y-axis shows the principle component PCA-2. Colors denote different clusters, the lower view is annotated with a PCA sensitivity measure [CCM09].



Figure 2.22: ‘Confidence Fog’ on a sample subset of virus mutation dynamics data [CGOG11]. The top purple and bottom blue runners fade based on confidence measures.

as can be seen in Figure 2.22. An employed color palette would fade colors more or less rapidly as a measure of uncertainty or ambiguity. Ward et al. [WXYR11] proposed to quality measures to assist analysts with selecting, transforming, and mapping data to improve quality and ultimately generate high quality visualizations.

Visualizing DQ information: Another group of visually representing quality is only encoding DQ information. DQ metrics or other quality measures are used to show overall

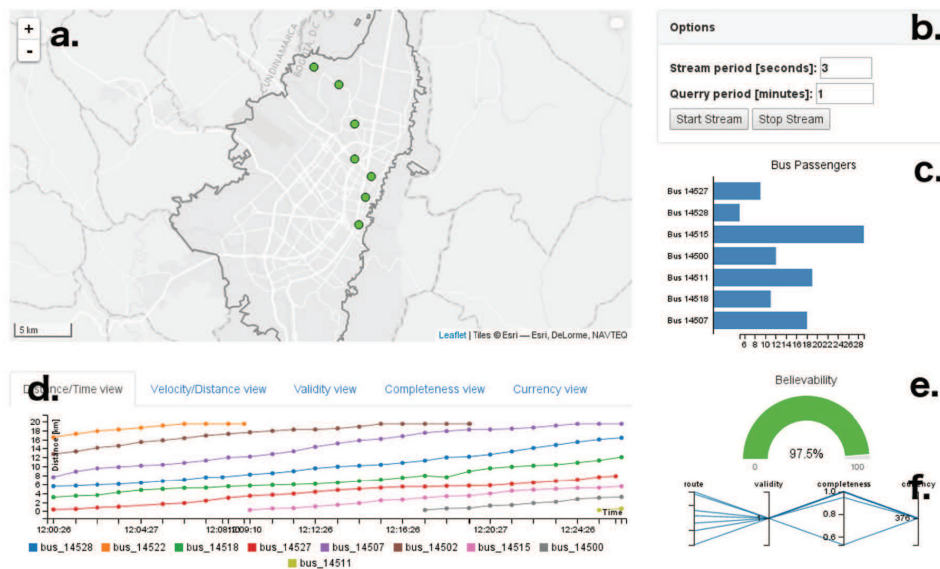
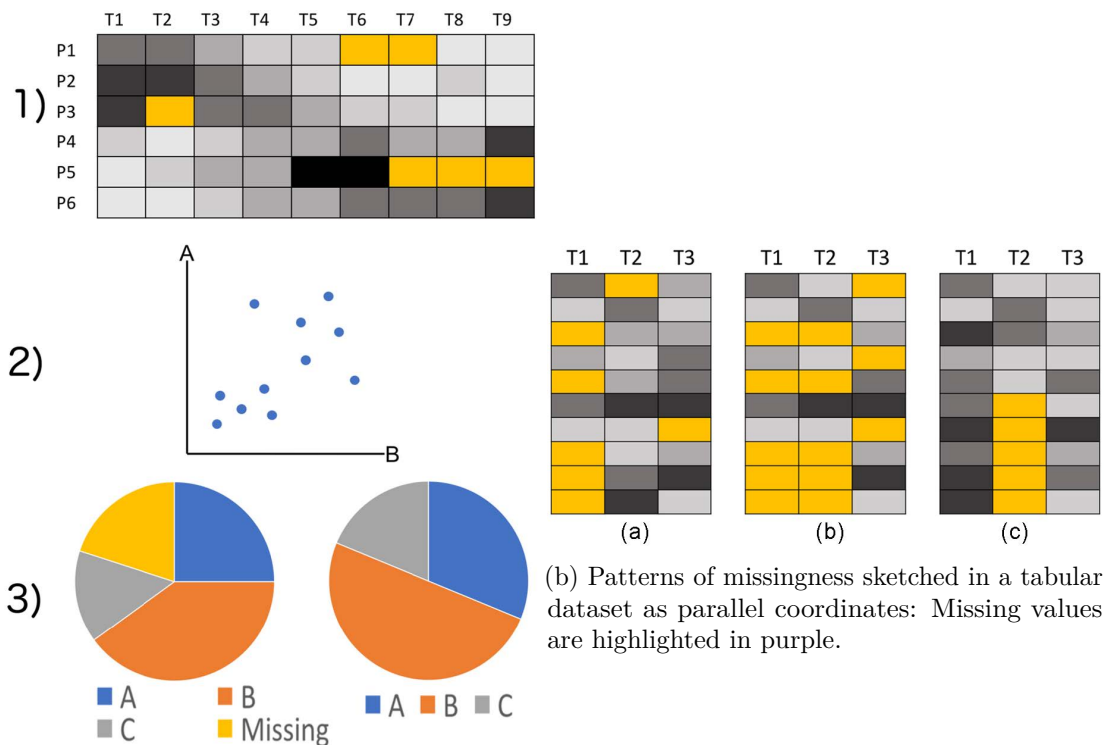


Figure 2.23: Composite view of spatio-temporal data showing bus trajectories. Elements (d) to (f) show different DQ metric views, giving overview of different quality characteristics in the data [TZHH18].

quality as single or multiple DQ metric values [TZHH18], inspect DQ error distributions [XWRH07], or locate DQ errors within a dataset, mapping DQ metrics to data value dimensions [TZHH18]. Xie et al. [XWRH07] encoded data value quality as a *Stripe Quality Map* and a *Histogram Quality Map*, showing quality distribution histograms, and aggregating quality measures across dimensions. Triana et al. [TZHH18] employ dedicated DQ metric views to investigate the quality of a dataset on different levels of granularity. The metrics are shown in line charts, mapping the values against the temporal domain of the original time series to find local phenomena (see Figure 2.23(d)). Aggregated values of DQ metrics give analysts insight into the overall quality (compare Figure 2.23(e,f)).

I have discussed three ways of visualizing data in order to assess quality. However, for visually exploring and inspecting different types of DQ errors, it is necessary to employ appropriate visual encoding techniques. Correll et al. [CLKS18] conducted a study about commonly used summary visualization techniques for data distributions, comparing density plots, histograms and dot plot visualizations. They concluded that density plots are robust to find missing data, outliers, and anomalous data. However, participant performance varied based on the bandwidth of these plots, and that participants would generally perform better with adequate histogram bin sizes and mark opacities. However, it is also necessary to investigate more means for visualizing DQ errors and will go into detail about various data domains.

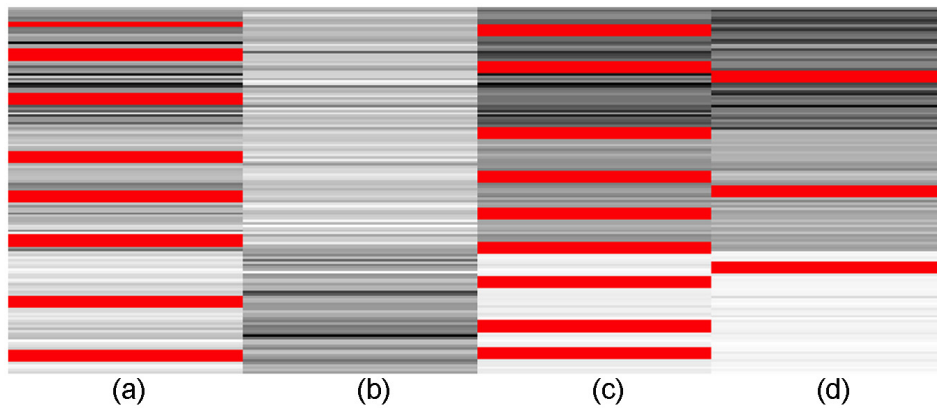
(1) Visualizing Incomplete and Missing Data: There are two approaches to account for missingness in data: removing entries with missing values or imputing missing



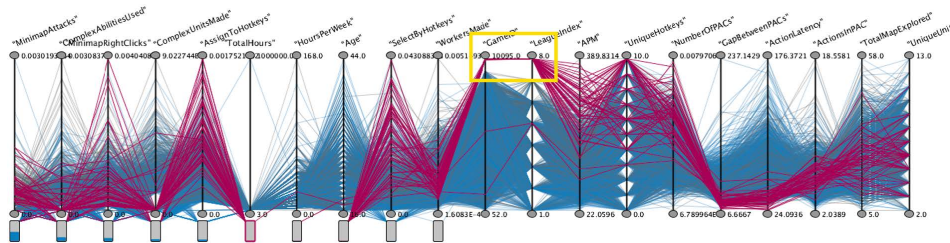
(a) Missingness impacting visualization: (1) missing values are perceivable, (2) missing values are invisible, (3) missing values introduce a bias.

Figure 2.24: Different views for analyzing missingness in data by Fernandes et al. [FWM⁺18].

values with estimations. Fernstad [Fer18] emphasized the importance of methods for visualizing missing data, where she first identified sources of missing data, how they can be identified, and how analysts can deal with them. Eaton et al. [EPD05] defined at which levels missingness can impact visualizations (compare Figure 2.24a): (1) being perceivable in the visualization, (2) missing values being invisible in the visualization, (3) biasing the visualization. Fernstad furthermore classified patterns of missingness into *amount missing*, *joint missingness*, *conditional missingness* (see Figure 2.24b). The user study she performed indicated that a *Matrix Plot*, a heatmap that highlights missing data with color, is the most appropriate for *amount missingness* and *joint missingness* value identification tasks, and parallel coordinate views are most appropriate for tasks related to *conditional missingness* (see Figure 2.25). Song and Szafrir [SS18] evaluated methods for visualizing missing values in line graphs and bar charts, they chose different representations of missingness: removing points, highlighting missing points, visually downplaying missing points, and annotating missing points. The results showed that “the ways systems impute and visualize missing data can also manipulate perceived



(a) Matrix plot displaying a dataset with four variables: (a), (b), (c), (d). Numeric values are represented by a grey scale, with missing values being represented in red.



(b) The user study indicated parallel coordinates to be well suited for identifying conditional missingness.

Figure 2.25: Visualizations suited for visualizing missing data, as evaluated in a user study by Fernstad [FWM⁺18]. The user study indicated (non-significant result) that the Matrix Plot (Figure (a)) is most appropriate for amount missingness and joint missingness.

data quality and confidence in results” [SS18, p. 9]. This also means that perception of quality depends on the data, problem, and domain and needs to be taken into consideration. In time series analysis, missing data is often imputed through statistical methods. Such methods inevitably introduce uncertainty into the data: Single value imputation methods neglect uncertainty altogether, while repeated sampling methods concretely compute imputation uncertainty. These uncertainties can be visualized using uncertainty visualization techniques shown in Section 2.4.

(2) Visualizing Outlying Data: Famously, the Anscombe Quartett [Ans73] shows the same statistical profile, while exhibiting vastly different distributions in a scatter plot representation, partly also due to outliers (see Figure 2.26). Visual data inspection allows users to perform summary statistics estimation and ultimately outlier identification. The most basic technique for visualizing outliers is the boxplot [PHKD06] and summary plots [PKRJ10] for one or two-dimensional data. Highlighting is the most prevalent method to emphasize potential outliers, for example using color through appropriate color maps [CAFG12, SNHS17], or by explicitly flagging outlying values (e.g., prediction-based

I		II		III		IV	
x_1	y_1	x_2	y_2	x_3	y_3	x_4	y_4
10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58
8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76
13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71
9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84
11.0	8.33	11.0	9.26	11.0	7.81	8.0	8.47
14.0	9.96	14.0	8.10	14.0	8.84	8.0	7.04
6.0	7.24	6.0	6.13	6.0	6.08	8.0	5.25
4.0	4.26	4.0	3.10	4.0	5.39	19.0	12.50
12.0	10.84	12.0	9.13	12.0	8.15	8.0	5.56
7.0	4.82	7.0	7.26	7.0	6.42	8.0	7.91
5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.89

(a) Four distributions with the same statistical profile.

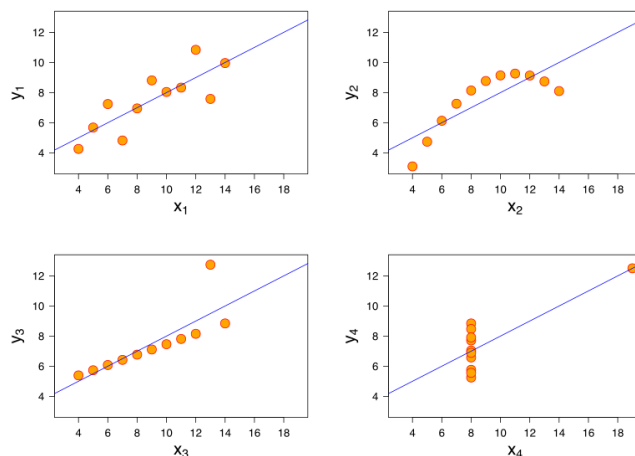


Figure 2.26: Anscombe’s quartet exhibits four different point distributions with the same statistical properties, but different representations when visualized [Ans73].

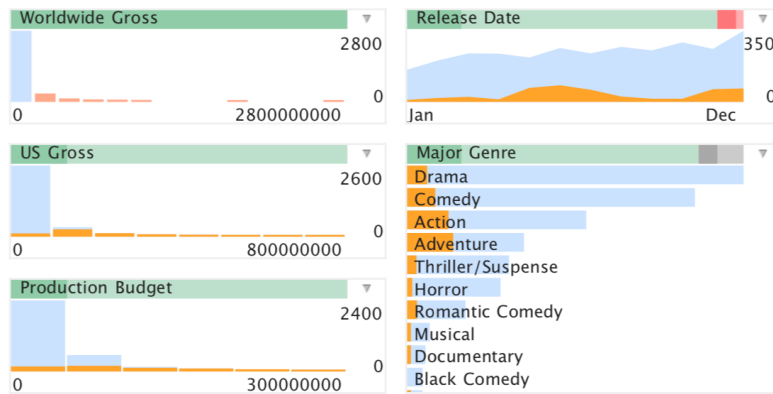


Figure 2.27: Automatically generated interactive summary visualizations show the histogram of numeric values, ordinal data are aggregated. Integrated in these visualizations, the orange bars show the distribution of quality problems in the respective columns, like missing values or outliers [KPP⁺12].

and clustering-based anomaly score [JSMK14]). Such methods utilize the underlying statistical distribution to calculate the outlier score. For example, Kandel et al. [KPP⁺12] use a summary visualization showing a histogram or barchart visualizations raw numeric data (compare Figure 2.27). This allows identification of outlying values in the distribution. In time series analysis and bivariate scatter plot visualizations, Correll and Heer [CH17] examined the use of regression information and how it aids users with trend estimation. They concluded that if outliers are important to analysis, designers should visualize the outlier sensitive models accordingly. A plethora of outlier and anomaly detection techniques can be employed that all come with particular strengths and weaknesses [CBK09], specifically high-dimensional methods, e.g., clustering- and classification-based techniques. Schulz et al. [SNHS17] introduce data descriptors among which also record descriptors can be used to detect outliers or duplicates.

2.3 Interactive Methods for Data Quality Assessment

Kandel et al. [KHP⁺11] described data wrangling as an iterative exploratory process. I propose that this holds true for DQ assessment. Interactive methods for exploring and improving DQ significantly improve and support analysts when conducting DQ assessment. Erroneous values can be hidden in plain sight if available visualizations are not appropriate for analysts to detect particular error types. For example, Figure 2.19 shows different representations of a graph. It demonstrates that the missingness features in visualizations can be different depending on the chosen visualization [KHP⁺11]: Figure 2.19(a) apparently shows a regular social network graph in the node-link representation. Figure 2.19(b) shows a matrix diagram showing connections between nodes, with automatic permutations to highlight clusters. Figure 2.19(c) shows the matrix diagram with its raw sorting, which makes it apparent that a significant part of the connections

seems to be missing. This raises the question how users can best be supported in their DQ assessment tasks. Interactively changing representation can help exploring view points. Interaction is an integral part of exploratory data analysis and exploratory visual analysis and helps achieving the goal of finding new insights [BH19], in the case of DQ assessment finding DQ errors or confirming that quality is sufficient for downstream analysis. Interaction techniques can facilitate exploration and inspection of local phenomena, like Focus+Context (F+C) and distortion techniques to allow exploration of details while maintaining the overview of the entire data. This is possible for multiple types of data, for example interactive lenses [TGK⁺17] for time series [Kin10, ZCPB11], geospatial data, flow data, volume data, multivariate data [RC94], node-link data, or text and document data. Rahm and Do [RD00] gave an overview of existing DQ assessment tools and differentiated between *data analysis and re-engineering*, *specialized cleansing*, and *extraction, transformation, loading tools*. However, they already motivated early on that such tools must support multiple aspects due to limitations in interoperability, which has been mainly addressed by commercial tools, to extend the palette of available functionality.

VA systems have been developed early on to help analysts with data cleansing, profiling, and wrangling, and continue to be developed, extended, and turned into commercial tools and systems. For profiling tabular datasets, Rao and Card introduced TableLens [RC94], a F+C method for getting an overview of a tabular dataset. It could be used to explore relationships and interesting patterns in the data. Sopan et al. [SFTM⁺13] extensively added interactivity and responsiveness to Rao and Card's TableLens by introducing sorting, brushing, interactive tooltips, and configurable heatmap overviews (see Figure 2.28). They use histogram, heatmap, or boxplot encodings and show two different representations for either single cell values or a compact row-based overview, which allows for inspecting individual rows, but also exploring the entirety of the dataset and discover interesting patterns. To support data profiling, brushing and linking methods were introduced to explore value distributions across dimensions [XHW06, XWRH07]. Xie et al. [XWRH07] furthermore introduced interactive methods for highlighting data based on quality measures in parallel coordinate views, e.g., quality measure based brushing to allow inspection of values with low/high quality. To assess the level of *preservation* of the original data, Ward et al. [WXYR11] introduced abstraction quality measures – *Statistical Measure*, *Histogram Difference Measure*, and *Nearest Neighbor Measure* – that can be used to explore different visual representations of the data. Cao et al. [CWR14] introduced an *Outlier Workbench* called *LEAP* to detect outliers in sliding window streams and along with it an interactive visual exploration system called *VSOutlier*. Using an interactive outlier type selection, analysts can inspect different outlier detection parameters in a juxtaposed comparison view to determine the prevailing type of outlier.

In an effort to facilitate data wrangling, *Potter's Wheel* was developed to support interactive transformations with **ease of specification**, **ease of interactive application**, and **undos and data lineage** (which will be further discussed in Section 2.5.1). The transfor-

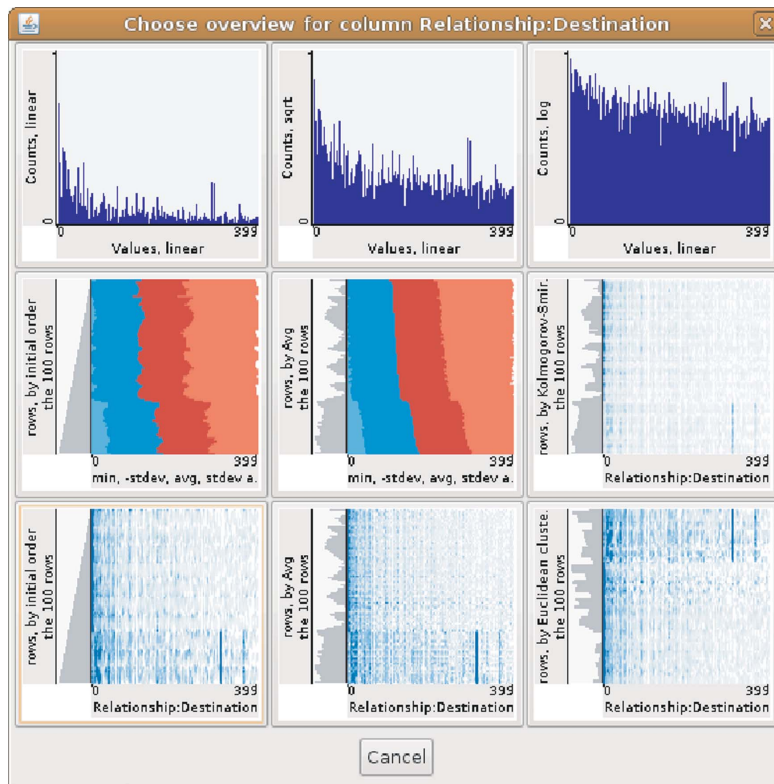


Figure 2.28: Table overview interactions: the user can decide between different overview representations for individual columns of the dataset [SFTM⁺13].

mations offered in the tool could be applied interactively, and overview histograms could be used to see the distribution of transformed values. The concept of easy transformation specification and execution was adapted in *Wrangler* [KPHH11, GKHH11] to provide a mixed-initiative user interface for interactive data wrangling. The prototype suggests applicable transformations and provides visual previews to show analysts the potential outcome of the transformation. This was combined with a transformation history to undo or redo actions. Basic interactive visualizations, like barcharts on top of columns showing potential erroneous data, are used to suggest transformations based on these data subsets. Transformations affect the data and potentially change the distribution, rendering the transformed data non-representative. *OpenRefine*, formerly known as *Google Refine*, is an open source tool, which has been under continuous development, to facilitate cleansing and wrangling data. It provides an underlying coding environment for data wrangling and cleansing, facilitated by a textual preview of the data. Small overview visualizations and summary tables help users to identify outlying and anomalous data and apply filters and transform the filtered data.

Kandel et al.’s [KPP⁺12] *Profiler* integrates the features from previously presented tools (compare [KPHH11, XWRH07]), supporting different data types, featuring anomaly

2. RELATED WORK

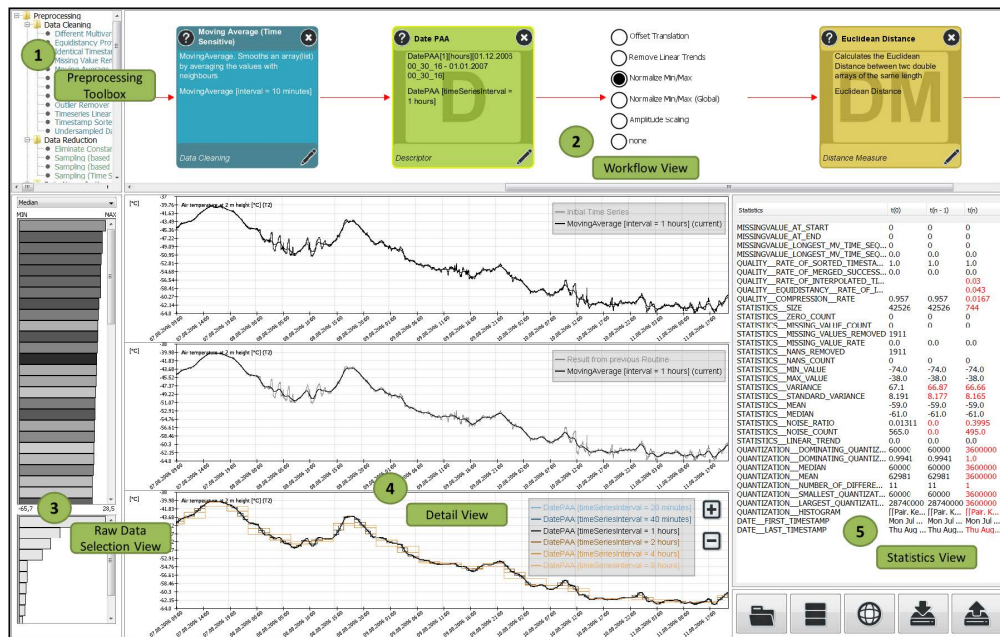


Figure 2.29: A visual-interactive pre-processing system for time series showing a toolbox of algorithms, the workflow, a raw data and visualization view, as well as statistics measures [BRG⁺12]: The system consists of (1) the Pre-processing Toolbox, (2) a Workflow View, (3) Raw Data Selection View, (4) Detail View, and (5) the Statistics View.

detection for common DQ errors and provides summary visualizations to facilitate DQ assessment. This is done by using data type inferences and a data mining-based anomaly detection engine to feed information into view recommendations and a view manager providing interactive linked views.

In time series analysis, data wrangling and cleansing is mandatory to enable downstream analysis, often due to the high specificity of processing algorithms. Bernard et al. [BRG⁺12] facilitated the *pre-processing workflow* of time series data by providing analysts with a toolbox of algorithms to build custom workflows (see Figure 2.29). They provided interactive views to preview inputs and outputs of the processing steps along with statistical measures to estimate the impact of the transformation steps on the data, and adjust parameters of operations accordingly. TimeCleanser by Gschwandtner et al. [GAM⁺14] employs dedicated time series visualizations to help analysts identify missing or outlying data. The dedicated visualizations emphasize on error discovery and are supported by the use of automatic *quality checks* supporting analysts in finding and correcting DQ errors, particularly in the time-oriented domain. Arbesser et al. [ASMP17] approach DQ assessment by employing hierarchically structured *plausibility checks* to guide data exploration with extensive drill-down features to let users identify erroneous data (see Figure 2.30a).



Figure 2.30: Visplause combines linked views: The central part is the DQ overview where analysts can explore the data by plausibility classes. Various time-oriented visualizations allow directed exploration of the data [ASMP17].

DQ assessment has continued to be an evolving field of research. Particularly with the continued increase of data size and dimensionality, data cleansing, wrangling, and profiling are vital tools for ensuring that the data is of adequate quality and still representative of the original dataset, with the premise to allow detailed inspection of the (pre-)processing workflow on demand [LMW⁺17]. Liu et al. [LAW⁺18] stated the importance of VA for ensuring DQ in multiple loops of the analysis pipeline. Figure 2.31 shows a framework of steering DQ with VA where user and system feedback are used to apply screening, diagnosis, and correction of data with the help of visual-interactive methods and visual summaries, two methods extensively used in VA.

Visual Analytics Methods for Uncertainty

Previously, I have discussed the definition, potential sources, and the quantification of uncertainty. Uncertainty is a crucial part of visualization and VA: it can be inherent to the data, generated along the processing workflow, or when generating visualizations or insights [SSK⁺16]. Uncertainty is important in decision making, where a biases in information can be assigned certain probabilities to determine the “better” or “more optimal” solution. Early works in uncertainty visualization were conducted in spatial data visualization and geo-information science, investigating visual representations of uncertainty [BW88, Mac92, MRH⁺05]. MacEachren et al. [MRH⁺05] recognized the importance of uncertainty in the process of analytical reasoning and how users cope with the existence of uncertainty, which he further specified for the particular field of VA in recent work, to address: “(1) understanding the basis for uncertainty; (2) understanding levels of uncertainty; and (3) understanding the role of information and knowledge in

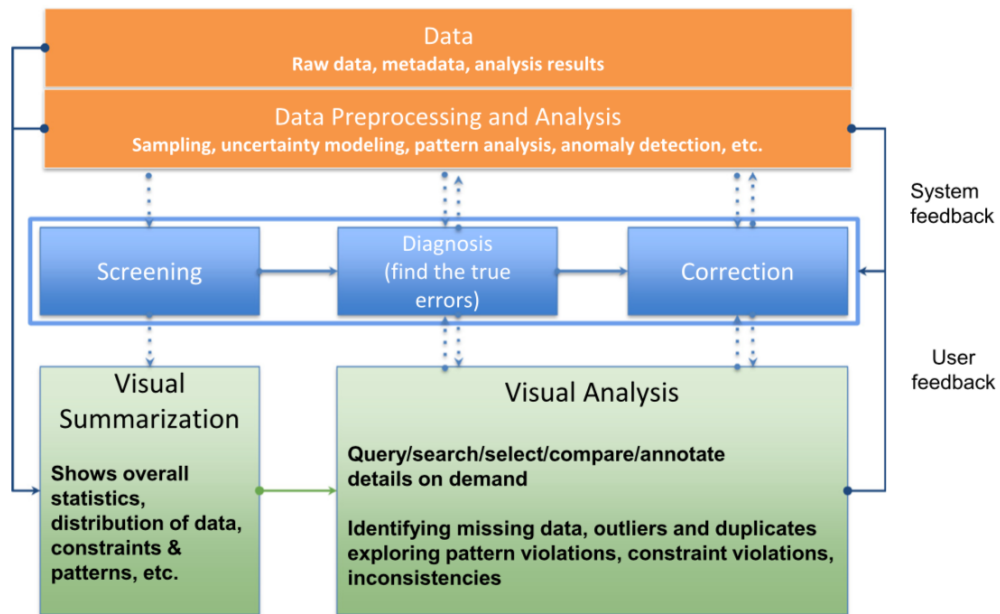


Figure 2.31: Framework for steering DQ with VA [LAW⁺18].

relation to uncertainty” [Mac15, p. 2]. We must be aware of both how uncertainty is communicated to the analyst, but also how it can be perceived and how it affects the mental model. Griethe and Schumann [GS06a] described uncertainty to be integrable into visualizations by (1) utilizing available graphical variables, using attributes as shown in Figure 2.34a, (2) integrating additional geometrical objects, (3) using animation, (4) using interactive representations, or (5) addressing other human senses. In the following sections, I will discuss (1) visualization of uncertainty in information visualization, (2) interactive methods for exploring uncertainty, and (3) the role and use of uncertainty in VA.

2.4 Visualizing Uncertainty

As stated before, uncertainty visualizations were initially researched in geo-information sciences [Mac92, MRH⁺05]. Hence, the used visual encodings of uncertainty were oriented towards to 2d or 3d maps (see Figure 2.33). Thomson et al. [THM⁺05] presented a typology for visualizing analytic uncertainty. Early on they stressed the importance of aggregating and propagating uncertainty. MacEachren et al. [MRO⁺12] constructed a typology of *information uncertainty* and constructed abstract and iconic visual variables to point sets (see Figure 2.34a) to conduct an empirical study evaluating the abstract and iconic intuitiveness and the subjectively assessed accuracy of visual encodings of uncertainty. Figure 2.34b shows the results for abstract visual variable intuitiveness. Among other results, the symbol sets *fuzziness*, *location*, and *value* received the highest

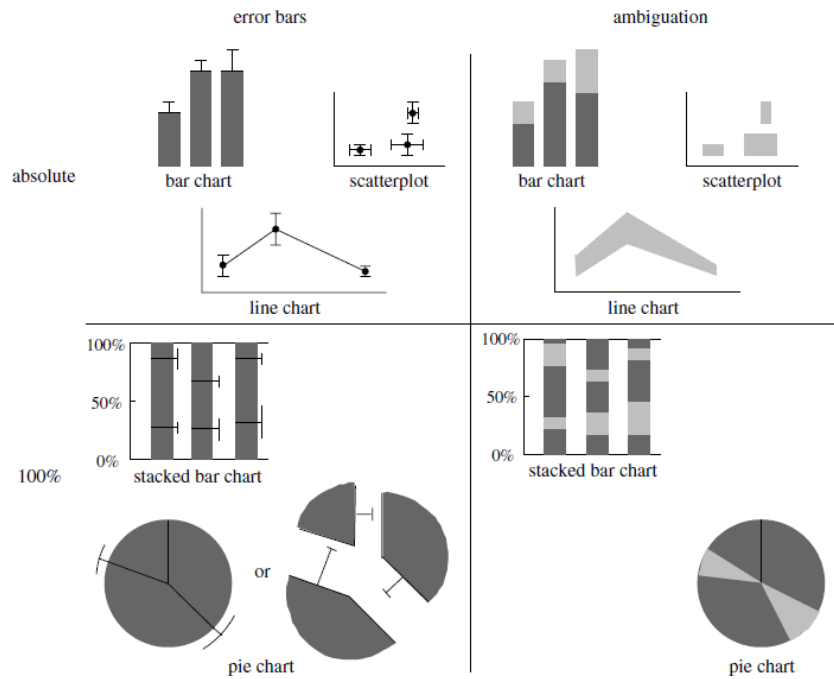


Figure 2.32: Early methods for visualizing uncertainty using error bars or disambiguation in various basic visualization techniques by Olston and Mackinlay [OM02].

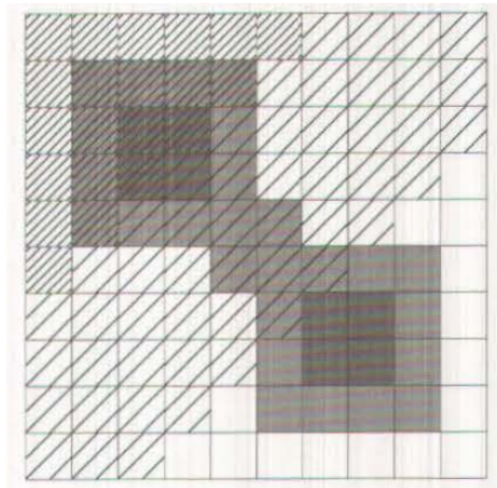


Figure 2.33: Bivariate map of visual encodings for risk (block shading) and uncertainty (block textures) [Mac92].

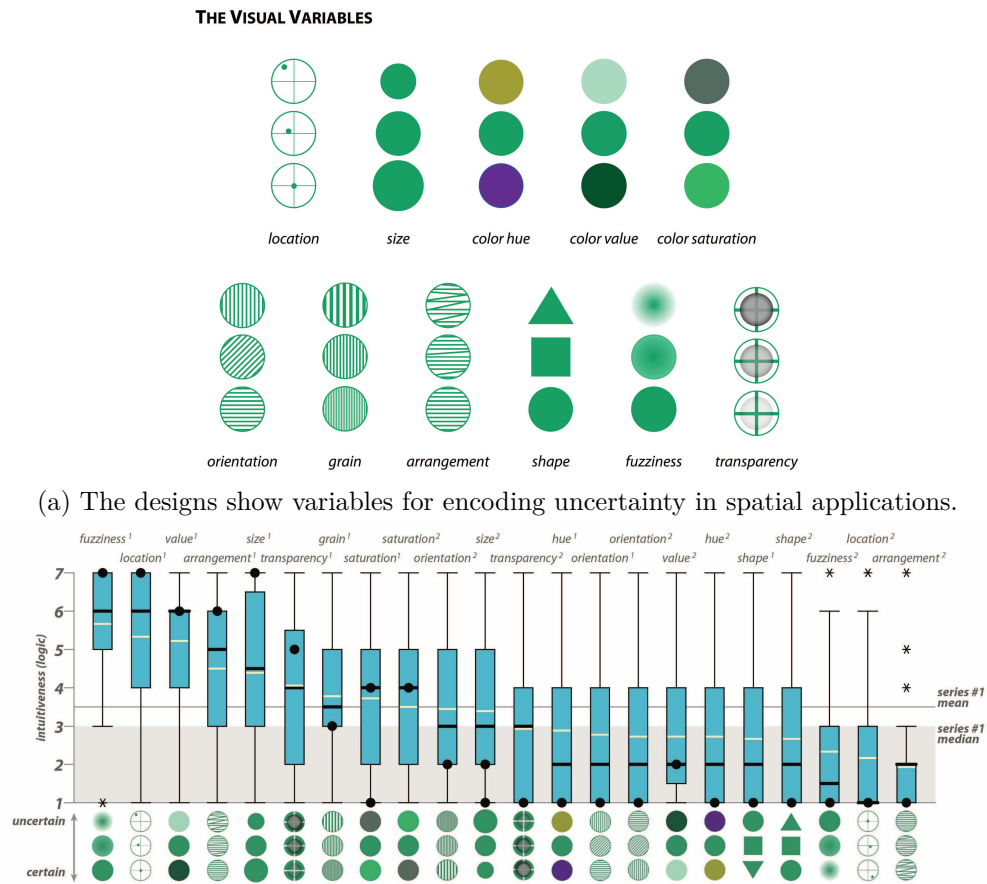


Figure 2.34: (a) Abstract visual variables for visualizing uncertainty in point symbol sets by MacEachren et al. [MRO⁺12] and (b) the corresponding results for designs evaluated towards their intuitiveness.

perceived intuitiveness for discrete ordinal uncertainty. They determined winning visual encodings for data domains (space, time, or attribute) and types of uncertainty as most appropriate recommendations for encoding uncertainty. Olston and Mackinlay [OM02] started exploring the design space on visualizing abstract data with bounded uncertainty, using error bars or disambiguation (see Figure 2.32). Griethe and Schumann [GS06a] described integration of uncertainty in visualization to be possible by using free/available graphical variables of the visual encodings, among others. Potter et al. [PKRJ10] used descriptive statistics measures to merge statistical summary plots and combined histograms and create a detailed uncertainty information glyph in datasets (compare Figure 2.35). Brodlie et al. [BAL12] gave a review of uncertainty visualization techniques in data visualization, using a typology by data dimensions ranging from 1D to multi-dimensional

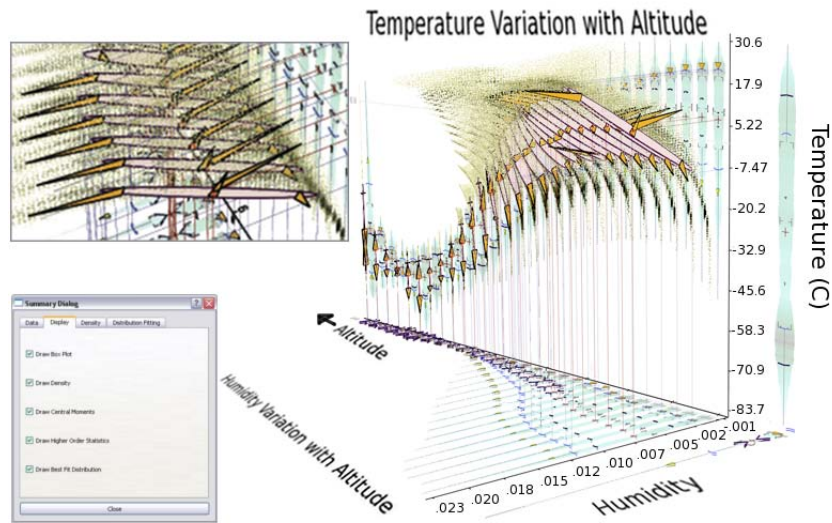


Figure 2.35: Visualizing uncertainty using joint histogram displays combined with covariance (purple rectangles) and skew variance (orange triangles) measures (top left) [PKRJ10].

and vector data uncertainty. Simultaneously, Bonneau et al. [BHJ⁺14] gave a more formal view on the quantification of uncertainty, but also discussed the state of the art in uncertainty visualization. They differentiated between traditional independent representations of uncertainty and uncertainty functions and the integration of uncertainty in visualization. They classified visualization techniques using uncertainty into **(1) comparison techniques**: showing side-by-side views for easier comparison, overlaying information, or producing difference representations, **(2) attribute modification**: mapping uncertainty to free visual variables, **(3) glyphs**: signifying data through certain parameters, and **(4) image discontinuity**: specifically utilizing humans' ability to detect discontinuities to communicate certain data characteristics. In another effort to classify the state of the art in uncertainty visualization, Potter et al. [PRJ12] used both data dimensions and uncertainty dimensions to distinguish different visualization techniques. In recent years, visual encodings of uncertainty have been subject of extensive evaluation in multiple domains, e.g., time-oriented data [GBFM16, WBFL17], abstract data [OJS⁺11], spatial data [MRO⁺12, OJS⁺11], medicine [AMTB05], model simulation, and weather forecasting [LMK⁺15]. Other works construct uncertainty visualization design based on uncertainty sources and generation. For example, Potter et al. [PWB⁺09] and Liu et al. [LMK⁺15] created uncertainty visualizations from ensembles, to communicate the average outcome of ensembles more appropriately to users. They use samples of spatio-temporal paths to create a scalar field and construct an elliptical approximation of storm path predictions, assigning uncertainty to the geospatial location (see Figure 2.36).

Efforts have been made to evaluate the effectiveness of color encodings [CMH18], anima-

2. RELATED WORK

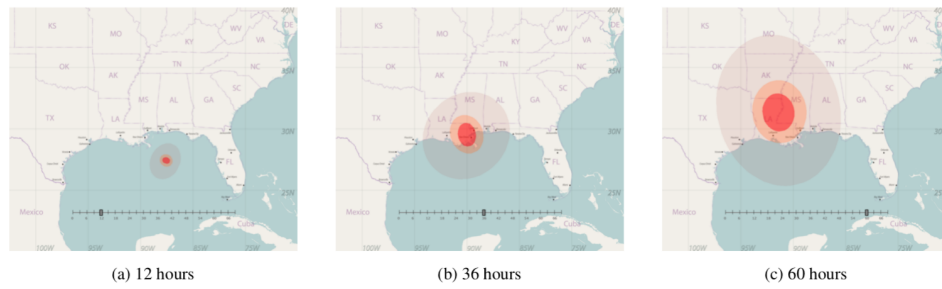


Figure 2.36: Minimum enclosing ellipses representing different levels of confidence for a hurricane, the orthogonal axes correspond to hurricane bearing and speed [LMK⁺15].

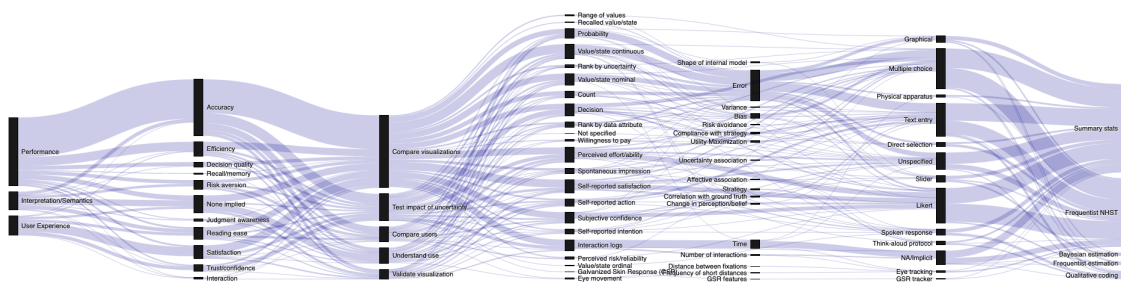


Figure 2.37: Evaluation paths of studies evaluating uncertainty visualization along six characterization properties [HQC⁺19].

tion [HRA15], and other alternative uncertainty displays [FWM⁺18]. These studies can be used to draw a more general picture about humans' perception and decision-making ability based on uncertainty visualizations. Gschwandtner et al. [GBFM16] compared uncertainty visualizations for uncertain start- and end-times of intervals, which resulted in favor for error bars and ambiguity plots in case of determining start and end points, as well as duration estimations. For mapping probability distributions, gradient plots were recommended over accumulated or violin plots, even though personal preferences differed. Hullman et al. [HQC⁺19] conducted a comprehensive survey of uncertainty visualization evaluation, coming up with an overview of evaluation decision levels in form of a Sankey diagram with evaluations being characterized along a path: (L1) Behavioral target, (L2) desired effect, (L3) evaluation goal, (L4) measure, (L5) elicitation, and (L6) analysis. Figure 2.37 shows the 372 paths from 86 publications on evaluating uncertainty visualization. They found that most evaluations focused on performance and user experience comparing uncertainty visualization designs with *confirmatory* evidence, as well as measuring accuracy and decision, suggesting to include confidence reports in studies on uncertainty. The final remark I found notable was their future evaluation suggestion to uncover strategies for completing particular tasks. This will be an important aspect that I will discuss in detail when discussing approaches in the upcoming section.

2.5 The Role of Uncertainty in Visualization and Visual Analytics

Thomas and Cook postulated visualizations techniques for VA to “*support the understanding of uncertain, incomplete, and often misleading information*” [TC06, p. 99]. Due to the uncertainty inherent in data, the processes and methods for visualizing and analyzing that data it is necessary to be aware of the extent of such uncertainties [CCM09]. Griethe and Schumann [GS06a] described the basic process of uncertainty visualization. They distinguish between uncertainty acquisition and visualization. Figure 2.38 shows these separated loops and their interrelations, due to uncertainty being generated throughout the visualization process as well. Correa et al. [CCM09] presented a framework for propagating uncertainty from data sources to the analyst (see Figure 2.39). In particular, the process of uncertainty modeling and propagation is separated from data and visual transformations. Furthermore, uncertainty is also generated from derived data, feeding into visual mappings and views. MacEachren [Mac15] advocated for using VA research to take the challenge of understanding the effect of uncertainty for decision-making, or risk assessment, and ultimately investigate the uncertainty that is inherent to reasoning and decision-making. Sacha et al. [SSK⁺16] extended their framework with the role of uncertainty during knowledge generation and trust building. Their model exhaustively describes how components of the VA process, on both the system and the human side, influence uncertainty and components influencing uncertainty. It shows the encompassing influence of uncertainty in VA, and serves as a basis for the upcoming overview of interactive methods for supporting uncertainty analysis. Since an exhaustive review of literature about different uses/roles of uncertainty in VA is out of scope for this work, I will focus on the topics related to DQ assessment and provenance capture and analysis (compare [SSK⁺16]). Out of the listed topics, the following topics were selected as important for the scope of this thesis: (*s2*) uncertainty sources and types, (*s3*) transforming data (*s5*, *s6*) model building, parametrization, and selection, (*s7*) uncertainty in visualizing data caused by resolution or overplotting, (*s9*) perception and uncertainty awareness, (*h8*) using uncertainty in systems, (*g4*) exploring uncertainty, (*h9*, *h6*) awareness and trust of uncertainty in models and data, and (*h4*, *h7*) internalizing knowledge and knowledge generation.

Along with their conceptual framework for analyzing uncertainty in VA, Correa et al. [CCM09] also presented the means for modeling, propagating, aggregating, transforming, and visualizing uncertainty of multivariate data. Initially, Zuk and Carpendale [ZC07] proposed storing and communicating propagated/derived uncertainty. This is done to enable both investigating the most reliable data by mapping uncertainty to transparency, or investigating regions/values with associated high uncertainty to discover sources of uncertainty. Uncertainty of variables is modeled using Gaussian Mixture Models, uncertainty propagated by the PCA transformation is quantified using linear regression. The analyst can explore the various measures of uncertainty, like variable sensitivity, PCA projection uncertainty, and variance in clustering (see Figure 2.21). As such, it is possible to evaluate how appropriately the data transformations have been applied and

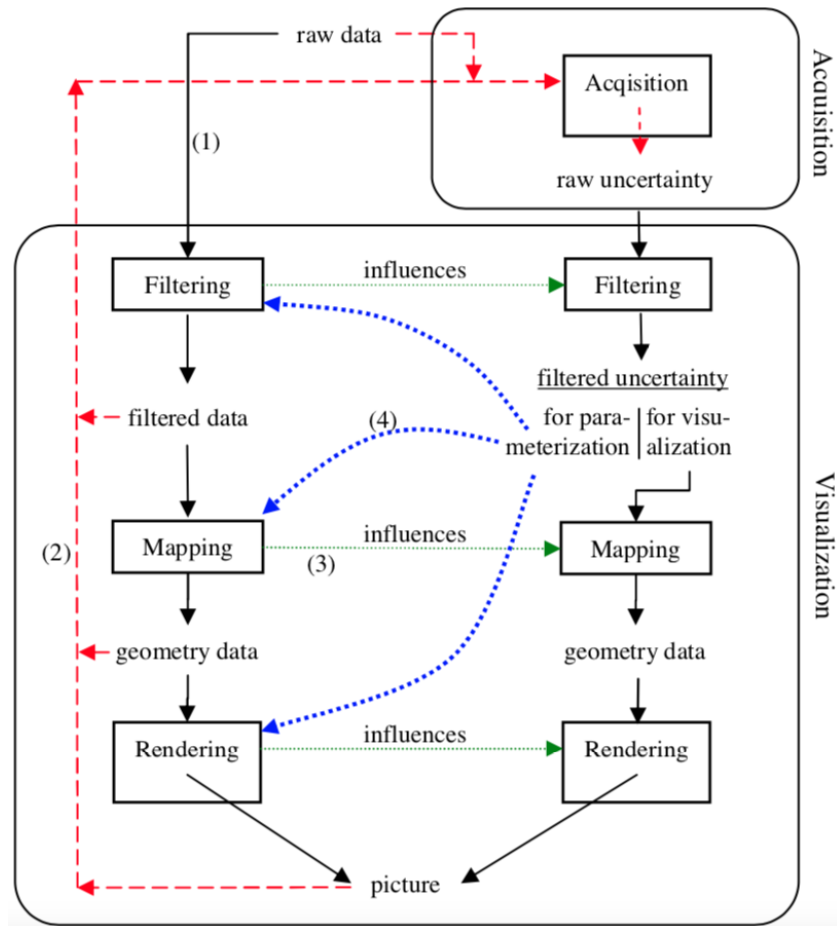


Figure 2.38: The process of visualizing uncertainty by Griethe and Schumann [GS06a]. Visualizing uncertainty is preceded by uncertainty flowing through the data model: (1) data and uncertainty flowing from the raw data through the entire transformation process, (2) in- and output of uncertainty data acquisition, (3) dependencies between the uncertainty transformation process and the raw data transformation, and (4) parametrization of the visualization pipeline.

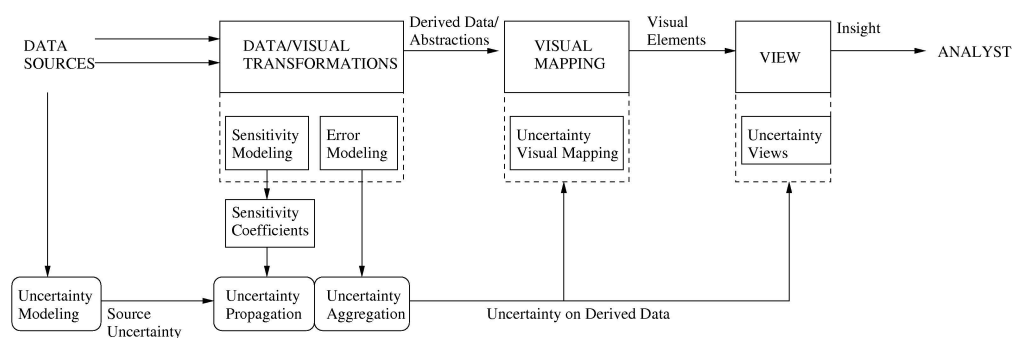


Figure 2.39: The uncertainty-aware VA process by Correa et al. [CCM09].

the uncertainty introduced during modeling itself. This shows the first approach towards raising awareness of uncertainty being introduced during transforming data for visual analysis.

Wu et al. used an uncertainty quantification measure to quantify uncertainty introduced at every step alongside an analytic processing workflow [WYM12]. They showed the importance of estimating the amount of uncertainty generated during analysis itself and explicitly communicate this uncertainty to the analyst, who is empowered to select analysis workflows that are least affected by uncertainty. The basis of their uncertainty propagation and visualization workflow is an automated system collecting “history information and estimating error ellipsoids of every data item, which are subsequently combined to obtain an overall uncertainty level for drawing the uncertainty flow visualization” [WYM12, p. 2528] They characterize the variations of uncertainty along the analytic workflow to construct composite uncertainty ellipsoids, serving as overviews of introduced uncertainty, and enabling the analyst to inspect the uncertainty in the value space. Von Landesberger et al. [LFR17] motivated the “long-standing challenge” of allowing analysts to check parameter settings and understand uncertainty “stack-up” over the course of a processing workflow or pipeline. Bernard et al. [BHR⁺19] used uncertainty quantification methods to construct graphs of overall uncertainty introduced along a MVTs pre-processing pipeline.

Using uncertainty in systems to have more informed insights into the qualitative aspects of multivariate datasets is a powerful feature of uncertainty-aware VA. Uncertainty information is often used to capture additional information on quality variations in simulation and measurement data. For ensemble computation in forecasting, Chen et al. [CZC⁺15] extended the usual descriptive statistics measures used for assessing the mean of different ensembles by employing distributions. They can be used to investigate patterns of uncertainty for individual data objects and variables. In addition, uncertainty-aware similarity/dissimilarity projection to a 2-dimensional plane allows visual identification of clusters.

Zuk and Carpendale [ZC07] proposed employing visual representations for reasoning to address uncertainty in the analytical process to make it comprehensible by collaborating users and enable exposing analytic gaps. Their typology of uncertainty to reasoning

extends Thomson et al.'s [THM⁺05] typology for visualizing uncertainty. The influence of uncertainty from the analytic process is based on analysts' confirmatory evidence and how they reason under the awareness of uncertainty [SSS⁺14].

In their review of uncertainty in data visualization, Brodlie et al. [BAL12] discussed the uncertainty introduced when generating visualizations, since a certain error is introduced into the visualization model even if we are certain about the data. Holzhüter et al. [HLS⁺12] presented an approach for visualizing uncertainty in biological expression data, addressing visual uncertainty. They identified uncertainties introduced by data acquisition – due to signal to noise ratio –, data transformation – due to processing –, and visualization – due to limited resolution and overplotting – to be relevant in their system. An overview shows both the visualization and data uncertainty and lets analysts go through multiple differently parameterized sets of expressions, and allowing exploration of the associated uncertainties. In their detail view, the uncertainty information complements the actual data and lets analysts assess the sources of uncertainty for individual results. This allows detailed exploration of the sources of uncertainty.

In an effort to understand how data workers cope with uncertainty, Boukhelifa et al. [BPHE17] analyzed workflows of study participants describing their interactions and encounters with uncertainty. One of the findings was the close relation between uncertainty characterization and data manipulation. Participants' coping mechanisms differentiate between active strategies – *understand*, *minimize*, *exploit*, and *ignore* underlying uncertainties – and tacit strategies which reflect domain practices and perceptions. These strategies could be found in the above mentioned works, where the active coping mechanism was supposedly supported to solve visual analysis tasks.

2.5.1 Decision-Making under Uncertainty

Sacha et al. [SSK⁺16] extensively covered the human factors in the knowledge generation process and related uncertainty directly to trust building. This was formulated in guidelines and, among others, included support for uncertainty-aware sensemaking, and leveraging human behavior to derive bias or trust-issues. Efforts to give insights into trust and knowledge generation have been pursued recently where Dasgupta et al. [DLW⁺17] evaluated trust in their analysis comparing a specific VA tool to conventional analysis methods, with the goal to allow analysts perform high-level sensemaking and interpretation tasks in a mixed-initiative system design. Similar to Hullman et al. [HQC⁺19], Kinkeldey et al. [KMRS17] provided a categorical overview of of uncertainty visualization, with an emphasis on decision-making. They differentiated between the type of uncertainty, visualizations, methodology, participant expertise, tasks, and effects. They give recommendations regarding study focus and design, evaluation methodology, effects under evaluation, the choice of appropriate tasks, the role of expertise, as well as decision-making theory. Fernandes et al. [FWM⁺18] explored decision-making for transit using uncertainty displays, where they found that uncertainty informed decision-making produced higher quality decisions. What they also found was that decision quality can improve over time.

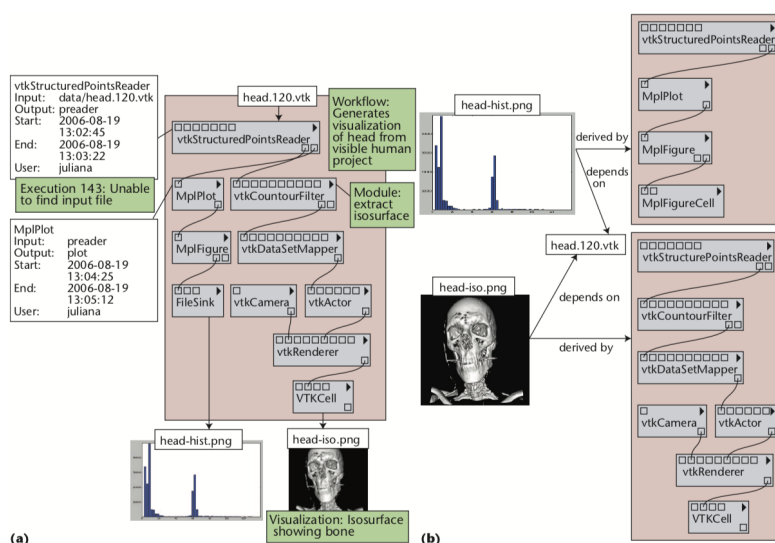


Figure 2.40: Provenance collected from an iso-surface visualization workflow [FKSS08].

It could be observed that VA systems were predominantly evaluated in qualitative studies, which may be more suited to capture more detailed insights and get results based on informed participants due to the ability to give them extensive introductions into the evaluated system. In particular decision-making and trust in data, knowledge generated, and insights is difficult to be quantified in study design, while it is important to understand how participants comprehend uncertainty and use it to refine their mental model of the data and analysis.

Visual Analytics Methods for Data and Interaction Provenance

In Section 2.1.4, provenance in data analysis has been a relevant topic of research in the fields of database management and scientific computation in the form of data provenance or data lineage. In early works, provenance of scientific and computational workflows [CFS⁺06, FKSS08] and interaction [GS06b] was used to support exploration. Using visualization and interactive exploration techniques, they served the purpose to share insights analysts gained, and facilitate the understanding of visual exploration processes and analysis workflows. Recently, in the field of VA, approaches have adopted provenance capture for more than just data lineage and the history of changes to data. Provenance is generated from visualization, interaction, insight, and rationale [RESC16]. Andrienko et al. [ALA⁺18] proposed use of provenance to not only map the VA process but to use it to externalize the mental model of the analyst in the form of prior knowledge. In visual-interactive systems, provenance can be integrated in different ways. Caching systems can observe operations and actions in a system to derive provenance [FSC⁺06], or operations self-invoking provenance storage. In the upcoming sections I will give an

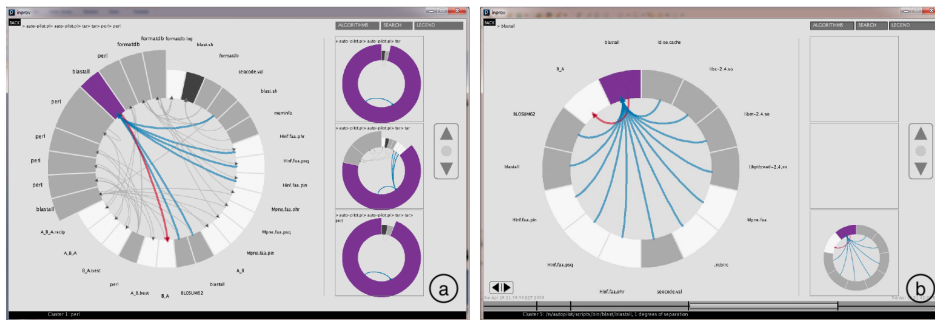


Figure 2.42: Radial provenance layout of file system browsing [BYB⁺13]: (a) shows the file system and bash commands grouped by their sub-nodes allows inspecting connections, (b) shows the file system grouped by time, filtering unassociated nodes for a particular command.

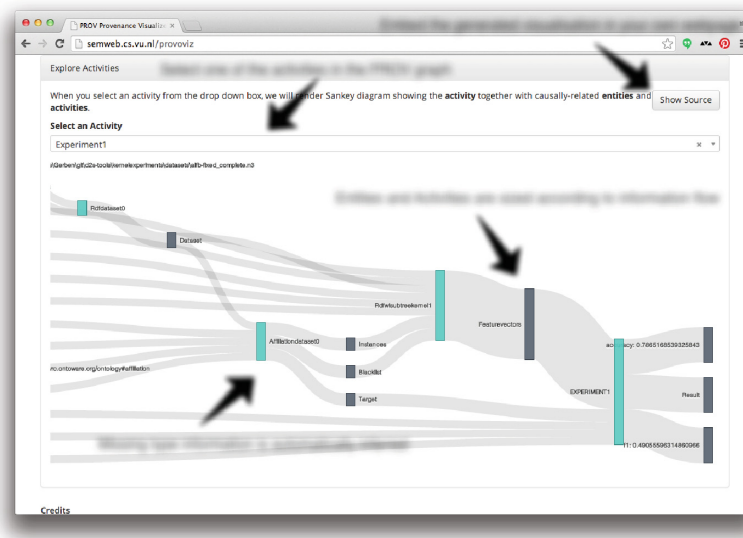


Figure 2.43: Provenance trace, capturing file usage and extracting inherent provenance if available. Particular activities within the provenance can be selected to only show the sub-graph for this activity [HG15].

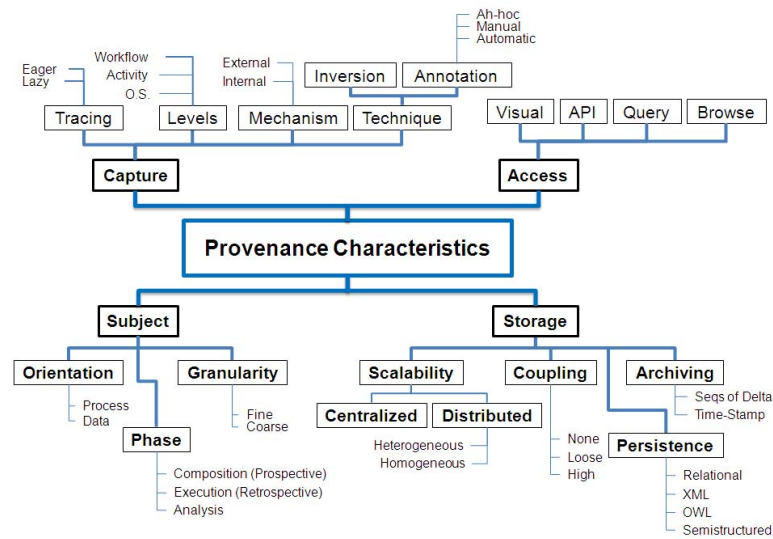


Figure 2.44: Taxonomy of provenance system characteristics [dCCM09].

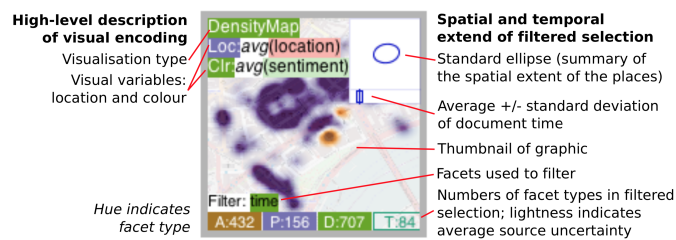


Figure 2.45: Graphical summary of a bookmark including meta-information about visual encodings, filtering, and other metrics [WSD⁺13].

Due to the directed flow of operations through workflows or processing, directed layouts are used. However, also other characteristics of provenance and provenance systems require consideration. The taxonomy of provenance by Cruz et al. [dCCM09] shows these system characteristics for scientific workflows (see Figure. 2.44). On-demand, contextual information can be displayed to the analyst, showing meta-information at fine granularity, retaining the overview of the entire graph while still giving in-depth information [HSN13]. Stitz et al. [SLSG16] employed aggregation methods for summarizing redundant or recurring operations, time-based filtering or de-emphasis of nodes based on less or more recent actions.

Summary graphics are used to share the development of data and visualizations throughout the analysis process [WSD⁺13]. Using such visual summaries (compare 2.45) allows analysts to revisit previously conducted processes and verify past interpretations (potentially made by others) and conduct visual analysis of previous graphical representations. *GraphTrail* allows analysis of visual exploration workflows [DHRL⁺12]. The visualization states are retained, small graph representations and interactions show user traces and

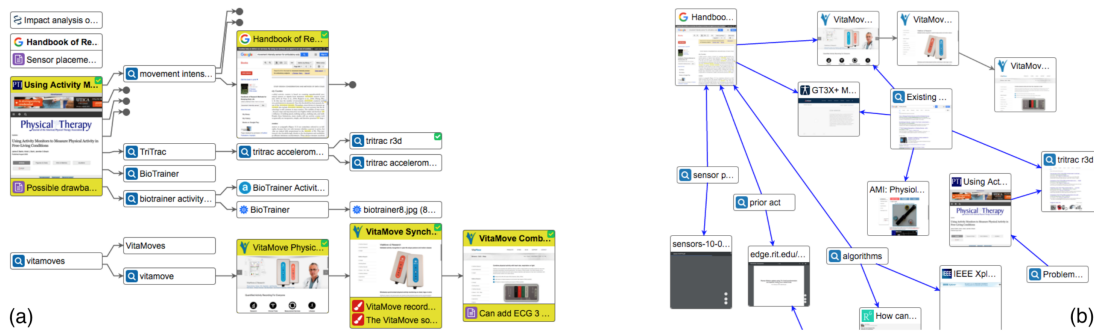


Figure 2.46: Linked views used in *SenseMap*: (a) The History Map shows user actions combined with data provenance to capture and visualize the sensemaking process, (b) The Knowledge Map allows curating information sources to use in the user’s analysis task [NXB⁺16].

paths of analyses. Summary graphics can be put into visual histories to represent the actual states of the data and visualizations as *visual provenance aid* at different levels of detail [RGT15]. This can be important if large sets of data are used during processing, which makes it necessary to perform an abstraction in order to maintain history to conduct analysis. Analytic provenance is used to support and facilitate collaboration and visualization design, using detailed breakdown of user actions, for example, querying, filtering, and transforming data for visualization [LWPL11, GZ09]. Brown et al. [BLBC12] use interaction provenance to construct a view that signals the analyst’s progress towards finding an appropriate distance function in point distributions. Changes are directly highlighted in the point distribution scatterplot. Nguyen et al. [NXB⁺16] capture and visualize the sensemaking process in *SenseMap* (see Figure 2.46), a tool that visualizes analytic provenance from and shows it in a history or knowledge map. The visualizations are composed by the analyst to construct a cohesive narrative, consisting of analysis results, analyst notes, visualizations used, and raw data. Hence, glyphs and icons are used to differentiate between the different sources and types of provenance.

2.7 Visual Analytics Methods Leveraging and Analyzing Provenance

The comprehensive characterization of provenance in visualization and data analysis [RESC16] gives a sensible classification between purposes for provenance, which showed that provenance is most prominently utilized for recall and replication. However, tools also use provenance for presentation, collaboration, meta-analysis, and action recovery. Within the context of VA, I want to elaborate the use of provenance w.r.t. two aspects: (I) using provenance to support VA applications and systems, and (II) using VA systems to facilitate provenance analysis. Nguyen et al. [NXW14] presented a survey on analytic provenance, and classified the uses of analytic provenance for (1) supporting

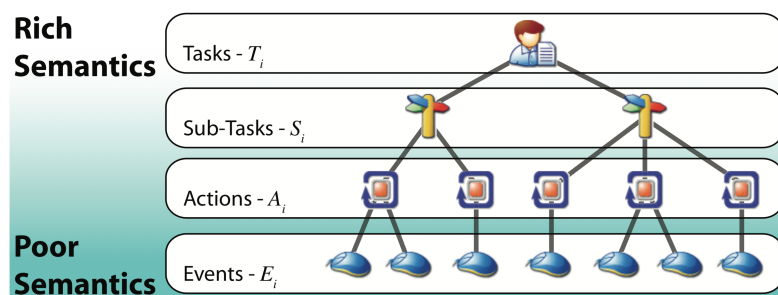


Figure 2.47: Analytic behavior can be captured at different semantic levels of granularity, based on events, actions, sub-tasks, and tasks [GZ09].

the analytical reasoning process by recalling the analytical process, reusing performed analyses, and using it as supportive evidence in constructing the reasoning process and (2) supporting collaboration through dissemination, discussion, and presentation.

Using Provenance to Support VA Applications Pirolli presented the iterative sensemaking loop [Pir05], which has since been a central element to support analysts in their analysis tasks. In an effort to aid sensemaking using interaction provenance, Endert et al. [EFN12] leveraged semantic interactions performed by the user to support the sensemaking loop by updating the exploration space depicted as a force-directed graph based on user interactions in an information foraging use case. Gotz and Zhou [GZ09] characterized actions taken during visual analysis, determining the set of operations necessary to capture visual interactive exploration using provenance. Analytic behavior can be captured on different levels of granularity and are associated to higher or lower levels of semantic actions. Such a hierarchical structure allows deriving semantically higher tasks from lower-level actions and events (see Figure 2.47). They validated their approach by implementing a visual analysis platform incorporating semantic interactions, and logging the trail of actions performed by the analyst. However, they also noted that only from actions logged it is not possible to derive insight provenance from the analyst. Analytic trails are rarely linear, analysts chain and accumulate insight from multiple trails to satisfy sub-tasks. Provenance can also be used to characterize users of VA systems, Brown et al. [BOZ⁺14] recorded mouse interactions to derive characteristics of the users and derive user groups with different traits, like personality. Captured analytic provenance can also be used to differentiate between different phases of the sensemaking model, e.g., analysts iterating between the exploration and verification phase [SBFK16]. This could be used to employ personalization, determine the level of domain expertise, and further support the analyst to make an appropriate level of tools available. In large collaborative visualization environments, visualization and interaction provenance can be captured to build a retrieval system for searching visualizations created by other users (see Figure 2.48), with certain properties, or finding similar views based on similar search criteria [SGP⁺18]. Nguyen [NXB⁺16] used analytic provenance in a sensemaking tool, capturing user’s actions to construct a history map of the sensemaking process.

2.7. Visual Analytics Methods Leveraging and Analyzing Provenance

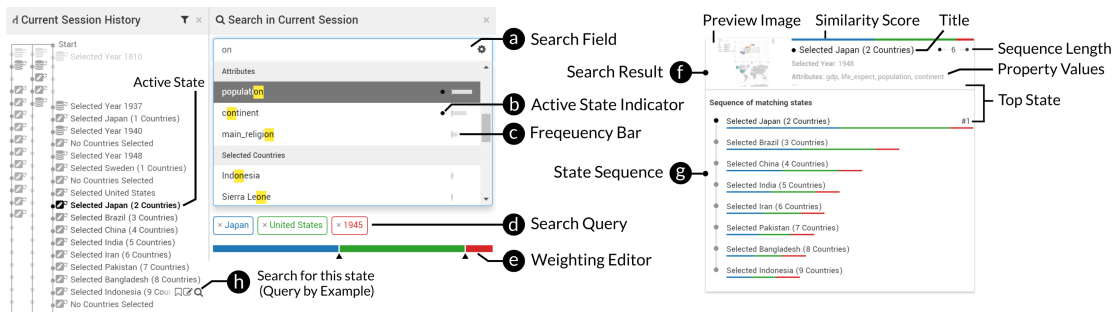


Figure 2.48: Building a retrieval system based on visualization and interactivity provenance: States are assigned with a similarity score based on visualization properties, a graphical summary (f) gives a preview on the best scored node. On the left-hand side the node is highlighted in the vertically aligned provenance graph [SGP⁺18].



Figure 2.49: A graphical history built from thumbnails of the previous visualization states and short descriptions of the performed actions/operations [HMSA08].

Users could curate relevant information, use it to organize information sources, and communicate their insights and sensemaking map to collaborators. However, effectiveness of the tool varied based on what experience the users/participants had with the tool.

For interaction and visualization provenance, research focused on constructing graphical histories of interactions to try and interpret insights. Heer et al. [HMSA08] enhanced Tableau to add a history interface to support analysis and communication of insights (see Figure 2.49). As with visual exploration systems, Tableau already featured an operation history, however, this was extended using action-based logging. Each history item is shown as a thumbnail of the operation and the corresponding view. With a large number of actions performed, the history could become cluttered and unreadable. To mitigate that, the analyst's actions are reduced in complexity, related actions are chunked, and a new action behavior called *undo-as-delete* serve as indicators for cleansing up the history. Sacha et al. [SSK⁺16] identified data and analytic provenance to predict user intent based on low-level interactions.

Provenance Analysis Using Visualization and VA Methods Carata et al. [CAB⁺14] compared various tools for exploring provenance. *ProvDMS* was developed to provide a web-based data provenance capture system including a provenance storage

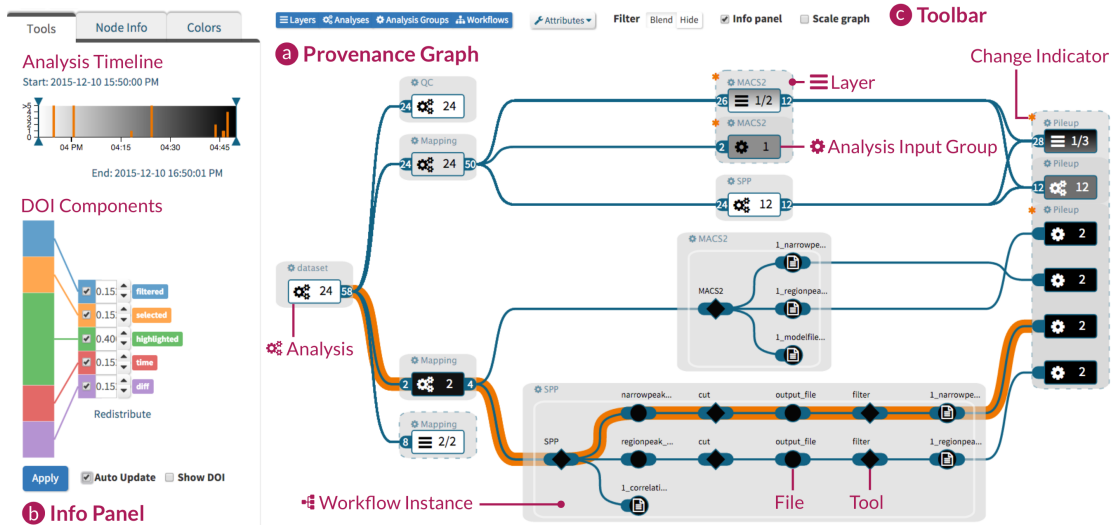


Figure 2.50: The AVOCADO workflow provenance exploration interface [SLSG16].

database [HSN13]. *Experiments* can be created to store sensor data provenance, derive experiments, and obtain the sensor status. The system uses force-directed graph layouts and a hierarchical tree graph view to give contextual information. Visualization provenance can be captured to support exploration and subsequently perform storytelling based on exploration history [GLG⁺16]. Advanced layout, filtering, and aggregation methods can be used to make provenance graphs more easily comprehensible. In *AVOCADO*, Stitz et al. [SLSG16] employed visual-interactive methods to edit degree-of-interest (DoI) functions and change provenance graph representations between *node-type-specific views* and attribute mapping (see Figure 2.50). Multi-step analysis workflows is facilitated using aggregation strategies, and time-based node coloring and filtering. Analysts can use these methods for exploring graphs based on various scenarios. In other work, Schreiber et al. [SS17] developed a comics-inspired method for generating self-explaining views of data provenance, using a consistent visual language resembling comic figures (for actors in the provenance graph) and panels (corresponding to different activities). In collaboration or analysis comparison scenarios, interaction and visualization provenance can be used to assess behavior. The graphical histories and classified actions presented by Heer et al. [HMSA08] can be used to compare analysis sessions of various users, and allows analysis of which commands and *worksheets* have been used and if there are noticeable patterns across users (see Figure 2.51).

2.8 Connecting Data Quality, Uncertainty, and Provenance

Early on, Raman and Hellerstein motivated that tracking transformations and data lineage is an important feature of their *Potter's Wheel* application [RH01]. Sacha et al. [SSK⁺16]

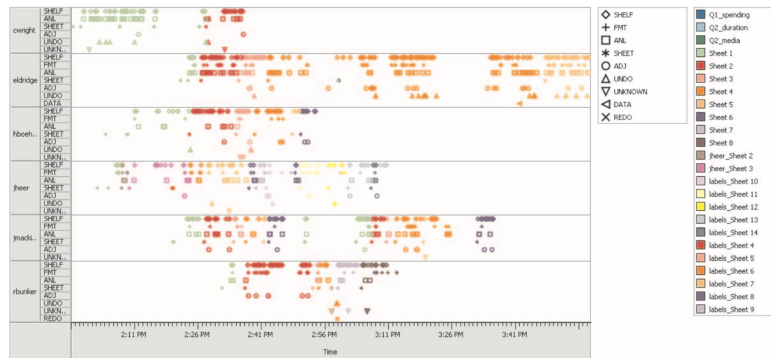


Figure 2.51: Scatterplot of provenance traces from multiple users over time shows patterns of used actions and worksheets in Tableau [HMSA08].

postulated various relations between data quality, uncertainty, and provenance. I want to elaborate the arguments and aspects described by them further to give a more specific discourse on the interrelations between those fields of research and how they can be used to amplify analysis.

Relating Data Quality and Uncertainty Different similarities between DQ and uncertainty quantification and capture can be identified: Specifically the DQ metrics presented in various forms [BS06, BS16] and types of uncertainty [MRH⁺05, THM⁺05] seem to be closely related and similar. DQ cleansing and wrangling often has a significant impact on the data. As a consequence the applied operations inevitably introduce systematic uncertainty that can be quantified knowing the data input, output, and influencing operation. That way, analysts can more closely associate the impact of operations on the data and assess if they are necessary in the extent they were applied. Domain experts who are unfamiliar with pre-processing algorithms can quantify the impact of uncertainty on downstream analysis and react accordingly.

Relating Data Quality and Provenance Herschel et al. [HDL17] classified applications of provenance, and emphasized one use of provenance to maintain process and DQ. They motivated that provenance may be used to improve quality or determine causes of quality issues. However, DQ metrics or functional dependencies need to be captured to adequately model data cleansing, profiling, and wrangling processes. To capture data lineage and provenance it is necessary to employ DQ mechanisms that allow analysts to not only draw quantitative conclusions from the data but also infer qualitative characteristics and how they have changed over the course time or during analysis. It was briefly discussed that different methods can be employed to capture provenance, either using an active approach and monitoring if operations and processes affected data, or by having processes generate provenance metadata themselves. Both ways are biased techniques if not applied comprehensively, if not all processes are monitored the captured provenance may not be complete and hence could lead to analysts drawing wrong conclusions. If

applied **correctly**, provenance systems can aid analysts conducting DQ assessment in capturing data cleansing, profiling, or wrangling traces across applications and systems. That way, it is less likely that operations performed are not tracked along the workflow and give a more complete representation of the entire DQ assessment workflow. Similar to predicting data transformations based on DQ characteristics [HHK15] and mixed-initiative wrangling [GKHH11], we can leverage insights from previously applied data operations and processed that have been stored as provenance and combine them with DQ metrics to facilitate quality improvement. Analytic and interaction provenance are both means for comprehending, understanding, and revisiting analytical processes, that should be stored and verified. It can help applying existing workflows to new datasets or scenarios by only requiring necessary changes in some steps instead of altering the entire workflow or building it up from scratch. Sharing processes could educate inexperienced collaborators, or serve as visual validation to intrigued customers.

Relating Uncertainty and Provenance Uncertainty can be observed during data generation/sampling, be generated by models or simulations, or be introduced during data processing, for example, DQ assessment [BHJ⁺14]. However, without adequate provenance capture and storage it is uncommon that uncertainty is actually retained in the data, due to potentially massive data overhead, or downstream inconveniences for handling uncertainty in the data. As a consequence, uncertainty is rarely communicated to the user unless explicitly requested by the analyst, which skews perception of the data. Adding uncertainty properties to data provenance models could allow designers and developers to leverage uncertainty in their visualization and VA approaches and can greatly improve analysts' awareness of uncertainty.

2.8.1 Summary and Conclusion

Between these fields of research I identified many parallels and mutual influences. Uncertainty is inevitably influenced by DQ assessment and pre-processing, and consequently analysts should be aware of the consequences of inadequate pre-processing. However, there is a lack of VA solutions available that associate these influences appropriately. Furthermore, the topic of data provenance has been extensively researched in the field of scientific computation and visualization, but only few approaches addressed data provenance with DQ or uncertainty aspects in mind. With my review of visualization and VA solutions in these fields, it can be seen that these methods can be beneficial to exploring and making sense of data using a VA solution that combines these similar, but yet still disconnected fields of DQ assessment, uncertainty, and provenance. The formal characteristics of time series makes time series data a good candidate for performing statistical analysis and DQ assessment. These characteristics have already been exploited in different ways for conducting quality- and uncertainty-aware analysis. However, most approaches fall short of seeing pre-processing as sources of uncertainty itself. This must be addressed in future work to make analysts more aware of the influences of pre-processing on the analysis outcome.

Part II

The Proposed Solution

Conceptualizations

In this chapter, I will lay the theoretical foundation for the VA approaches presented in the upcoming chapters, leveraging data quality metrics, uncertainty measures, and provenance concepts. Describing these concepts extends theories and principles presented in the Related Work chapter and are used as a basis for the further developed VA solutions. The concepts are used to generate information beneficial to understanding data quality and associated uncertainty, and store the provenance of developments of a dataset throughout pre-processing and DQ assessment. The conceptualizations in this chapter are elaborations of previously published works [BBGM17, BGK⁺18, BBB⁺19, BGM19, BHR⁺19].

3.1 Defining Data Quality Metrics

These conceptualizations were published in [BGK⁺18].

In Section 2.1.1 I showed existing DQ taxonomies that cover both task-dependent and task-independent metrics, and often discuss generic or domain-specific effects of DQ errors manifesting in the dataset. Task-independent DQ metrics can cover commonly occurring errors, like entry and tuple completeness [BS16], or invalid entries, violating type constraints [ORH05]. Oliveira et al. present an organizational data model and aptly illustrate the potential sources of DQ errors (compare Figure 2.5). This shows the broad scope where DQ metrics have to be employed to ensure adequate coverage of error detection. Within the extent of this thesis a DQ metric is defined to be the quantified measure of a DQ dimension (compare Section 2.1.1) that gives quantitative information about the lack of quality w.r.t. a certain data property. Generic data quality models [DDG⁺16] allow for flexible implementation and metric data types, for example, Boolean values if requirements have been met, or numeric values if a metric expresses a quantitative measure. Hence, how the lack of quality is measured or determined can be subject to the specific DQ dimension and implementation.

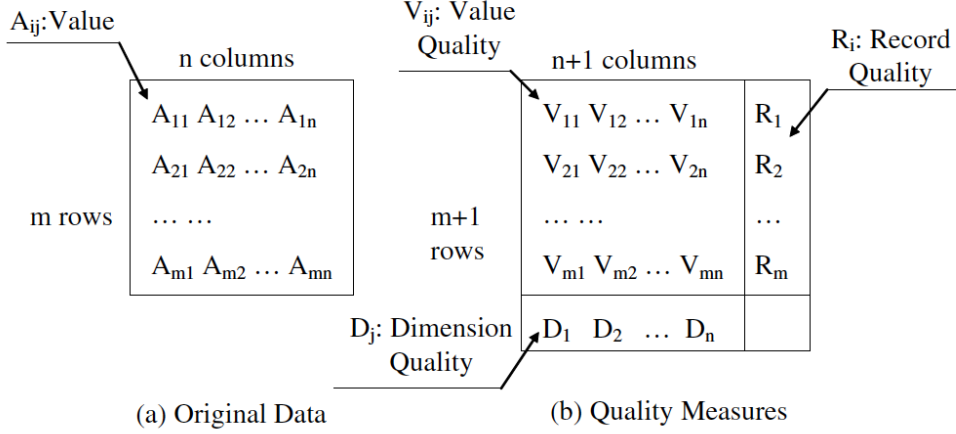


Figure 3.1: Conceptual structure of annotated quality measures for tabular data. (a) shows the column and row structure of the original dataset, (b) shows calculated quality for individual cells of the data and subsequently aggregate column and tuple quality measures [XHWR06].

The data type predominantly analyzed for DQ assessment is tabular (or relational) and time-oriented data, hence the metrics presented in this section will cover quality dimensions of these types. We calculate a DQ metric across entries of tabular datasets according to Xie et al. [XHWR06].

For column-wise metric computation, we follow the quality measure notation used in Figure 3.1, i.e., D_i , a quality metric Q_D of a quality dimension D and a column $j \in S$ accumulates the measures of quality $V_{i,j}$ for each value $A_{i,j}$ of the dataset S .

$$Q_D(j) = \sum_{i=0}^n V_{i,j}$$

Specifically, I define the measure of quality $V_{i,j}$, also referred to as the dirtiness of an entry (compare [GGAM12]), to be determined by validation function calls vf , evaluating a value $A_{i,j}$ against a specific quality criterium c .

$$V_{i,j}^c = \text{vf}_c(A_{i,j})$$

DQ metrics potentially have multiple validation criteria associated with them ($\text{vf}_{c_1} \dots \text{vf}_{c_m}$), so multiple functions can be logically concatenated. As previously mentioned, these functions can return any dirtiness value. The aggregation function VF is specified to return a proportional value between 0 (not dirty) and 1 (dirty).

$$\text{VF}_{i,j}(A_{i,j}) = \bigcup_{m=0}^M \text{vf}_{c_m}(A_{i,j}) \in [0, 1], \text{ for } \bigcup \in [\wedge, \vee, \neg, \oplus]$$

$[\text{VF}_{1,1,1}, \dots, \text{VF}_{1,1,k}]$	$[\text{VF}_{1,2,1}, \dots, \text{VF}_{1,2,k}]$	$[\text{VF}_{1,3,1}, \dots, \text{VF}_{1,3,k}]$	\dots	$[\text{VF}_{1,j,1}, \dots, \text{VF}_{1,j,k}]$
$[\text{VF}_{2,1,1}, \dots, \text{VF}_{2,1,k}]$	$[\text{VF}_{2,2,1}, \dots, \text{VF}_{2,2,k}]$	$[\text{VF}_{2,3,1}, \dots, \text{VF}_{2,3,k}]$	\dots	$[\text{VF}_{2,j,1}, \dots, \text{VF}_{2,j,k}]$
\dots	\dots	\dots	\dots	\dots
$[\text{VF}_{i,1,1}, \dots, \text{VF}_{i,1,k}]$	$[\text{VF}_{i,2,1}, \dots, \text{VF}_{i,2,k}]$	$[\text{VF}_{i,3,1}, \dots, \text{VF}_{i,3,k}]$	\dots	$[\text{VF}_{i,j,1}, \dots, \text{VF}_{i,j,k}]$
$Q_{1,1}$	$Q_{2,1}$	$Q_{3,1}$	\dots	$Q_{j,1}$
$Q_{1,2}$	$Q_{2,2}$	$Q_{3,2}$	\dots	$Q_{j,2}$
\dots	\dots	\dots	\dots	\dots
$Q_{1,k}$	$Q_{2,k}$	$Q_{3,k}$	\dots	$Q_{j,k}$

(a) DQ metrics create a matrix of quality validation function arrays for each value $A_{i,j} \in S$.

$\text{VF}_{1,c_1,1}$	\dots	$\text{VF}_{1,c_j,1}$	\dots	$\text{VF}_{1,c_1,1}$	\dots	$\text{VF}_{1,c_j,1}$
$\text{VF}_{2,c_1,1}$	\dots	$\text{VF}_{2,c_j,1}$	\dots	$\text{VF}_{2,c_1,1}$	\dots	$\text{VF}_{2,c_j,1}$
\dots	\dots	\dots	\dots	\dots	\dots	\dots
$\text{VF}_{i,c_1,1}$	\dots	$\text{VF}_{i,c_j,1}$	\dots	$\text{VF}_{i,c_1,1}$	\dots	$\text{VF}_{i,c_j,1}$
$Q_{D_1,C}$			\dots	$Q_{D_k,C}$		

(b) Spanning DQ metrics create an array of validation function matrices spanning multiple columns $C = [c_1, \dots, c_j]$.

Table 3.1: Tables showing arrays of quality validation functions for single column and spanning column DQ metrics: (a) describes the structure of single column DQ metrics, (b) extends this structure to create spanning DQ metrics.

Subsequently, the value of a DQ metric Q_D is the normalized measure of all validation criteria VF_c with $c \in 1, \dots, m$ for all values $A_{i,j}$ of column j .

$$Q_D(j) = \sum_{i=1}^N \bigcup_{m=0}^M \text{vf}_m(A_{i,j}) \in [0, 1], \text{ for } \bigcup \in [\wedge, \vee, \neg, \oplus]$$

This gives us an array matrix of quality validation functions $\text{VF}_{i,j,k}$ for values $A_{i,j} \in S$ and DQ metrics $Q_{j,k}$ (see Table 3.1a). It is also possible to construct a DQ metric employing validation functions across columns $C = [c_1, \dots, c_j]$ resulting in an array of validation function matrices (see Table 3.1b).

Revisiting the definitions in Sections 2.1.1 and 2.1.1, I stated DQ metrics represent concrete implementations of DQ dimensions, mapping measures of quality related to concrete use cases and domain-specific context. It is often not possible to account for specific manifestations of DQ errors in a generic way. In the upcoming sections I will describe conceptual methods for determining the quality of tabular and time-oriented data.

3.1.1 Metrics and Data Quality Dimensions of Tabular Data

Tabular data has particular characteristics that allow for providing generic DQ metrics to quality certain DQ dimensions. For example, information on column data types and empty data indicators will allow generalized validation functions to be implemented.

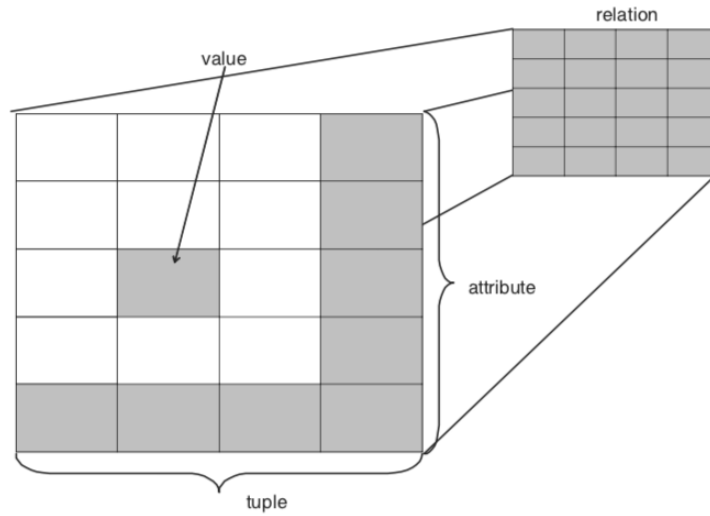


Figure 3.2: Types of completeness in tabular and relational data.

Leveraging this information will allow us to develop different types of column-wise metrics for the following DQ dimensions: *completeness*, *validity*, and *plausibility*. A generic tuple-wise DQ metric across multiple columns can evaluate the *uniqueness* of entries.

Completeness/Missingness. The completeness of a dataset defines the degree to which data values represent the real world, as far as it can be matched in the data structure. In DQ research it mainly refers to missing values, hence the term missingness can be used in analogy. The following definitions are based on Batini et al.'s [BS06] definitions of completeness of relational data. For tabular data we can compute different types of completeness: *Value completeness*, *tuple completeness*, *column completeness*, and *relation completeness*.

Value Completeness identifies an entry A_i as dirty if it is either empty (missing) or marked as empty through a syntactic identifier, e.g., NaN in R or *matlab*.

$$\text{Comp}_v(A_i) = \begin{cases} 0 & \text{if } A_i = \text{null} \text{ or } A_i = \{\text{NaN}, -, \dots\} \\ 1 & \text{else} \end{cases}$$

Tuple Completeness characterizes the completeness of a tuple T for all its respective values $A_i \in T$. It quantifies the ratio of complete values and the total number of attributes m .

$$\text{Comp}_t(T) = \frac{\sum_{i=1}^m \text{Comp}_v(A_i)}{m}$$

Column Completeness (or *attribute completeness* [BS06]) characterizes the completeness of a single column C_j for all values $A_{i,j} \in C_j$. Again, the completeness is quantified as

the ratio of complete values to the total number of values n in C_j .

$$\text{Comp}_c(C_j) = \frac{\sum_{i=1}^n \text{Comp}_v(A_{i,j})}{n}$$

Relation Completeness quantifies *column completeness* across all attributes m of a dataset S .

$$\text{Comp}_r(S) = \frac{\sum_{j=1}^m \text{Comp}_c(A_{i,j})}{m} = \frac{\sum_{j=1}^m \sum_{i=1}^n \text{Comp}_v(A_{i,j})}{m \cdot n}$$

Identifier Completeness characterizes a more holistic view of missingness in data. Identifiers available outside the relational and tabular dataset can act as indicators of missing tuples. For that, an array of identifiers $I = \langle i_1 \dots i_k \rangle$ is used to compute the ratio of completeness, where I are attributes of S and K is the size of the array of identifiers.

$$\text{VF}_{\text{comp}_i}(i_k, C) = \begin{cases} 1 & \text{if } i_k \in C = [A_{1,c}, \dots, A_{i,c}] \\ 0 & \text{else} \end{cases}$$

$$\text{Comp}_i(I, C) = \frac{\sum_{k=1}^K \text{VF}_{\text{comp}_i}(i_k, C)}{k}$$

Validity. Invalid entries might impede calculations or skew statistical evaluations. The reasons for data being invalid are highly diverse and context-dependent [BG05]. Identifying values as invalid is a task that demands comprehensive domain knowledge, which can rarely be performed automatically but must be done by the user. However, it is possible to perform initial general validation in the form of data types. Such a general validity metric includes a check to evaluate if a value $A_{i,j}$ complies with the automatically detected, or manually specified data type of the column.

$$\text{VF}_{\text{valid}}(A_{i,j}, \text{type}) = \begin{cases} \mathbf{0} & \text{if } \text{typeOf}(A_{i,j}) = \text{type}, \text{ for } \text{type} \in \{\text{numeric}, \text{string}, \text{date}, \dots\} \\ \mathbf{1} & \text{else} \end{cases}$$

Similar to the completeness metric, it is possible to quantify this metric for individual values VF_{valid} , tuples Valid_t , columns Valid_c , or relations Valid_r , i.e., across all attributes of the dataset.

$$\begin{aligned} \text{Valid}_t(T) &= \frac{\sum_{i=1}^m \text{VF}_{\text{valid}}(A_i)}{m} \\ \text{Valid}_c(C_j) &= \frac{\sum_{i=1}^n \text{Valid}_v(A_{i,j})}{n} \\ \text{Valid}_r(S) &= \frac{\sum_{j=1}^m \text{Valid}_c(A_{i,j})}{m} = \frac{\sum_{j=1}^m \sum_{i=1}^n \text{Valid}_v(A_{i,j})}{m \cdot n} \end{aligned}$$

Format compliance can also be extended into value compliance for specific data types. For example, ensuring that numeric values are only positive, or validating string characters are valid according to a particular coding, like UTF-8. But such context-specific validity constraints should be integrated by domain experts if necessary.

Plausibility. Statistical measures make it possible to gather distribution information about numeric attributes in tabular and relational datasets and subsequently get insights of implausible and extreme entries. Such entries might manifest in datasets due to erroneous data generation (e.g., human-created values) or inconsistent sources (e.g., different sensor calibration) [GGAM12]. A plausibility metric could detect outlying entries by using non-robust (statistical mean \bar{x}_{col} , and standard deviation $std(X_{col})$) or robust statistics measures (median \tilde{x}_{col} , and a robust inter-quartile range estimator $s_{IQR} = \frac{IQR}{1.35}$) to determine extreme values.

Uniqueness. Relational and tabular data often contain unique key attributes or attribute pairs, which should not be duplicate. If these key attribute columns C_1, \dots, C_j are specified, it is possible to check for potentially duplicate tuples.

$$\text{Uniq}_t(C_1, \dots, C_j) = \begin{cases} \text{true} & \text{if } \forall x \in M : M(x) = 1, \text{ for} \\ & M = \{\{A_{i,j} | A_{i,j} = (A_{i,C_1}, \dots, A_{i,C_j}) \text{ for } i = 1 \dots n\}\} \\ \text{false} & \text{else} \end{cases}$$

Time Interval Metrics. When analyzing time-oriented data, the validation of intervals usually requires prior transformation steps to explicitly determine the interval duration. The interval metric evaluates a specified interval without making changes to the data necessary. It allows for checking if the interval $v_{col_b, row} - v_{col_a, row}$ is smaller than, larger than, or equal to a given duration value, or both larger than and smaller than a duration d . Additionally, a second metric allows performing outlier detection on interval lengths.

$$\text{VF}_{interval}(A_{i,col_a}, A_{i,col_a}, d, \triangleright) = \begin{cases} \text{true} & \text{if } (A_{i,col_a} - A_{i,col_a}) \triangleright d, \text{ for } \triangleright \in \{<, \leq, >, \geq, =\} \\ \text{false} & \text{else} \end{cases}$$

Temporal and Value Outlier Detection. With robust outlier detection measures, outliers can be automatically identified and highlighted. However, judging if these outliers – either in the temporal domain or in the data domain – represent anomalies requires additional contextual information. Thus, it takes the user’s domain knowledge to reason about the identified outliers. As such, marking outliers as well as rasters which contain outliers and saving this meta information for subsequent analysis is advisable and allows more informed decisions.

Missing Timestamps and Temporal Values. Similar to tabular and relational data, empty intervals can signal quality issues, and more specifically for rastering tasks could imply inappropriate raster window size. The distribution and amount of empty rasters can be visually inspected for finding a suitable rastering.

3.2 Uncertainty in Time Series Pre-Processing

These conceptualizations were published in [BBB⁺19].

The conceptualization of uncertainty in MVTS presented in this section is based on probabilistic uncertainty modeling presented by Bonneau et al. [BHJ⁺14]. Even though pre-processing inevitably introduces uncertainty by altering the original data, these routines are rarely analyzed towards their impact on uncertainty. When analyzing MVTS, pre-processing is an integral part to enable further analysis.

3.2.1 Sources of Uncertainty

How uncertainty was introduced into the data is distinguished by the different sources of uncertainty (compare Section 2.1.3), including observations inherent to the data, generated by models or simulations, or introduced by the processing or visualization processes [PRJ12, BHJ⁺14]. Several approaches analyze uncertainty introduced by pre-processing [CCM09, WYM12], aggregating uncertainty for individual processing steps. When assessing the influence of uncertainty on MVTS, inappropriate aggregation could omit temporal characteristics that can also be affected by processing (e.g., rastering [BBGM17], or sampling).

In the previous section, I defined measures of quality for time series based on domain-specific characteristics, specifically for intervals and rastering transformations. To assess quality and the impact of pre-processing and transformations on time series in a more general application, a generic model can ensure that quality assessment is enabled for more types of time series processing operations. To allow this I define a model of uncertainty quantification for MVTS data. The model is based on three dimensions of a Quantification Cube, shown in Figure 3.4a: time and variables of a MVTS, and pre-processing steps. One aspect that becomes apparent from the previous example of rastering is that frequently aggregation is occurring during pre-processing, which also needs to be considered during quantifying changes as uncertainty. The measures of uncertainty are stored as additional dimensions for every attribute/dimension of the MVTS, similar to the previously defined DQ metrics.

3.2.2 Quantifying Uncertainties

I refer to a p -dimensional time series by $\mathbf{X} = \{\mathbf{x}_{(t_1,v)}, \dots, \mathbf{x}_{(t_n,v)}\}$ measured at time point t_1, \dots, t_n with variables $v = 1, \dots, p$ (compare Figure 3.4b). A pre-processing pipeline for MVTS consists of m pre-processing steps that modify the MVTS and introduce uncertainty. Each pre-processing step s takes a MVTS $\mathbf{X}_{s-1} = \{\mathbf{x}_{(t_1,v,s-1)}, \dots, \mathbf{x}_{(t_n,v,s-1)}\}$

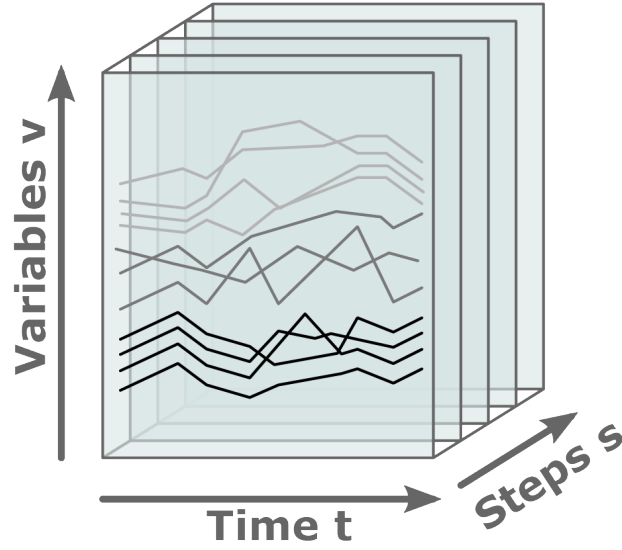


Figure 3.3: The time and variables (also referred to as data dimensions) of the MVTS, and the pre-processing steps span a cube of dimensions influencing uncertainty introduced by MVTS pre-processing.

as input and generates a modified MVTS $\mathbf{X}_s = \{\mathbf{x}_{(t_1,v,s)}, \dots, \mathbf{x}_{(t_n,v,s)}\}$ which is the input of the next step. \mathbf{X}_0 is the MVTS as input to the whole pre-processing pipeline, \mathbf{X}_m the resulting MVTS, and \mathbf{X}_s with $s = 1, \dots, m - 1$ the MVTS between the single pre-processing steps. The natural atomic representation of uncertainty for such a processing step is determined by the quantification function $u(\mathbf{X}_s, \mathbf{X}_{s-1})$ that computes the uncertainty per timestamp and variable $u(x_{(t,v,s)}, x_{(t,v,s-1)})$. However, depending on the pre-processing operation, the uncertainty quantification can only be done on a specific level of granularity, if the temporal domain or the dimensionality of the MVTS are affected. In the following I discuss the different cube dimensions' dependencies on quantification.

Dependency on Variables. If MVTS variables are individually analyzed, it is sufficient to determine the absolute value difference between the input and output time series of a pre-processing step: $u_{abs}(abs(z_{(t,v)}))$, where

$$z_{(t,v)} = x_{(t,v,s)} - x_{(t,v,s-1)}$$

denotes the value difference. This results in an uncertainty value that is value domain dependent, as it needs to be considered in the context of the respective scale of the value domain. Thus, if uncertainties of variables with different value domains are to be compared or assessed simultaneously, normalized relative differences need to be

determined instead

$$u_{rel}(z_{(t,v)}) = \frac{z_{(t,v)} - \mu_z}{\sigma_z}$$

where μ_z is the mean difference and σ_z the deviation.

This way, the influence of multiple variables on the uncertainty at time $x_{(t,s)}$ is comparable for any v . If the uncertainty of each variable cannot be quantified for single time points, the uncertainty needs to be computed for single variables across all time points $u_t(x_{(v,s)}, x_{(v,s-1)})$. This is for example the case, if the temporal space is modified, like temporal sampling or rastering (only u_v is applicable).

Dependency on Time. The quantification of uncertainty over single time points and dimensions $u(x_{(t,v,s)}, x_{(t,v,s-1)})$ allows to identify time points or time ranges that have a high, low, or normal level of uncertainty in the value domain. If the uncertainty of time points cannot be quantified for single variables, the uncertainty needs to be computed for single time points across all variables $u_v(x_{(t,s)}, x_{(t,s-1)})$. This is for example the case, if the time series dimensionality is altered, e.g., by dimensionality reduction routines (only u_t is applicable). In the case of aggregating over time (see Section 3.2.3), e.g., for rastering or sampling a time series to a coarser temporal granularity, the uncertainty introduced in the temporal domain needs to be considered in the quantification. This can be done by computing the relative or absolute temporal differences Δt of all time points that are merged in the raster intervals of the coarser granularity level, similarly to computing relative value differences formalized for variables, but in the temporal domain.

Dependency on Pre-Processing Steps. Each pre-processing method has different effects on the introduced uncertainty. However, these effects can be derived when taking into account the error that is introduced by the specific method and its parametrization. Moreover, this on average introduced error can be estimated (e.g., moving average changes the value domain consistently). We formalize the introduced uncertainty accordingly: $u_{err}(x_{(t,s)}) = f_{err}(x_{(t,s)}, \mathbf{k})$, where f_{err} is an error function for quantifying uncertainty, and $\mathbf{k} = \{k_1, \dots, k_l\}$ is the current parameter vector of the pre-processing method.

3.2.3 Aggregating Uncertainties

Figure 3.4c illustrates the different types of aggregation of uncertainties over all processing steps. As with quantifying uncertainty, aggregation can be applied on all of the cube's dimensions: time and variables of the MVTs, and pre-processing steps. Generally it is advisable to quantify uncertainty at the finest granularity level and aggregate to coarser granularities if necessary. A general $agg_{i=1}^n(\cdot)$ function indicates a generic aggregation function, because various aggregation methods could be applied, or interchangeably used. More specifically this can be a simple summarization $\sum_{i=1}^n(\cdot)$, a multiplication $\prod_{i=1}^n(\cdot)$, or other statistical aggregations of uncertainty, like the mean uncertainty $\mu(u)$, mean squared uncertainty $\mu(u^2)$, or root mean squared uncertainty $\sqrt{\mu(u^2)}$. Figure 3.4c shows different aggregation methodologies that prioritize aggregation (c_1) by time, (c_2) by

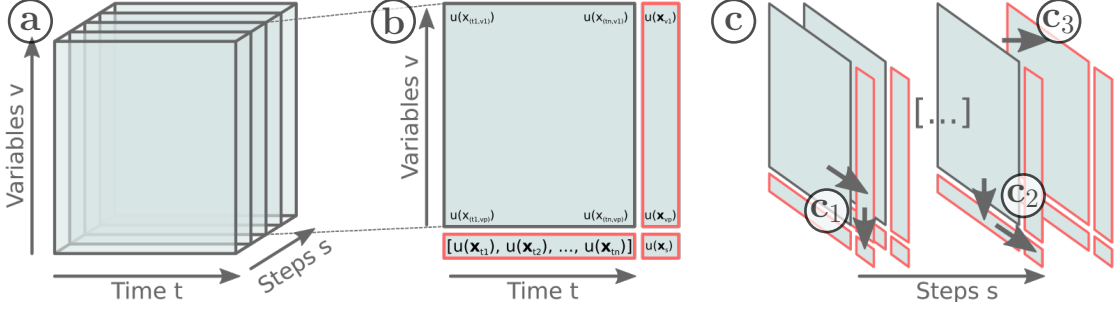


Figure 3.4: Illustration of quantification of uncertainties and aggregation on values and uncertainties. (a) shows the three variables time, variables, and processing steps. (b) represents a single processing step slice with the dimensions time and variables together with the uncertainty aggregation, either across time or variables (shown as red boxes). (c) indicates the different aggregation paths within single processing step slices (c_1 , c_2) and across all steps (c_3).

variable, or (c_3) by pre-processing step. Consecutively, other aggregation steps can be added. In Figure 3.4c this is marked by the perpendicular arrows, where aggregation is performed by time and variable (c_1), and by variable and time (c_2), respectively.

Aggregating by Time. Quantifying uncertainty on timestamp (or entry level $A_{i,p}$) granularity is not always beneficial. Analogous to visualization of large MVTS, aggregating uncertainty to a coarser temporal granularity allows maintaining a representative dataset if the scale of the original data is too large. Aggregating uncertainty can be done on different levels of temporal granularity. To remove the temporal dimension from the quantified uncertainty, we can aggregate over the entire time dimension $u(\mathbf{x}_{(v,s)}) = \text{agg}_{t=1}^n(u(x_{t,v,s}, x_{t,v,s-1}))$. This allows an abstract representation of uncertainty without time, e.g., a single value of uncertainty for an entire time series variable v , and pre-processing step s .

Aggregating by Variables. Analyzing uncertainty of individual variables allows detailed inspection of effects on the value domain. However, variables can be affected differently by pre-processing. Uncertainty can be aggregated by variables $u(\mathbf{x}_{(t,s)}) = \text{agg}_{v=1}^p(u(x_{t,v,s}, x_{t,v,s-1}))$ to determine a single value of uncertainty for these variables, e.g., $\mu(u(\mathbf{x}_{(t,s)}))$.

Aggregating by Pre-Processing Steps. To obtain an overview of uncertainties for one step s of the pre-processing, we compute the uncertainty of each pre-processing step $u(\mathbf{x}_s)$. Comparison of different steps can be done on different levels of aggregation, by variable:

$$u(\mathbf{x}_{(t,s)}) = \text{agg}_{v=1}^p(u(x_{t,v,s}, x_{t,v,s-1}))$$

or time:

$$u(\mathbf{x}_{(v,s)}) = \text{agg}_{t=1}^n(u(x_{t,v,s}, x_{t,v,s-1}))$$

However, it is also possible to aggregate over a whole pre-processing pipeline, to assess the introduced uncertainty of a sequence of pre-processing steps:

$$u(\mathbf{x}_{(t,v)}) = \text{agg}_{s=1}^m(u(x_{t,v,s}, x_{t,v,s-1}))$$

To enable more distinct assessment, aggregation can be nested consecutively. Aggregating by variables allows comparison over time:

$$u(\mathbf{x}_{(t)}) = \text{agg}_{v=1}^p \text{agg}_{s=1}^m(u(x_{v,t,s}, x_{v,t,s-1}))$$

This allows more detailed inspection if the time series was affected by pre-processing uniformly. Conversely, aggregating by time allows comparison over variables:

$$u(\mathbf{x}_{(v)}) = \text{agg}_{t=1}^n \text{agg}_{s=1}^m(u(x_{v,t,s}, x_{v,t,s-1}))$$

Ultimately, aggregating over time, variables, and pre-processing steps produces a single value of uncertainty for the entire pre-processing pipeline (compare Figure 3.4c₃):

$$u(\mathbf{x}) = \text{agg}_{t=1}^n \text{agg}_{v=1}^p \text{agg}_{s=1}^m(u(x_{v,t,s}, x_{v,t,s-1}))$$

3.3 Data and Insight Provenance from Data Quality

These conceptualizations were published in [BGM19].

Implementing data quality metrics (compare Section 3.1) allows for detecting quality issues in a dataset. They can serve as a measure of overall quality for a dataset. However, to understand the impact data transformations and pre-processing operations have on data quality, it is necessary to capture the state of data quality for multiple points in time. Descriptive information of the data's quality is required to audit wrangling operations and assess if they were applied appropriately. By logging what actions were used alongside the wrangling process (e.g., data profiling, filtering, cleansing), it is possible to gain understanding of employed transformations, and make sense of the DQ assessment process. I propose to employ measures of quality throughout each processing step to allow judgment if quality was affected throughout the wrangling process. To store these types of information, I present a generic model of provenance generation for data wrangling (see Figure 3.5).

The different entities incorporate different types of provenance (according to [SPG05]). The main entities involved are the data, and correspondingly data revisions, generating data provenance, being generated by transformations, generating workflow provenance. The data can be filtered by a condition into a working dataset. We store the information on each revision, capture which filters were applied, and derive data descriptions to annotate the corresponding revision.

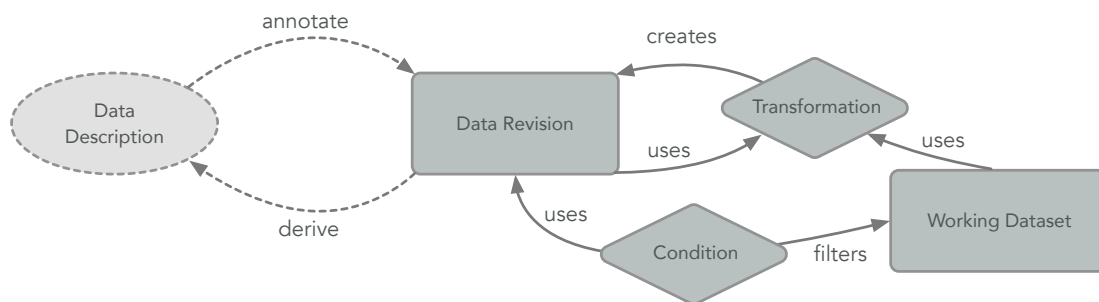


Figure 3.5: Model for storing data provenance from data wrangling. The base data is stored as a data revision (i.e., revision 0). A transformation uses a data revision or a filtered working dataset to create a new data revision. Additional data descriptions are derived from every data revision and are used to annotate it subsequently.

Data Transformations Information on data wrangling transformations is provided as a log, with the ability to undo/redo. The transformations are stored as workflow provenance, showing the actions taken by the user. Utilizing this logging information, we can construct a provenance graph from these transformations. From each operation we derive parameters and affected rows and columns.

Applied Data Filters Data filters are employed to process subsets of the data, this can be done to transform a specific selection. Utilizing this information can give users implications whether the analysis was only conducted on a particular subset of the data. This information is stored as row-level data provenance.

Data Descriptions Interactive profiling of data can be employed during data wrangling to determine data characteristics of the data, e.g., data distributions, anomaly detection. The overall meta-information about the dataset and column characteristics can help to further validate or identify data rows. Descriptive statistical figures of a dataset are often used by data analysts to determine if a dataset is appropriately processed and fit for use. Leveraging these descriptive features for estimating and validating datasets, we can annotate the information extracted from the transformation and filtering operations to make them more meaningful and comprehensible to the user. The data descriptions are stored as row-level or column-level data provenance, depending on the information type.

Visual Analytics Approaches

In the upcoming chapters, I will present various VA approaches specifically designed and developed to address the challenges and white-spots identified in my comprehensive problem statement (see Part I). The goal of these approaches is to develop methods that aid users with their data cleansing and wrangling tasks, as well as provide interactive

means for exploring pre-processing workflows and pipelines. According to the paradigms of using VA and human-centered design in the form of the data-users-tasks design triangle [MA14], I developed these approaches to allow users to explore visualizations and interact with data and annotated information derived from the data. The designs are based on requirements and tasks analyses that preceded prototype development, and the working prototypes are also evaluated in user studies respectively. The evaluation of the VA approaches will be presented in Part III alongside results that will be further condensed and wrapped up in Part IV.

Visual-Interactive Customization of Data Quality Metrics

The design and implementation was published in [BGK⁺18].

When working with data, analysts require some form of probing for assessing the appropriateness of a dataset. For example, a regulatory government institution concerned with monitoring and releasing data on an open data portal needs to quickly assess the quality of the data and ensure its usability. Quality of provided datasets can be highly variable and data providers need to be notified if the quality needs to be improved to maintain the quality standards on the platform. Moreover, datasets may be frequently updated and thus, analysts working at the government institution who are responsible for qualitatively evaluating submitted datasets need to assess them in a timely manner, validate changes in the structure of the dataset, and ultimately evaluate its quality. It is a difficult task to quickly evaluate datasets that are either unknown to the user or to detect changes in quality and structure of frequently updated data. One approach at assessing the quality of a dataset is providing summary visualizations [KPP⁺12] to get a sense of the data distribution and anomalies. Summary visualizations lack flexibility to accentuate different aspects of DQ. I argue that automatically computed DQ metrics can facilitate quality assessment and expedite validation. The conceptualization of DQ metrics in Section 3.1 show means for determining the overall quality of a dataset, as well as for defining, measuring, and managing the quality of information and data [Das13]. In contrast to isolated quality checks (compare [GE18, GAM⁺14], DQ metrics can be used to validate various data characteristics and properties simultaneously. However, the general measures presented (compare 3.1) are often not sufficient for determining quality issues specific to a certain data domain. Context-dependent and intrinsic properties of a dataset, along with domain knowledge, require adaption and customization of employed metrics. To support analysts in effectively adapting data DQ metrics, they need to be able (1) to customize DQ metrics interactively to specific datasets and domains, (2)

validate the appropriateness of the newly customized metrics, and (3) to assess quality easily and quickly.

4.1 Requirements Analysis

Before starting the development and design of MetricDoc, an environment for the visual-interactive customization of DQ metrics, I recapitulated requirements that should be met by our approach. The requirements were derived from (i) literature research and identified shortcomings in other DQ projects (e.g., [KHP⁺11]), (ii) our long lasting experience with visual-interactive DQ projects [GAM⁺14, GGAM12, KPS14a, KPS⁺14b], as well as from (iii) our collaborations with various company partners: in multiple discussions with the target users of such a system, i.e. data analysts dealing with DQ. Human-computer interaction (HCI) experts were actively involved in the design process (see Figure 8.1) giving feedback to requirements regarding visual elements and general user experience.

Moreover, Miksch and Aigner’s [MA14] design principle of data, users, and tasks was pursued: (1) The *users* are DQ analysts with expertise in data profiling and comprehensive knowledge in their respective working domains. (2) The *data* consist of a tabular dataset subject to analysis, with quantitative, qualitative, and time-oriented data supported for analysis. (3) The *tasks* for assessing DQ are split into:

- T1. performing a first assessment of the quality of a dataset (using general DQ metrics),
- T2. adding custom quality checks and customizing DQ metrics to fit the dataset,
- T3. exploring the dataset and inspecting detected dirty entries, and
- T4. reviewing the overall quality of the dataset for a downstream analysis.

To successfully implement an environment that supports those tasks, I defined the following requirements:

R1 Customizable DQ Metrics. DQ metrics should appropriately reflect the quality of the data at hand. To accomplish this, users should be able to adapt DQ metrics to account for domain-specific contingencies or special cases. On the other hand, parameters of predefined ready-to-use metrics should be easily adjustable to ensure flexibility of usage.

R2 Data Quality Overview. A visual overview about a dataset’s quality should be provided. It should specifically convey proportional information on potential errors detected in the dataset.

R3 Error Information. Detailed information about potential dirty data should be communicated to the user down to individual data entries. This information should facilitate the identification of error sources.

R4 Error Distribution. Errors in a dataset rarely occur in an isolated way. Thus, users should be able to view the distribution of errors within the dataset which may reveal patterns. Furthermore, the tool should facilitate the detection of correlations of errors across several data table columns.

R5 Data Exploration. To facilitate the inspection of dirty data, the user should be able to be directed to data table entries with detected quality issues.

Based on these requirements I determine design rationales that should be taken into account during development. The design rationales should ensure that the functional requirements are also reflected in the design. These rationales were adhered to during the design and subsequent development of MetricDoc.

4.2 Design Rationales

The design on MetricDoc should comprise a tabular data representation enhanced by visual elements for presentation and navigation of the dataset based on DQ information. Interactive feedback should support the user during quality metric customization and provide immediate computation results. Usually, these users – data analysts, data scientists, or statisticians concerned with DQ and pre-processing – rely on scripting and textual interfaces for profiling data and developing DQ metrics, hence they cannot easily explore the raw data based on the results of the computed metrics. According to the design methodology by Sedlmair et al. [SMM12], particular data abstractions, visual encodings, and interaction techniques are required to develop effective visualizations. This methodology was applied to our quality metric and error distribution data with an emphasis on visual presentation and exploration. Our design was influenced by current wrangling, profiling, and cleansing approaches [GAM⁺14, KPHH11, KPP⁺12], as well as tabular-like overview visualization techniques [RC94, SFTM⁺13] with orientation towards interactive exploration [Kei02]. Accordingly, the following design rationales were distilled based on the requirements defined in Section 4.1.

D1 Providing Consistent, Informative Visual Encodings [R2–4]. Due to the potentially large scale of the data, the analysts need to detect data problems efficiently. Therefore, the visual encodings of quality and error information should be consistent throughout the environment to avoid misinterpretation and to recognize certain information that is – albeit in different granularities – displayed repeatedly. Alternatively, a number of specific representations for different data types and quality dimensions could be employed. Utilizing only basic visualization types keeps the learning threshold for users low. Especially for large scale datasets, data aggregations are common means for efficient visual representation. On the other hand, such aggregations could potentially mask quality problems in the data, and are thus, not applicable for the task of data profiling. For this reason, simple but intuitive elements are employed to show error information, to support the user’s understanding, and to lower the barrier of entry for inexperienced users.

D2 Employing Multiple Linked Data Perspectives and Views [R2–4]. Users’ data analysis workflows and tasks may differ considerably, requiring access to different data aspects and visual representations, including DQ information. DQ analysts often resort to raw data representations or statistical overviews of datasets, switching constantly

between different representations. Showing exclusively detected errors without providing context, prevents users from determining possible causes of errors. A comprehensive overview requires knowledge of the errors persisting in the data, which is often not feasible. Thus, our environment should provide an overview of the dataset and its quality, while simultaneously maintaining detail information about the dirtiness in the data. Supporting brushing and linking [Mun14] across visualizations and data views facilitates conducting the quality assessment tasks. Leveraging effective exploration techniques on different granularity levels is supposed to allow quickly identifying quality issues throughout the dataset, by inferring location information and contextual information on surrounding data.

D3 Interactively Supporting Quality Metric Customization [R1]. DQ metrics are potentially complex measures (see Sections 2.1.1 and 3.1) and require domain-specific adaptations [Das13]. Developing and tailoring quality checks to extend the effectiveness of a quality metric in detecting dirty data, and to contextualize domain characteristics, respectively, is important. Iteratively building and customizing metrics is difficult without constant feedback on syntactical and semantic changes on calculations. If no feedback is provided during metrics development, users have to resort to external tools for determining the appropriateness of the current metric, which disrupts the development process. Supporting interactive customization also implies increased computation effort, which could impede interactivity of the entire environment. However, immediate feedback allows the user to verify if changes resulted in a more adequate domain mapping or improved error detection of the metric. Such feedback should be provided through notifications and the exploration environment accordingly. The aim is to encourage analysts to continuously refine the DQ metrics and model the data domain most adequately to identify quality issues and reduce the classification of false positives.

D4 Guiding Users during Data Exploration [R5]. DQ metrics evaluate the quality with respect to specific characteristics or aspects of the data. The user should be informed of such aspects when exploring data, and be able to comprehend the evaluation schemata of metrics, especially if they are complex. However, varying types of users follow different workflows when exploring dirty data, assessing DQ, and developing DQ metrics. By offering a workflow to be adhered to throughout analysis, expert users are likely to be put off by feeling too constrained. Without any visual assistance, on the other hand, novice users are likely to be lost in a complex exploration environment. Thus, used visual encodings should quickly communicate where investigation is required, e.g., highlighting problematic data entries. Users should also be notified of changes in quality – as a result of metric recalculation or changes to the original data. Visual cues are used to point the analyst to DQ problems, while the absence of such visual cues signifies high DQ and no need for intervention.

With these design rationales defined I proceeded with prototyping the MetricDoc environment. Chapter 8 describes how visual encodings and interactions were subject to change during iteration cycles, with the core elements left widely unchanged.

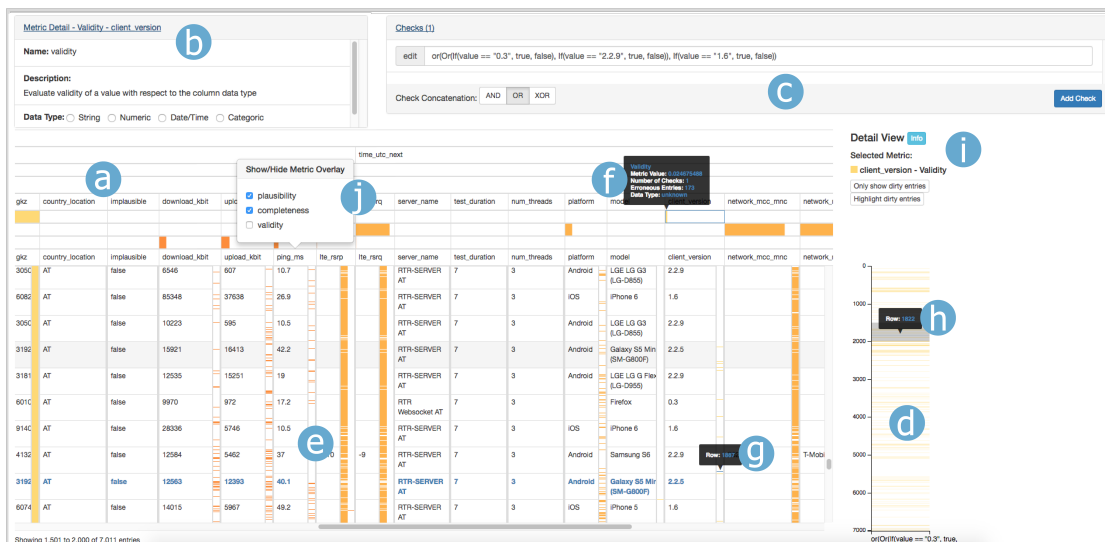


Figure 4.1: MetricDoc: An interactive visual exploration environment for creating and customizing DQ metrics and assessing DQ (this is a composed view, which shows multiple popups and tooltips at the same time). The environment consists of the Quality Metrics Overview (a), the metric information view (b) and customization tabs (c), the Metric Detail View (d), and the tabular Raw Data View enhanced with Error Distribution Heatmaps (e). Mouseover tooltips provide detail information on metrics (f) and data errors (g,h), Metric Distribution Heatmaps can be enabled and disabled individually (j). Case Study (see Chapter 7.1) Task (1): Entries are highlighted that show test devices performed with outdated client versions (row 1892). The labels (a-k) are used in subsequent figures to retain reference to the rest of the environment.

4.3 Visualization Design

MetricDoc's web user interface provides a visual exploration environment that features both a raw dataset representation and an overview of DQ metrics along with a representation of the distribution of dirty data entries within the dataset (see Figure 4.1). Users can manage the deployment of DQ metrics and corresponding quality checks on datasets. The interface emphasizes on visual support for dirty data exploration as well as visual feedback during metric customization. In the following section I elaborate the employed visual encodings to ensure that easily comprehensible exploration and metric customization is provided to support DQ assessment.

4.3.1 Quality Metrics Overview

The metrics overview (see Figure 4.2) is one of the main components in MetricDoc. For single source metrics, the representation resembles a tabular structure, column by column indicating a DQ summary, while rows in this table correspond to different DQ metrics. The tabular representation aims at inducing a relation to columns in the original data

4. VISUAL-INTERACTIVE CUSTOMIZATION OF DATA QUALITY METRICS

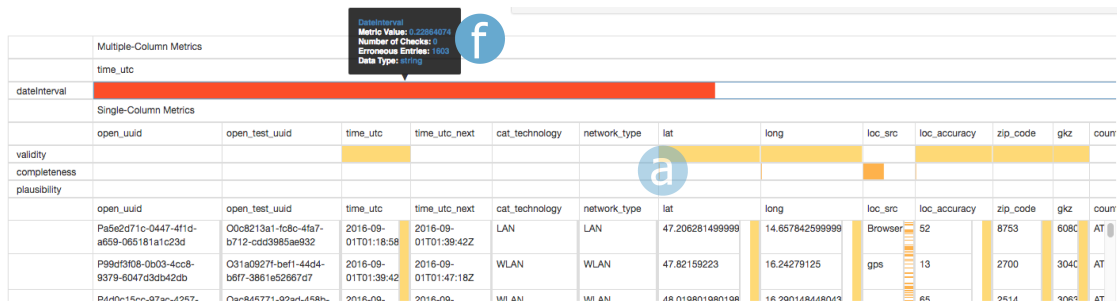


Figure 4.2: Quality Metrics Overview (a), including metric detail within a mouseover tooltip (f). Single-Column and Multiple-Column Metrics are visually separated to emphasize information disparity (compare Fig. 4.1a,f).

table by aligning the *metrics overview* with the tabular representation of the original data which is positioned directly below. For each metric and each column the amount of identified dirty entries is indicated by an error bar. Spanning metrics correspond to multiple source columns and implicit information cannot be deduced from one singular column. Hence, for these metrics positional alignment with the Raw Data View is omitted and instead the columns to indicate which are evaluated by means of such a spanning metric are labeled. The width of error bars representing spanning metrics is accordingly spanning the whole data table width, to distinguish them from normal metrics. With these features the design satisfies R2 and keep consistent with D1, by informing the user about the general dirtiness of a dataset and providing an overview of any available DQ metrics.

The error bar indicates the ratio of dirty entries discovered for the computed quality metric by proportion to the entire metric cell width, orienting the user towards columns lacking quality. Hence, an empty bar represents the absence of dirty data and implies cleanness. The overview can be sorted by dirtiness per data column, combined for all metrics, to guide users to columns that require inspection. Tooltips give on-demand information (see R3) about the absolute amount of dirty entries, the actual error percentage, and other metric details (e.g., Figure 4.1f). Upon selecting one or multiple DQ metrics the Metric Detail View shows information for further inspection.

4.3.2 Metric Detail View

To represent detailed quality metric information, the prototype features a schematic error view (see Figure 4.3) that shows error information for all entries in a dataset. The result is a heatmap visualization showing the distribution of the errors in the dataset, a representation adapted from distribution column overview heatmaps by Sopan et al. [SFTM⁺13]. For large datasets that exceed pixel-wise entry representation, data are aggregated with color intensity corresponding to the number of errors in the aggregated data rows of the heatmap. Each quality check contained in a metric corresponds to one vertical column in the detail view. As such, the view can be used for error type

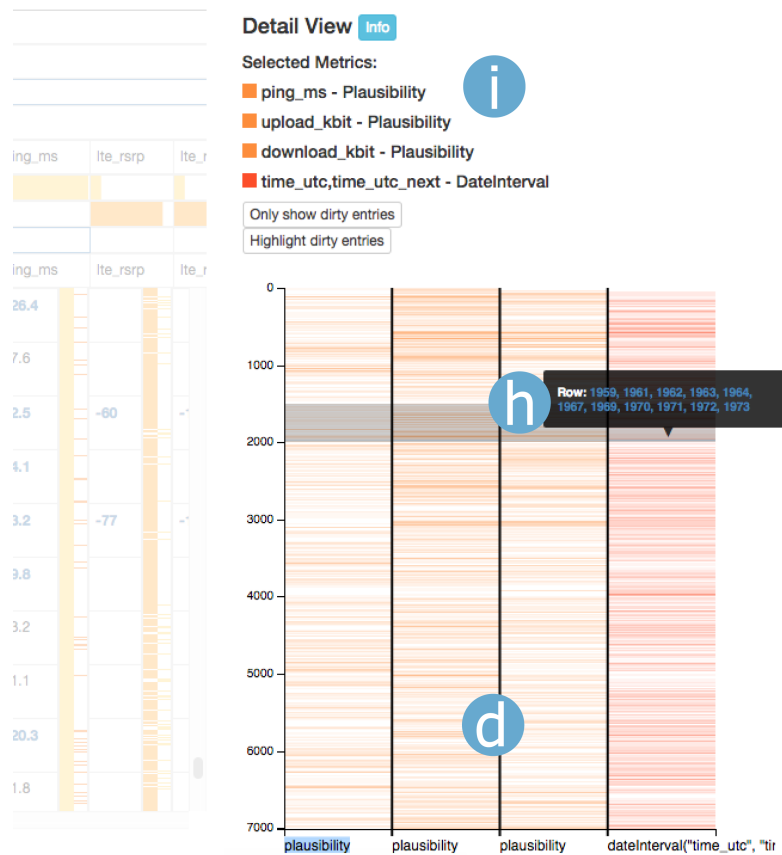


Figure 4.3: Metric Detail View (d) showing the error distribution throughout the dataset of both the *completeness* metric of column *long* and the *date interval* metric for columns *time_utc*, *time_utc_next* as can be seen in the legend (i). Users can toggle showing only dirty entries to facilitate comparison of such entries, or highlight dirty entries to see them within the context of the entire dataset. The mouse is hovering over the visualization, giving tooltip information about erroneous rows (h). Users can interact with this view to interactively browse to regions of interest in the raw data table. This allows for detailed inspection and swift exploration. By enabling selection of multiple metrics at once, error correlations (like in this example the *plausibility* metrics of columns *upload_kbit* and *download_kbit*) can be inspected and analyzed (compare Figure 4.1d,h). Case Study (see Chapter 7.1) Task (3): The Metric Detail View shows test entries being called within a 10 seconds time frame. The upload and download plausibility metrics show a large number of outliers, implying that there are excessively low and high down- and upload rates throughout the dataset. Some of which could be connected to a short time between tests performed (column four *dateInterval*).

exploration (by checking the errors for different checks individually, satisfying R3) and navigation (satisfying R5). The analyst can determine patterns and increased error occurrences directly from the view or – if necessary – adapt quality checks with respect to the detected inconsistencies, depending on the situation that false positives or true negatives are detected to improve error detection accuracy or comprehensiveness.

Annotations give analysts additional feedback about the location of an erroneous value in the dataset. When multiple metrics are selected in the Quality Metrics Overview, the view shows all metrics simultaneously. This allows for error reconciliation and more sophisticated analysis, especially for errors that manifest in several aspects of the data or different information channels (also across other columns). The analyst can quickly jump to the row of detected dirty entries and inspect them in the raw data table, having contextual information from neighboring columns and entries (D2). The view can be toggled to either display all entries in the dataset, with optional highlighting, or only dirty entries with respect to the currently selected metric/s, hence contextual dependencies among erroneous data can be observed more easily. This is emphasized by color-coding disabled rows in the view. The Metric Detail View is linked to the raw data and Error Distribution Overview and infers the current position in the dataset, users can interactively browse into subsets of the data.

4.3.3 Error Distribution Overview (Figure 4.1e)

In addition to the heatmap-like overview of error distributions given in the Metric Detail View (see Figure 4.3d), the raw data table features heatmap-like elements to meet D1 (see Figure 4.1e). Each column of the raw data table is annotated with a scrollbar-like visualization, representing the relative position of dirty entries of the respective column. For large datasets the table representation is paginated to facilitate navigation and thus, the Error Distribution Overview is showing only errors for the selected table page. In combination with the error distribution in the Metric Detail View, the analyst has at his/her disposal a twofold exploration system for either quick navigation of the overall dataset or detailed inspection of the raw data. With the error distributions for all single column metrics being juxtaposed, analysts can leverage their perceptive ability to discover error patterns that spread across columns. Interactions are consistent across the Metric Detail View and the Error Distribution Overview, featuring mouse-over tooltips on error position and selection highlighting of raw data entries.

With the two ways available to navigate the dataset and detected errors, analysts are able to explore and validate the data based on their preferences (either scrolling through raw data entries or utilizing the detail view for jumping to points of interest). DQ analysts could find the multitude of juxtaposed scroll elements distracting, hence the Error Distribution Overview can be disabled for each metric and column individually. In addition, only displaying the metrics that are currently of interest to analysis allows putting the analytical focus on particular data columns. While initially only few DQ metrics might be added to the dataset, this error distribution provides additional overview information and hence directs the analyst towards adding new DQ metrics that fill the

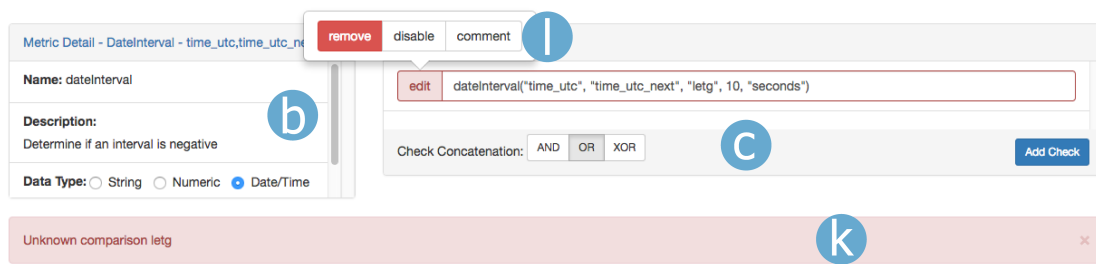


Figure 4.4: Metric information (b) and customization tab (c), with edit feedback notification (j). In this case the third metric parameter is misspelled, the user is informed by an alert. After editing a metric, a notification informs the user of changes in the amount of detected errors. Quality checks can be disabled or removed entirely. Comments can be added to checks to give contextual information (k). Concatenating checks gives additional flexibility for the validation of data entries (compare Fig. 4.1b,c).

blank-spots the analyst discovered while skimming the data. The analyst can select a column header to directly create a metric for the respective column, allowing a more streamlined user experience and aiding the analyst in dataset orientation (D2). The overview can be sorted by the amount of dirtiness detected per column for all metrics, if necessary/desired.

4.3.4 Metric Customization

Based on R1 DQ metrics not only need to detect default errors specified by the data analyst, but should also be customizable to account for domain-specific data constraints. Therefore, our environment provides means for adding or customizing quality checks in order to evaluate different domain-specific constraints and dependencies, increasing a metric's effectiveness and expressiveness for detecting errors in a dataset. A Quality Check panel is provided (as can be seen in Chapter 4.1) that lets users edit DQ metrics, and gives additional information about the metric type. In the *checks tab* (compare Figure 4.1), quality checks can be scripted in OpenRefine's GREL scripting language, which provides the freedom to perform calculations and check if an entry satisfies or violates an arbitrary condition. These scripts are dynamically evaluated for syntactical and semantic validity (e.g., invalid function parameter) on the server side and users are dynamically notified. Textual input offers enough flexibility for users. For further information, all available custom metrics, quality checks, and helper functions can be accessed in a popup view, giving information on functionality, parameter usage, and default configuration.

Furthermore, the Metric Customization panel allows disabling or deleting checks as well as creating new checks. Changing a metric or quality check causes a re-validation of the DQ which is immediately reflected in the metric visualizations (*metric overview*, *Metric Detail View*, and *Error Distribution Overview*). Moreover, the user is notified (see Figure 4.1k) about changes to error count and overall quality. Adding new metrics prompts a creation form, that gives quick information about the data type distribution

for selected columns, and which metrics can be created – depending on which metrics are already being evaluated. The data type overview provides details about the column’s type distribution to let users assess which metric is appropriate.

Disabling or enabling specific metrics or checks lets DQ analysts build up a backlog of quality checks that can be enabled for quick validation. This potentially boosts productivity, sophisticated checks do not have to be rebuilt from scratch but can be reused and adapted to domain-specific circumstances. With support for multiple data projects, users can more quickly assess quality and validate new projects and furthermore discover errors in the data by reusing (custom) metrics from previous projects.

Capturing Provenance from Data Wrangling

This design and implementation was published in [BGM19].

When analyzing data in any way, the initial step before actual analysis is preparing data and ensuring that it is of adequate quality. Data quality management has developed to be an integral part of almost any data processing workflow, to increase the reliability of analysis results.

Data wrangling is often employed in order to improve the quality of a dataset. However, the outcome should still be representative of the original dataset. The steps taken pre-processing a dataset in order for it to be usable are often not documented, and hence are seldom reproducible. When using large datasets and obtaining data from different data sources, it is increasingly difficult to perform quality inspection on the raw data. Transformation histories automatically generated by data wrangling tools are often not available outside of the system, and thus, the history of data transformations can not be audited when importing the data for subsequent data analysis (usually different tools are used for data pre-processing and data analysis). Also, there is a lack of context if these wrangling operations led to the desired outcome so that issues were actually resolved.

Data provenance is captured to allow retracing how it was created, from what data it was derived, and how it was changed. This allows to retain sources of errors and allow re-tracing of previously applied operations. Especially across multiple systems provenance can allow tracing back to sources of changes. Simmhan et al. [SPG05] described a graph structure to be adequate for storing data provenance, however provenance is mostly captured in scientific workflow applications, and rarely logged in data quality management. Storing the data states (also called as revisions) in the graph's nodes and the transformation processes in it's edges gives an explorable overview of the provenance structure. The inherent a-cyclical structure shows the data lineage and allows the

identification of process sequences. When capturing provenance during data wrangling, actions can be annotated with contextual information, to give more semantic meaning to the wrangling operations and their impact on the data. So far, existing data wrangling tools and solutions have not embraced data provenance proficiently enough to have analysts benefit from their wrangling attempts. Context information is used in data profiling to recommend data transformations (e.g., Wrangler [KPHH11], Trifacta Inc., etc.). Interactive methods for data profiling are often employed to analyze certain characteristics and dimensions of the data, like specific columns of interest, or particular data types, which can be leveraged to facilitate data wrangling. The MetricDoc VA solution showed that annotating a dataset with data quality metrics allows for detecting quality issues and serve as a measure of overall quality for a dataset (see Chapter 4). The interactive approach allows exploration of metrics to assess the prevalence of certain types of errors in the data and estimate the quality of a dataset in detail. I propose that leveraging data quality metrics as data provenance can aid the user in understanding the development of the dataset’s qualitative conditions. This builds confidence in the reliability of a dataset.

By providing an approach for exploring data and workflow provenance captured from data wrangling steps, I will illustrate how users are able to build trust in a wrangled dataset. By logging what actions were used alongside the wrangling process (e.g., data profiling, filtering, cleansing), it is possible to gain understanding of sequences of transformations, and make sense of the entire process. Computing quality metrics continuously for each state of the dataset is supposed to give users the ability to quickly assess the qualitative condition of the data and determine if quality has changed throughout the wrangling process. I found that current approaches for exploring provenance are lacking the ability to annotate the data sufficiently to give contextual insights into the data wrangling process. Furthermore, interactive exploration of preceding alternative branches is not possible.

5.1 Provenance Model Implementation

In Section 3.3 I conceptualized a model for generating provenance from data wrangling and annotate it with DQ information. The DQ metrics utilized in this model are based on the MetricDoc approach. This approach annotated data with data quality metrics to provide means for visually exploring the quality of tabular datasets. To implement the enhanced provenance model, these metrics are used to save column- and row-level data provenance to capture contextual information, allowing analysts to analyze the development of quality over time.

Revisiting the definition from Section 3.1, a DQ metric is defined as *“the quantified measure of a data quality dimension that gives proportional information about the lack of quality regarding a certain information aspect”*. For each employed metric, the dirtiness of one or multiple columns is measured with respect to a certain quality dimension. The overall measure is the inverted ratio between determined dirty tuples and the number of

rows in the dataset. This yields a normalized measure between 0 and 1 for each metric, which can also be interpreted as the percentage of dirty tuples detected by the respective metric. The evaluation of a tuple is done through a validation function $vf_m(\cdot)$, returning a Boolean measure of dirtiness. However, the engine also retains information on the position of the dirty tuple within the dataset so they can be located. This information will be used as a data descriptor to annotate the data provenance.

The list of available DQ metrics stored as provenance are obtained from the MetricDoc engine. The provenance model data structure is implemented as an extension to the open source wrangling tool OpenRefine, to support the DQ framework [BGK⁺18]. The two integral extensions of the existing data quality framework are the data quality engine and the provenance model (compare Figure 5.1). The data quality engine automatically recommends DQ metrics based on column type. To accomplish this the engine features a heuristic validation schema that determines the predominant data type for each column. Custom quality checks can be added in the separately available MetricDoc environment [BGK⁺18] to detect domain-specific issues and hence improve the accuracy of the issue detection. The provenance model implementation is integrated into MetricDoc's web front-end and allows the definition of custom quality checks that validate the data with respect to domain-specific characteristics, if necessary (e.g., numeric constraints, text validation). The second feature extending the OpenRefine tool is the addition of the provenance annotation model. DQ metrics are automatically computed and annotated for every data revision and are stored in a provenance graph structure that extends the default data storage. Based on the data quality metric structure, the annotated information stored as data provenance ranges from the overall dirtiness of a particular column and metric, down to the individual indices of dirty tuples. The metrics calculation and provenance annotation is automatically computed on server-internal engines, which reduces the impact of performance during wrangling to a minimum. For a typical wrangling scenario with multiple wrangling branches, the data structure size can be fetched via `http`-access. Since the provenance model extends the default data structure, additional data storage is minimal and only concerns workflow provenance and column- and row-wise data provenance.

5.2 Requirements Analysis

In the Related Work Section 2.5.1 I gave an overview of VA research in provenance generation and data quality management, and motivated the opportunities for combining these fields. It can be seen that trends have developed towards interactively inspecting data quality based on quality metrics [BGK⁺18], facilitating data wrangling through recommending data transformations [KPHH11] and making the effects of such transformations easily comprehensible [KPP⁺12]. To determine the tasks that should be supported by a system that combines provenance and data quality analysis, I performed a requirements analysis of different taxonomies and research directions: Kandel et al. [KHP⁺11] motivated the development of means to (1) diagnosing data problems, (2) editing and auditing transformations, (3) using provenance to track data lineage, and

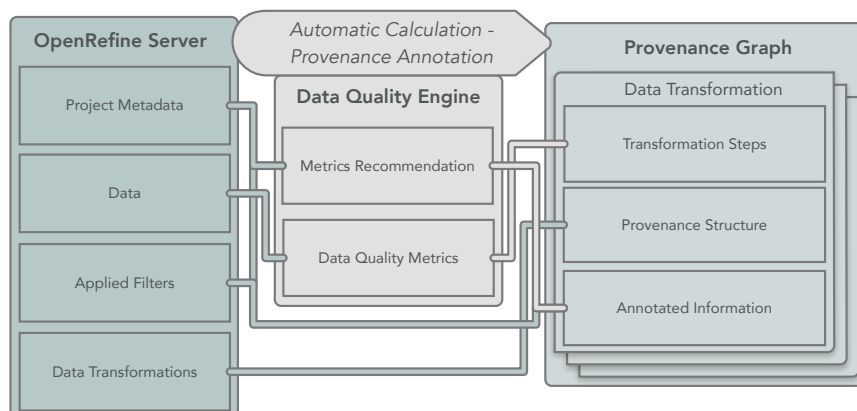


Figure 5.1: Overview of our extension for capturing provenance from OpenRefine. Information is propagated from the server to the data quality engine and provenance model. For every data state and wrangling step, the extensions process information from the project to store it as provenance. The result is an annotated provenance graph that can be used for analyzing the outcome of the wrangling process.

(4) understanding why actions were performed. I elaborated these means further towards the purpose for analyzing provenance, according to Ragan et al. [RESC16].

5.3 Task Considerations

I deem the following task considerations to be important to effectively support users with analyzing provenance from data wrangling. Prior to defining the tasks, I applied the Data-Users-Tasks design triangle by Miksch and Aigner [MA14] to first determine the users of our approach – data analysts, software developers, and domain experts concerned with data management –, and the data used – provenance captured during the data wrangling process. The following tasks can be distinguished according to Ragan et al.’s [RESC16] characterization of provenance purpose within the scope of assessing data quality. I analyzed low-level events and actions within the context data wrangling and cleansing domain to derive high-level actions that are pursued by users operating applications with data quality context in mind.

T_{act} Action Recovery. The analyst wants to see the transformation sequences applied to a dataset and the quality issues retained throughout the process at the level of individual columns. This includes the types of operations, their parameter settings, and the subset of data the operations were applied on.

T_{pres} Presentation. If multiple alternative operation sequences have been created, the analyst wants to visually inspect the differences between different wrangling branches. This includes information if an operation impacted the dataset, what part of the dataset

(column- or row-wise changes), and more particularly, if quality was affected. Furthermore, the analyst wants to inspect if subsets of the data exhibit more issues than others (e.g., the sensors of a weather station introduced more measurement artifacts than all others).

T_{meta} Meta-Analysis. When inspecting a sequence of operations, the analyst wants to audit the dataset if it can be trusted for further processing or analysis. To do this, the analyst monitors the development of different quality problems over time to eventually decide on the usability of a dataset. Also, the analyst wants to reconcile what operations the different branches have in common. The analyst wants to use these insights to determine how issues in the dataset were addressed and decide what operations solved these issues most appropriately for downstream analysis.

T_{rec} Recall. The analyst wants to compare the remaining issues in the dataset for two branches (at a time) in order to determine if error patterns were addressed in a similar way, or if different wrangling approaches were employed. By investigating the quality metrics of the dataset over the course of multiple operations, the analyst wants to identify if changes had qualitative impact and trace changes in quality back to the operations that caused them. This includes validation, if either the entire dataset or a particular subset of the data (that has been selected for further analysis) exhibits sufficient quality (e.g., auditing the columns of a dataset).

T_{rep} Replication. The analyst wants to be able to revert the current dataset to previous transformation steps, to either use the dataset for downstream analysis, or as a starting point for further data wrangling. If problems persist in a particular state, the analyst wants to inspect them in detail.

T_{coll} Collaborative Communication. The analyst wants to inspect a sequence of previously applied operations and, in particular, their consequences in terms of quality.

5.4 Usability Inspection Study

Various approaches can be employed for data wrangling, depending on the methods for exploration or evaluation. Individual analysts can have vastly different demands on the quality of a dataset. I conducted a usability inspection study to receive feedback on different design alternatives of an early paper design of our prototype (see Appendix 13). The test subjects were all undergraduate computer science students, with basic knowledge of information visualization. The reason these participants were selected is that they are similarly trained in methodologically approaching data analysis as our target user group. They were split up into two groups, where the first group (four participants) was interviewed individually on the designs, and the other group (six participants) conducted a focus group usability inspection. The collected positive and negative feedback served as a basis to determine the important design requirements and refine them for the final prototypical implementation.

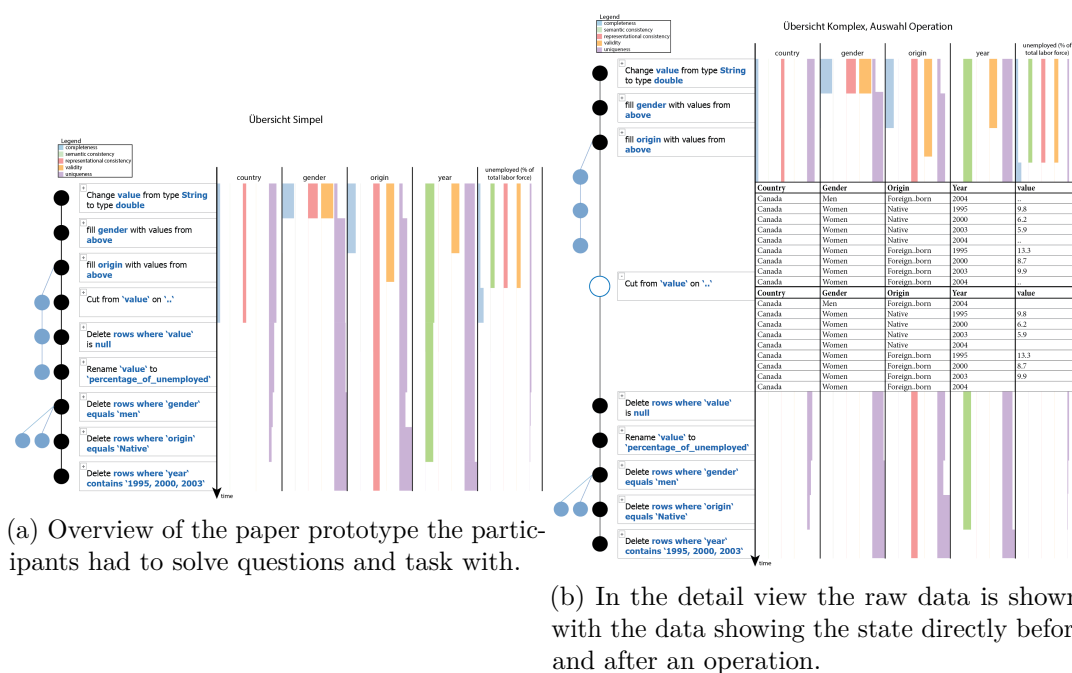


Figure 5.2: Initial design prototypes of the quality flow view used during the usability inspection evaluation.

The evaluation was split into two study groups, interviews and a focus group. Both groups received an introduction to the subject. The participants received paper prototypes and had to solve questions and tasks, as well as give feedback on the usability in the end. These questions and tasks involved validation if the design was intuitively comprehensible, and finding improvements to the design. During evaluation, the investigator continuously guided the participants through the experiment, asking questions for different tasks and consecutive operations. Figures 5.2a and 5.2b show the design of some paper prototypes used during the experiment, and participants could use pencils to add notes.

Feedback showed that people preferred and appreciated that (1) less colors indicated that the dataset is cleaner, and that operations were shown prominently and pleasantly. (2) The similarity to `git`'s commit graph and alternate graph branches were noted positively. Participants also noted (3) highlighting of the operations and a visual representation of structural changes in the dataset as beneficial. Lastly, participants commented that (4) the overview of changes and the well understandable structure can guide users to follow the effect of operations for most parts. Hence, I tried to integrate this feedback into new iterations of the prototypical design.

5.5 Design Rationales

In order to support analysts' different approaches, the design is supposed to allow users to navigate through the available quality information from different perspectives and enable users to initiate exploration by investigating details, but also pursue a classical overview-first analysis. In both cases, users should be able to progress towards their specific goal of assessing the quality of the targeted data state, processing branch, or data column or row. Finally, design rationales were established to ensure that they support users in performing the tasks presented in Section 5.3.

- R₁ Allow analysts to navigate through the available quality information from different perspectives.** Enable exploration by investigating details, but also by pursuing a classical overview-first approach. Analysts should be able to navigate towards their specific goals (analyzing a specific branch, data revision or column/row).
- R₂ The design should emphasize the impact of operations on quality.** This helps users to associate transformation steps with changes of quality.
- R₃ Cleanness of the dataset should be signaled by cleanness of the visualization** This should emphasize the analyst's perception that no problems can be observed any more.
- R₄ A graph of operations should show the different wrangling branches.** The branch of the currently selected wrangling operation sequence should be traceable.
- R₅ The overall size of the dataset and number of quality problems for every data revision should be communicated.** This should help analysts identify what parts of the data are changed during a transformation.
- R₆ The detail view should give additional information on operations and changes in quality.** This should help to provide insights into how and why an operation influenced the dataset.

During development, the task considerations and design rationales were consulted to prevent inappropriate design or functionality. The upcoming section describes the core features of our prototype, where single or multiple tasks identified were used as design goals for individual components. At the initial design stage, applicability of the design rationales to the components were determined. Throughout design and implementation, the components continuously underwent inspection if design rationales were supported and maintained.

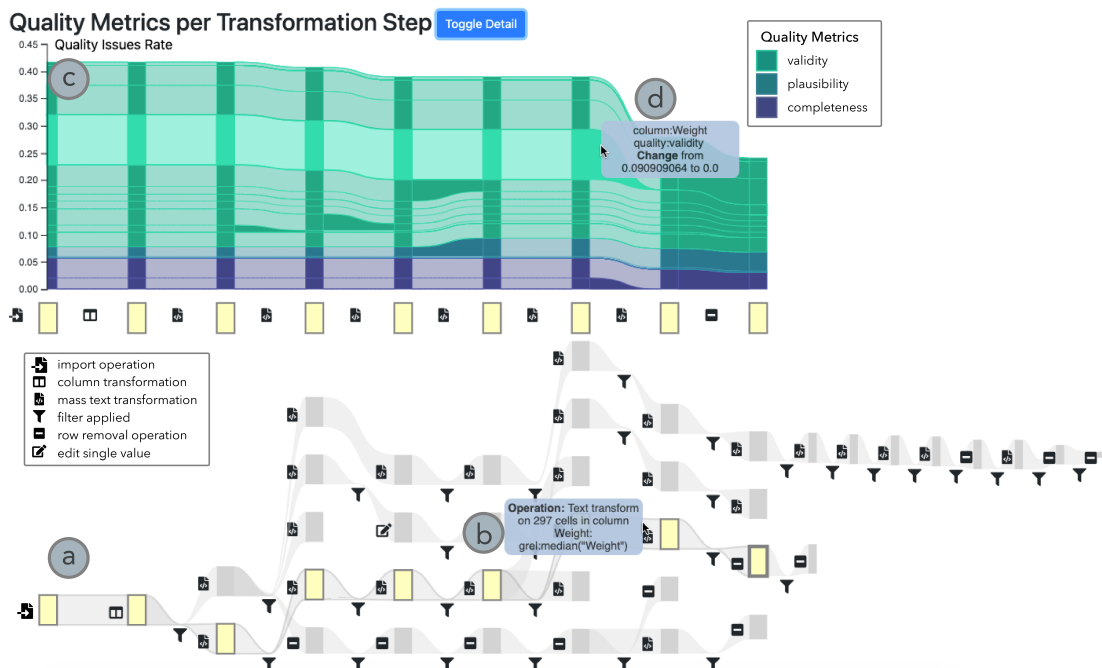




Figure 5.3: Two linked views of DQProv Explorer: (a) The Provenance graph view allows navigation of the individual data states. The height of the nodes and edges encodes the row size of the data (R_5). Bright yellow graph nodes indicate the currently selected branch, icons indicate the type of operation. (b) On-demand mouseover information on the nodes and vertices shows details on the operations and the dataset size: vertices show information on the employed filtering and transformation parameters, nodes show the number of rows in the dataset. (c) In the Quality Flow View users can observe the development over time for a selected wrangling branch. The bar height indicates the proportional amount of issues detected, color encodes different types of quality metrics. On the horizontal axis, the data revision nodes are duplicated from the selected graph branch to align with the stacked bars to facilitate relating operations to changes in quality. (d) On-demand information on the Quality Flow View highlights the flow of the currently inspected metric (*validity* metric in the *Weight* column) and shows additional provenance information.

5.6 Visualization Design

I present Data Quality Provenance Explorer (DQProv Explorer), a VA approach to visualizing provenance that was captured by our data wrangling provenance model. I employed Shneiderman’s visual information seeking mantra by giving overview of wrangling provenance in a provenance graph view as well as details on quality in a flow-like visualization. I provide three interactively linked components in our system, the Provenance Graph View (see Fig. 5.3a), the Quality Flow View (see Fig. 5.3b), and the Issue Distribution View (see Fig. 5.4), usually located to the right side of the Quality Flow View). Provenance Graph View consists of a graph of provenance generated by wrangling the current dataset. The Quality Flow View shows a selected wrangling branch in detail, the Issue Distribution View shows how quality issues are distributed across the dataset for the currently selected revision.

5.6.1 Provenance Graph View

The Provenance Graph View serves as the central (overview first, \mathbf{R}_1) navigation element of DQProv Explorer (see Figure 5.3a), showing the captured wrangling provenance (\mathbf{R}_4). Inspired by Wu et al.’s [WYM12] uncertainty flow visualization approach, it shows an acyclic graph flow structure, representing transformation operations and data flow between data states. Upon selection of a graph node (i.e., a revision state), the path of transformations is highlighted as a bright yellow path (compare Figure 5.3a), and the Quality Flow View is aligned respectively (see Figure 5.3b). The node heights encode the relative number of rows (compared to the maximum number of rows) in the current data state (\mathbf{R}_5). Icons show the operation types for each revision node (e.g.  indicates a text transformation), and filter icons () along the graph vertices indicate if the dataset was filtered before applying an operation. This overview lets analysts assess which operations were applied at a glance. On demand, detailed information on the applied operations and filters is available (compare Figure 5.3b, \mathbf{R}_6).

The Provenance Graph View can be used to analyze different aspects of the wrangling provenance model. By following the flow of data along the graph’s vertices and the node height, it is possible to see if operations were only applied to subsets of the dataset. Together with the Quality Flow View, branching and branch lengths in the provenance graph shows analysts the history of previous wrangling attempts: short paths or a large number of branches could imply unsuccessful wrangling attempts; Long paths with the same operation icons can indicate small, repetitious operations without significantly changing the dataset (e.g., editing single cells) or impacting quality.

5.6.2 Quality Flow View

DQProv Explorer’s Quality Flow View (see Figure 5.3c) shows the overall development of quality issues in a dataset over the course of a selected wrangling branch (\mathbf{R}_1). The view shows the proportional amount of errors identified in the dataset by stacking bars for each employed data quality metric (for applicable columns) in the data quality engine.

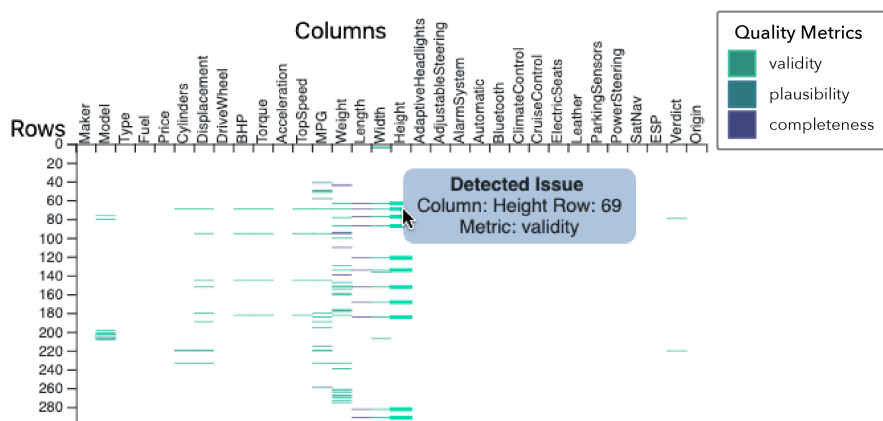


Figure 5.4: The Issue Distribution View allows the inspection of issue patterns detected in the current data state. In this particular case it can be observed that (among others) row 69 exhibits multiple errors. The view is linked to the Quality Flow View and mouseover interaction highlights the respective metric flow (compare Figure 5.3d).

This results in a vertical column of quality issues for each data revision. Different colors indicate different types of quality metrics, and correspondingly different types of issues. By showing the development of these quality issues along a selected provenance branch and the corresponding operations, the analyst can assess which wrangling operation changed the dataset and resolved data quality issues (\mathbf{R}_3). The stacked bars are connected with a flow-like encoding. The flows are de-saturated for metrics that remain unchanged between revisions and are saturated to highlight a change of a quality metric measure between two revisions. If all issues detected by a particular quality metric are resolved during a wrangling operation, the corresponding flow bundles to zero (compare Figure 5.3c: the metric value changes to zero, indicating that the detected *validity* issues of the column *Weight* have been resolved by this operation). Because the Quality Flow View is aligned with the Provenance Graph View, changes in quality can be traced back to the performed transformation operations and the analyst can gain insights if wrangling operations influenced quality (\mathbf{R}_2).

Mouseover interaction highlights the entire quality metric’s history in the current branch, giving information on the quality metric values (\mathbf{R}_6). Figure 5.3 shows an example where the initial actions did not affect quality. However, after the fourth operation, the number of quality issues continuously decreases. Inspecting the saturated flows with mouseover interactions shows the name of the affected column and metric type (compare Figure 5.3d).

5.6.3 Issue Distribution View

The third component in the DQProv Explorer is the Issue Distribution View, which can be used for detailed inspection of the distribution of quality issues within the dataset. It shows the relative location of dirty rows within the tabular structure of the dataset.

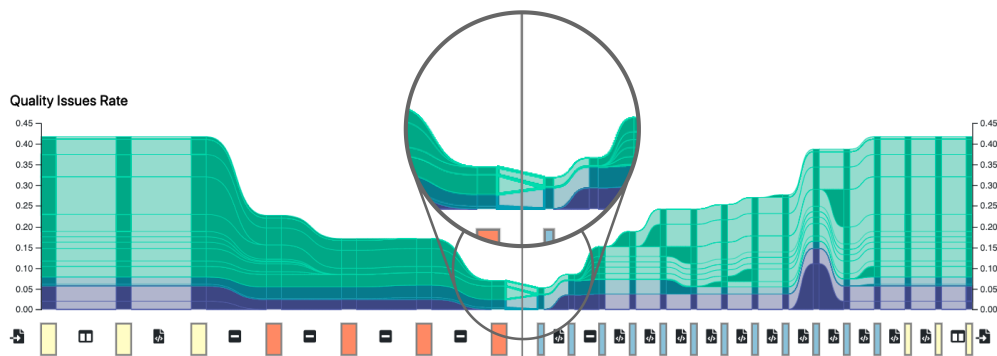


Figure 5.5: The comparison mode juxtaposes two wrangling branches. The first branch runs from left to right, while the second branch is flipped to run from right to left. This allows for direct matching of the end-states of the dataset of the selected branches. In particular, differences in quality are more easily identifiable. In this image the first three nodes are shared between both branches. It can be observed that different approaches to improve quality have been employed, while in the left branch data elements were removed (multiple -operations), in the right branch elements were edited or imputed (consecutive -operations). However, both approaches led to a reduction of quality problems (height of bars).

Erroneous entries in the dataset are shown as heat bands, with color encoding the issue type (corresponding to the quality metric identifying the issue). This visualization implies the cleanliness of the dataset, with a close to empty view signaling the absence of quality issues (**R₃**). Inspecting the Issue Distribution View helps discovering error patterns in the dataset (compare Figure 7.3, Analysis Step 1). It is an extension of the *schematic error view* presented in the MetricDoc environment [BGK⁺18]. If the number of rows in the dataset exceeds the number of rows available in the visualization, the rows are aggregated to accommodate for insufficient screen space, accumulated errors correspond to higher saturation. That way, it is possible to display datasets exceeding 10,000 entries. When entering into Comparison Mode (see Section 5.6.4), the difference of the two issue is computed, which allows inspecting the differences in error distribution between these two revisions.

5.6.4 Comparison Mode

To enable the comparison of the overall quality between two different wrangling branches of the provenance graph, the Quality Flow View design was extended to oppose two wrangling branches (see Figure 5.5). The view is displayed when two branches are selected, mirroring the Quality Flow Views, allowing a direct quality comparison of the branches' end points. This view lets analysts compare the flow of quality over time, but also inspect the difference of employed wrangling operations. For example, if an analyst has to decide to continue analyzing the data, and two wrangling attempts (branches) look similar, it is possible to use the comparison mode to assess which sequence of operations yielded

better quality, or used less wrangling steps but was equally as effective. To retain linking to the transformation operations applied to the selected branches (**R₂**), the branches are duplicated and positioned below the Quality Flow View. The selected branches in the Provenance Graph View are highlighted in clearly distinguishable colors, including shared nodes that are bright yellow.

Quantifying Uncertainty in Time Series Pre-Processing

In VA research and related fields, the awareness and need to incorporate uncertainty information into the analysis has increased considerably. This holds true for both a methodological, design, and implementation perspective. How uncertainty was introduced into the data can be distinguished by the different sources of uncertainty, including observations inherent to the data, generated by models or simulations, or introduced by the processing or visualization processes [PRJ12, BHJ⁺14]. Even though pre-processing inevitably introduces uncertainty by altering the original data, these routines are rarely analyzed towards their impact on uncertainty. When analyzing time series and MVTS, pre-processing is an integral part to enable further analysis. Several approaches analyze uncertainty introduced by pre-processing [CCM09, WYM12], aggregating uncertainty for individual processing steps. When assessing the influence of uncertainty on MVTS, inappropriate aggregation would omit temporal characteristics that can also be affected by processing.

The upcoming section will demonstrate a VA approach time series rastering that integrates DQ metrics and uncertainty to provide essential information for the rastering process, and to produce output metadata that gives insights into this pre-processing step. The DQ metrics increase the awareness of the introduced uncertainties and quality issues for further analysis. For discussing the critical rastering aspects below, I consider relevant characteristics of time and time series data from the work by Aigner et al. [AMST11].

6.1 Quantifying Uncertainty from Rastering

These designs and conceptualization were published in [BBGM17].

When rastering time series, unevenly distributed time points and their corresponding values are being aggregated and binned into evenly spaced time intervals, while still retaining the original data's structure. Rastering transforms the original data for the sake of (a) consistent value distribution, (b) smoother value curves, (c) and reduced data size. However, to adequately transform a time series for subsequent analysis requires extensive knowledge about the data domain as well as temporal data characteristics.

For illustrating the challenges and critical aspects in time series rastering I give an example of unequally spaced time series sensor measurements from the environmental domain. Such measurements contain various formats and are used in many application domains. The *Opensense Project* in the city of Zurich [LFS⁺12] measures different environmental variables, like meteorological data, air pollutants such as O_3 , NO_2 , NO , SO_2 , VOC , and fine particles. The interval length, with measurements varying around 20 seconds (s), has the following properties: median of 20s, interquartile range (IQR) of 15s, and median absolute deviation (MAD) of 1.4826s. Finding an adequate interval length for the rastering transformation is context specific and depends on domain properties. In this illustration, choosing a too short interval length, e.g., less than 20s in this example, would generate many raster intervals with no data, and therefore introduce missing values. On the other hand, too long intervals will mask interesting patterns in the time series. Immediate visual feedback on the new raster aggregation and important quality information are required to find an optimal configuration. This quality information includes introduced missing values, value ambiguity, or reduced temporal granularity.

6.1.1 Critical Aspects for Rastering Time Series Data

In many cases, automatic methods for rastering time series data are not effective due to mutually exclusive dependencies, e.g., reducing the amount of empty rasters and minimizing loss of accuracy. During data transformation and aggregation uncertainty information is likely to be introduced, as the data's structure is altered and sampling operations are applied. By sampling or aggregating values, the original measurement accuracy is lost. Current data processing systems merely store this information indirectly (i.e., provenance aware systems) if at all. By externalizing this uncertainty information, users are made aware of the impact of different rastering operations.

DQ information can also be helpful to assess effects of rastering operations on datasets. DQ metrics [PLW02] – proportional measures of data quality dimensions [Red12] – quantify quality aspects to give expressive assertions to certain data properties. The aim is to introducing DQ metrics that are specific to time-series data to allow informed rastering. The following list discusses contingencies that need to be considered when rastering different types of time-oriented data.

Characteristics of Time-oriented Data. Aigner et al. [AMST11] have extensively characterized properties of time-oriented data. One characterization is distinguishing time series data into either states or interval records. State changes occur either at the exact time entries were recorded at, or have changed at any time since the last

measurement. When converting information from individual timestamp values to equally spaced intervals, there is an inherent loss of accuracy in the temporal domain, and uncertainty is introduced in the value domain. When rastering a time series, the user needs to be aware of this varying influence of uncertainty with respect to different input time series and different rastering parametrizations. Consequently, the time series visualization requires appropriate representation considering these influential factors and results. When considering aggregating time series data containing intervals, original intervals are potentially split if raster lengths are incompatible. The time series visualization should represent the time intervals appropriately, and the rastering algorithm should feature options to allow retaining the original intervals' sizes or proportionally creating new rasters from multiple intervals.

Temporal Granularity. If time series need to be rastered with finer granularity than provided by the original data, data values of one interval need to be divided into smaller intervals – this division must be done based on assumptions, e.g., by computing a time series model based on the input data and super-sampling entries. Analogously, if the time series is rastered into a coarser granularity, details can get obfuscated, e.g., masking outliers by smoothing the time series through aggregation, and classical error margins may get broader. Depending on the goal of the user this is undesirable and should be indicated accordingly.

Ambiguity. It is implied that ambiguities might be introduced into the data during rastering, specifically when dealing with qualitative or discrete data values. Aggregating or sampling values during rastering often requires imputation from time series, or reducing raster granularity. Introduced ambiguity should be marked as such and explicitly communicated in further analysis steps. This information potentially influences analysis, specifically if users are unaware of inherent ambiguities and assume the data as explicitly correct.

Statistical Measures. How an optimal binning size is chosen can be inspected using statistical measures to provide important domain independent quality information, for example (a) mean range spread, i.e., the absolute difference how far values deviate from the mean, $\text{spread}_t = \sum_{i=0}^n \text{abs}(\mu_A - A_i)$ for interval I_t and values $A_i \in I_t$, (b) temporal deviation, i.e., how much a timestamp's t_{A_i} temporal value is changed during aggregation to the mean interval timestamp μ_T , $\text{tempDev}_t = \sum_{i=0}^n \text{abs}(\mu_T - t_{A_i})$ (c) point density per interval, i.e., how many timestamps are aggregated into an interval, $\text{tempDens}_{t_i} = \sum_{i=0}^n ||A_i||$, for all $A_i \in t_i$. These measures give insights into the amount of uncertainty introduced by different interval sizes and anchor points and should be considered when trying to identify a suitable rastering of the time series at hand. To construct more expressive measures, DQ metrics can be employed to aggregate this information to different granularity levels. They can provide local information and allow comparison of overall granularity measures.

Temporal and Value Outlier Detection. With robust outlier detection measures, outliers can be automatically identified and highlighted. However, judging if these outliers – either in the temporal domain or in the data domain – represent anomalies requires additional contextual information. Thus, it takes the user’s domain knowledge to reason about the identified outliers. As such, marking outliers as well as rasters which contain outliers and saving this meta information for subsequent analysis is advisable and allows more informed decisions.

Missing Values. Similar to tabular and relational data, empty intervals can signal quality issues, and more specifically for rastering tasks could imply inappropriate raster window size. The distribution and amount of empty rasters can be visually inspected for finding a suitable rastering.

6.1.2 Visual Analytics Approach

With these considerations and aspects laid out, in this section I conceptualize a workflow for rastering unevenly spaced time series data and illustrate the application of these principles in our VA approach. Figure 6.1 shows a mockup with a design that supports the workflow discussed below. A description of the composed multiple-coordinated views and their use can be found in the respective caption.

For the rastering transformation, an interval length needs to be determined that is appropriate for the original dataset and the usage of the transformed data. The optimal raster window size depends on the data domain, different data characteristics, the further usage of the data, quality information, and introduced uncertainties. In the current state, the user can interactively choose a raster window size (see Figure 6.1e). To assist the user in supervising the rastering process and determining optimal rastering results, our approach considers the special characteristics of time-oriented data to provide important contextual information. The provided quality and uncertainty measures need to be interpreted in the light of the users’ domain knowledge in order to draw correct conclusions from the rastering result. Moreover, time’s inherent structure is used for calculating statistical measures (see Figure 6.1c,d).

The time series Rastering Preview (see Figure 6.1a) is interactively browsable, showing a superimposition of both the original time series and a preview of the results of the current rastering configuration. This view also serves as input interface for defining the raster length and initial raster anchor point. These parameters are selected via *drag&drop* (see Figure 6.1e) in the Rastering Preview. During dragging, the raster values are calculated and interactively updated based on the current configuration (grey dotted line in Figure 6.1a). The multiple coordinated views are dynamically updated during the dragging interaction to show the impact of the chosen configuration on the rastering outcome, like raster length, distribution, and possible empty rasters.

I argue that knowledge about DQ and uncertainty facilitates the rastering and assessment process for users. Contextual information on temporal characteristics in the form of DQ



Figure 6.1: An overview of our interactive time series rastering approach. (a) The interactive Rastering Preview allows defining the raster window size through *drag&drop* interaction as well as comparing the current raster configuration to the original data. Alternating consecutive raster backgrounds and original value point colors per raster facilitates distinction. Empty raster intervals are highlighted by red segments. (b) In the Result History View users can compare previous rastering results represented by small multiple line charts. Selecting a quality indicator (in c) overlays the small multiples with a heatmap of individual quality measures per raster. This view can be switched to the *quality overview* which gives multivariate quality and uncertainty information on recent raster configurations (see Figure 6.2). (c) The *aggregated quality and uncertainty indicator* view features a sortable and customizable heatmap view representing the aggregated quality and uncertainty measures for each raster configuration in the history view. Color intensity corresponds to higher values (see Figure 6.2). (d) Overview information of rastering results, including meta information, general uncertainty measures, and introduced quality issues based on calculated DQ metrics. (e) *Drag&Drop* interactive selection of rastering window length.

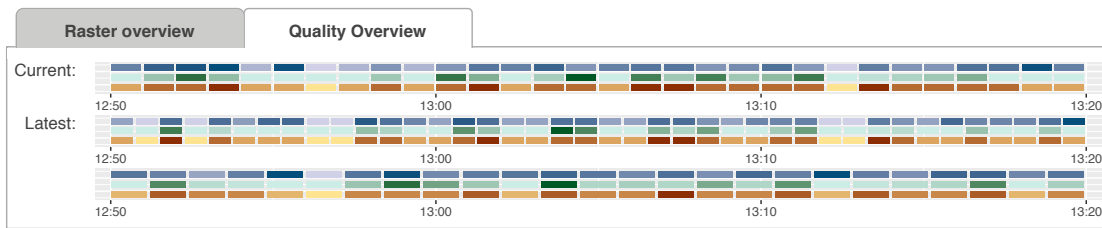


Figure 6.2: Result quality overview (*truncated*): In this view the user can directly compare quantified quality and uncertainty information of the raster result history for individual rasters encoded in a colored heatmap. The context of the coloring corresponds to the aggregated indicators in Figure 6.1c and helps to identify conspicuous entries.

metrics and uncertainty information allow the user to appropriately prioritize certain characteristics and assess rastering results, e.g., minimizing the median spread but consecutively disregarding the actual raster window size. These measures are shown in an aggregation overview to allow comparison with previous configurations (see Figure 6.1c).

Aside from showing the global quality and uncertainty information, the Result History View (see Figure 6.1b) shows a juxtaposition of previous rastering results as either small multiple line charts (*raster overview*) showing the time series and single quality measures, or as heat bands for displaying different quality and uncertainty measures at once (Quality Overview, see Figure 6.2). The view is interactively browsable and helps users to visually assess different rastering parametrizations. The Result History View furthermore allows for comparison between the latest rastering results to determine an optimal configuration where e.g., empty rasters are minimized without losing too much detail information due to value aggregation. The view can be interactively browsed to facilitate the exploration and validation of large time series.

To compare quality and uncertainty information, the *quality overview* (see Figure 6.2) allows to analyze quality information for individual rasters. For example, if DQ metrics indicate that ambiguities or missing values are less frequent in a particular raster configuration, it could pose significant benefits over small decreases of accuracy. With these analysis options at the user's disposal, the awareness about the influence of rastering transformations on the quality and uncertainty measures is increased. The approach allows to compare alternative rastering configurations with respect to the critical aspects outlined above and to the desired properties of the data for subsequent analysis tasks. Figure 6.1d gives a comprehensive overview of different properties from the current raster parametrization.

6.2 Quantifying Uncertainty from General Pre-Processing

In the previous section I illustrated one specific use case and VA design for addressing the task of rastering time series. However, similar problems persist for various other pre-processing algorithms. Specifically when pre-processing MVTs, a common processing

pipeline would consist of multiple consecutive steps: (1) imputing missing values, (2) performing linear interpolation, (3) smoothing the time series by applying a moving average kernel, and (4) sampling the data to reduce the size. How these processes influence uncertainty, but also how subsequent steps of downstream analysis are affected needs to be externalized and communicated to the analyst in a generalizable manner. In the Conceptualization Chapter (compare Chapter 3.2.2) I provided a formalization of uncertainty from MVTs pre-processing routines. This allows inspection of individual pre-processing routines. Consecutively executing pre-processing routines propagates uncertainties throughout the processes, which makes it increasingly difficult to determine the amount of uncertainty introduced at individual step. Applying the uncertainty quantification concept allows for adequately monitoring uncertainty during pre-processing and allow identifying individual steps that alter the value or temporal domain inappropriately. I will show the application of the uncertainty quantification technique in a concrete case study for pre-processing weather experiment data (compare Section 7.3)

6.2.1 Critical Aspects of Uncertainty Quantification for Visualization Design

The Uncertainty Quantification Cube distinguished uncertainty on the timestamp level, the data variable level, as well as uncertainty introduced at each step of a data pre-processing pipeline. To allow for each of these levels of detail, uncertainty should be quantified for the finest granularity level possible (i.e., for each timestamp, data variable, and pre-processing step), as higher levels of aggregation are not sufficient for **all** analysis tasks. If coarser uncertainty information is required to support effective analysis, this fine-grained uncertainty can subsequently be aggregated. On the other hand, it is not always possible to quantify uncertainty at the finest granularity level. Some pre-processing methods transform the granularity of the MVTs, such as dimensionality reduction or temporal sampling. The affected dimensions of the Quantification Cube need to be accounted for in the employed quantification method, because heuristic comparison of the pre-processing step's input and output values might not be feasible.

While the visual representation of uncertainty information and the need to include information about the uncertainty of the data that is visualized into VA environments gains awareness, it is often assumed that the uncertainty information is given. Yet, almost any data analysis is preceded by data pre-processing which also introduces considerable uncertainty into the data. The Uncertainty Quantification Cube formalizes the quantification and aggregation of uncertainty from MVTs pre-processing. Uncertainty can be analyzed to wither evaluate the appropriateness of the pre-processing pipeline as such, but also to propagate uncertainty into the final data representation to foster informed reasoning. This formalization helps visualization designers to understand and consider relevant aspects in this context. The upcoming section will describe how such different sources of uncertainty can be integrated into different representations of MVTs, and how different visualization techniques can affect the perception of uncertainty.

Part III

The Evaluation

Case Studies

In this chapter, I demonstrate case studies and usage scenarios to exemplify their application in real-world scenarios for the approaches presented in Chapters 4, 5, and 6. Exhibiting of visualization designs and VA solutions in a common work practice allow demonstrating and assessing their potential usefulness [IIC⁺13].

7.1 Case Study – Analyzing ISP Connectivity Data

This case study was published in [BGK⁺18].

In this case study, the MetricDoc environment is demonstrated in a real-world use case that (1) elaborates the functionality of our environment, (2) describes possible insights that would otherwise not be possible to obtain with existing approaches, and (3) shows a concrete analysis scenario of a real-world sample that shows how errors in a dataset can be discovered, and metrics can be customized based on the dataset at hand. By employing immediate feedback as well as effective interaction and navigation techniques analysts are able to iteratively develop DQ metrics and immediately incorporate them in their analysis. With overview and detail visualizations, the analyst can evaluate both new as well as updated datasets. As an example a DQ analyst explores a *net-test* dataset, an open dataset from the Austrian Regulatory Authority for Broadcasting and Telecommunications (RTR) to test Internet service quality (important data columns can be found in Figures 4.1 and 4.2 in Section 4.2; for more information please be referred to RTR's NetTest Documentation¹). The primary use of this dataset is to compare different Internet Service Providers' (ISP) service quality, logging information, like download and upload speed (in *kbit/s*), latency (in *ms*), and signal strength (in *dBm*). Also, anonymous meta information (device name, network information, unique identifiers, etc.) is collected, in order to compare different ISPs. The following tasks should be completed: (1) checking

¹<https://www.netztest.at/en/OpenDataSpecification.html>, accessed 02/11/2019

if outdated client versions have been used in recent connectivity tests, (2) inspecting implausible download and upload rates and ping latencies, and (3) developing a metric that highlights entries where performance issues occur when multiple tests are performed in a small time frame, to furthermore investigate if and how performance has an impact on test results.

To validate if only the newest client versions are present (i.e., browser clients 0.3, *iOS* devices 1.6, and *Android* devices 2.2.9) the validity metric of the *versions* column is customized by adding checks for these constraints in the metric customization view (see Figure 4.1c). Browsing the Metric Detail View (see Figure 4.1d) entries can be identified in the data that validate negatively against the constraints. This reveals that some devices are still operating outdated connectivity test versions (see Figure 4.1: The highlighted row in (e) shows a test performed on a Galaxy S5 with an outdated client version 2.2.5 instead of 2.2.9). After further browsing the dataset, three indications can be distinguished: Tests by desktop devices were all using the current client. For *Apple* devices, the analyst could not determine any consistent scenario when tests were performed by outdated clients. For the *Android* client versions it can be traced that mainly phones manufactured by *Samsung* (but not entirely) were still using outdated versions. By adding a check for Android firmware and analyzing distribution versions, it could be concluded that phones that have a firmware version older than 4.1 installed are not executing the latest client version. To make the metric more expressive and specifically determine how many *iOS* or *Android* devices were using outdated client versions, the current metric is split up and a quality check is added to the validity metric of the *platform* column. In the Quality Metrics Overview both metrics (validity metric of *platform* and validity metric of *client_version*) are selected and merged them to create an expressive metric across multiple columns.

For Task (2) the plausibility metric is leveraged for investigating implausible download and upload rates, as well as latency. Extremely low values, as well as extraordinarily high values might indicate DQ problems: unreasonably low download and upload rates could be caused by client issues, rather than actual bad connectivity and low quality Internet service. On the other hand, high download rates could be spurious entries that boost ISPs' ratings. The implausible values can be explored by simultaneously selecting the *plausibility* metrics for the columns *upload*, *download*, and *ping_ms*. In Figure 7.1 entries that were identified implausible by all three metrics are highlighted. These three *plausibility* metrics can be then merged into one custom metric. Logically concatenating them detects entries which have been detected as implausible in all columns, which is a strong indicator for erroneous entries. Showing only entries that violate the metric through the Metric Detail View's *only show dirty entries* button the dataset can be explored focusing solely on potential erroneous entries. On the other hand, it is also possible to highlight erroneous entries within the entirety of entries (preserving context information), by toggling *highlight dirty entries* in the Metric Detail View. Some detected entries implicitly indicate measurement errors, but naturally also positive outliers – performance tests with significantly high results – it becomes apparent that not all

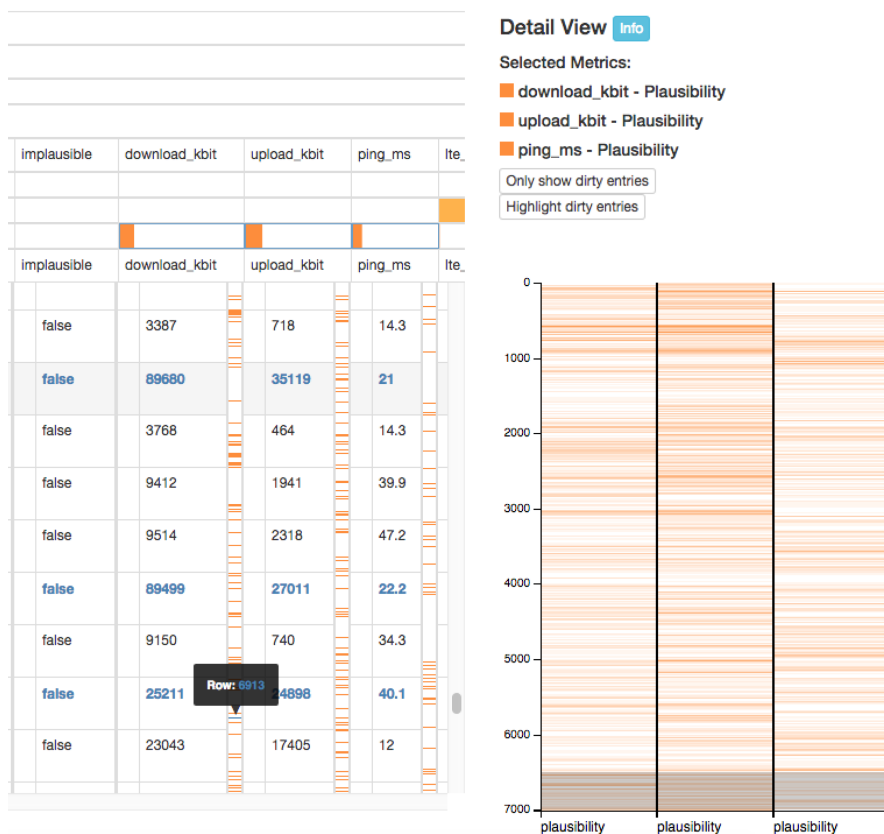


Figure 7.1: Task (2): Extreme values can be observed, these might be subject to erroneous generation skewing the ISP performance results.

entries detected by the *plausibility* metric directly correspond to outliers. The *plausibility* metric's parameters are adapted, switching from the default *robust* to *standard* outlier detection, as well as from *global* to *local* outlier detection, which only includes the latest entries for calculation. Hence, temporal server performance issues are not influencing outlier detection, which benefits the detection of actual implausible values. That way the *plausibility* metric gives contextual information only on significant changes in download and upload size, indicating outliers as expected. Since all metrics are implemented as functions, they can also be transferred to the OpenRefine wrangling tool to apply a filter for the implausible performance test entries and remove such implausible values from the dataset.

Lastly, for Task 3, performance drops are investigated on the assumption that they are linked to multiple connectivity test runs performed in quick succession. A *date interval* metric with columns *from* and *to* as parameters is added, highlighting entries lying within 10 seconds of the next, and checking them against *plausibility* metric defined for *download*, *upload*, and *latency* in Task 2. It is suspected that the server could be overloaded with connectivity test requests which leads to server bottleneck issues. A *threshold check*

n	implausible	download_kbit	upload_kbit	ping_ms	server_name	test_duration	num_threads	platform
n	implausible	download_kbit	upload_kbit	ping_ms	server_name	test_duration	num_threads	platform
	false	93	71	834.7	RTR Websocket AT	7	1	
	false	254	98	54.9	RTR-SERVER AT	7	1	iOS
	false	107951	44585	28	RTR-SERVER AT	7	3	iOS
	false	86942	38862	20	RTR-SERVER AT	7	3	iOS
	false	115	37	472.1	RTR-SERVER AT	7	1	iOS
	false	387	201	478.3	RTR-SERVER AT	7	1	iOS
	false	115	45	140.3	RTR-SERVER AT	7	1	iOS

Figure 7.2: Filtered raw data view showing only data that have been detected as erroneous in the currently selected metrics. Task (3): It can be observed that tests which had a restricted number of threads (see column `num_threads`) predominantly had low download and upload rates as well as high latency.

is added to determine if extremely low download and upload scores are present and is validated against entries detected by the 10 second interval metric specified before (see Figure 4.3). This leads to an unexpected insight: Not only entries with low download and upload rates occur, but also some that reach high rates. The combined date interval and plausibility metrics are able to highlight potential performance inconsistencies.

All customized metrics can be utilized for subsequent data exploration of newer connectivity tests, since the dataset is updated in monthly intervals. The previously created and customized metrics are readily available and can be re-computed within seconds for new data. The updated data and metrics can immediately be explored for identifying issues and validating changes. The newly calculated metrics can be directly compared to the old ones to iteratively check if inconsistencies that have been discovered in old datasets could be resolved in more up-to-date connectivity tests. Data columns can be quickly sorted by their dirtiness to check dirty columns more effectively in the Quality Metrics Overview.

7.2 Case Study – Analyzing Provenance from Wrangling Operations

This case study was published in [BGM19].

I illustrate a use case that shows DQProv Explorer in a concrete wrangling scenario.

Consider an analyst concerned with the task of wrangling a car dataset (see Appendix 12 for detail information): The analyst investigates the Issue Distribution View showing the automatically computed data quality metrics (compare Figure 7.3 Analysis Step 1) for three types of issues (invalid, incomplete, and implausible entries). It shows that 12 of the 33 total columns have issues that need to be taken care of. In the detail view of the initial data state, we can see issue patterns which indicate that a few erroneous rows are responsible for multiple detected issues (compare Figure 7.3 Analysis Step 1). After identifying the dirty data rows in columns that contain the most errors – namely the *weight*, *width*, *height*, *displacement*, and *miles per gallon* (*MPG*) column – and removing them in the data wrangling system, the analyst returns to the DQProv Explorer to check how many issues still remain. The analyst finds that most issues have been solved, but the *MPG* column still retains implausible values (compare Figure 7.3 Analysis Step 2). Upon inspection the analyst determines that these are the result of hybrid cars having better fuel efficiency and reasons that the metric shows false positives.

Upon further inspection of the raw data, the analyst notices that some entries represent electric cars, that should not be removed from the dataset, because otherwise electric cars would be omitted from the dataset. Hence the analyst reverts all operations and restarts the wrangling process. Filtering for NA values in the fuel column brings up multiple electric cars. The analyst proceeds to fill in missing cells (*cylinders* with 0, *displacement* with 0, and *MPG* with -1 because the column is not applicable and the numeric value will not create issues in further analysis instead of NA) and removes five data rows that exhibit missing values in multiple cells. For the remaining detected issues in columns *width*, *height*, *weight*, and *displacement*, the analyst decides to impute missing values with the column's median value instead of removing the entries, like in the first wrangling attempt. The analyst imputes all relevant columns' missing values and returns to DQProv Explorer for comparing the overall quality of the second wrangling branch with the first one, where quality was improved mainly by removing data entries (compare Figure 7.3 Analysis Step 3). Summary information on the provenance graph's nodes shows that the analyst could retain 293 rows in the second wrangling attempt as opposed to 244 rows in the first wrangling attempt (compare Figure 7.3 Analysis Step 4). The analyst continues with selecting two nodes for comparison and inspects the differences in overall quality of the two end states of the branches. It reveals that s/he could successfully remove the similar amounts of errors in the second attempt, but with the benefit of retaining more information by not removing data entries.

7.2.1 Discussion

The examples employ quality metrics to detect issues of the types completeness, validity, and plausibility. But our approach is extensible to different types of metrics: Using performance metrics from machine learning algorithms and allowing users to explore the results on different training data could lead to a better understanding of how influential the datasets are on the final algorithmic outcome. Also, measuring introduced uncertainty from wrangling processes could be quantified by quality metrics.

7. CASE STUDIES

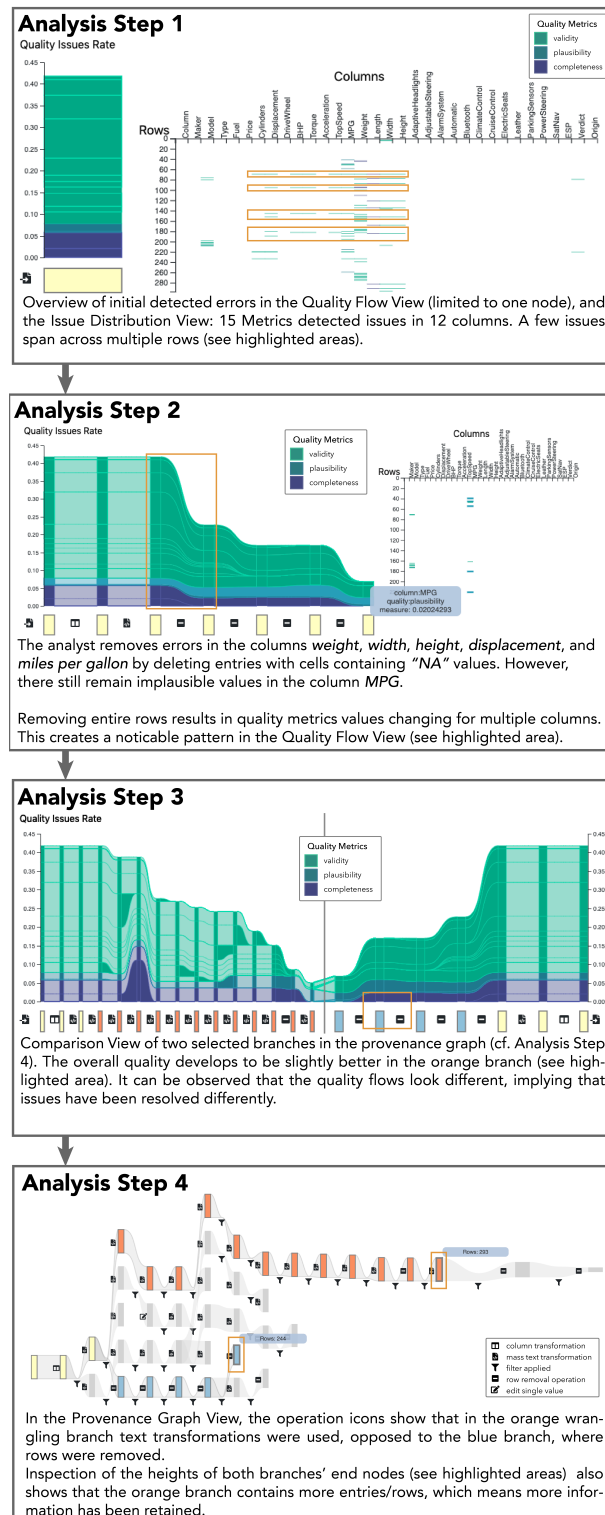


Figure 7.3: Visual overview of the wrangling process on a car dataset. The four steps show different stages of the analysis process and how the analyst can use the different views and interactions to determine if the overall quality has improved.

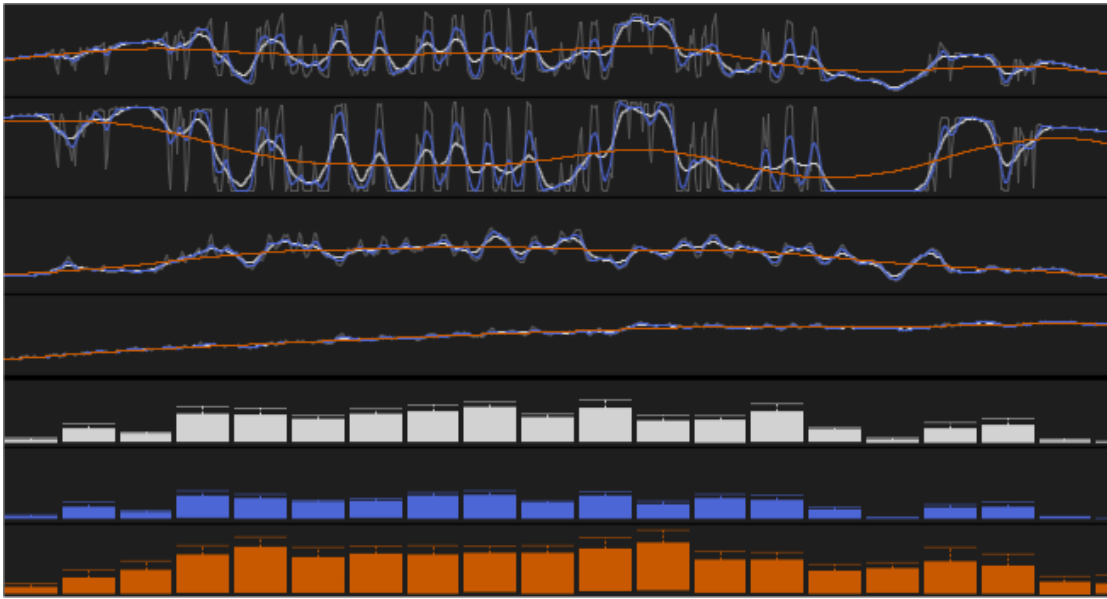


Figure 7.4: Analysis of a Moving Average pre-processing step. Multiple MVTs dimensions are visualized with three different parameter settings (top), for each parameter uncertainty is aggregated by dimensions and time to give three boxplots over time (bottom) [BHR⁺19].

7.3 Usage Scenario – Quantifying Uncertainty from Pre-processing Weather Experiment Data

This usage scenario was published in [BBB⁺19].

The MVTs processed in the scenario contains weather experiment data measured in Antarctica [RLKL⁺12] and used by our collaborator for downstream analysis. The two primary analysis goals of our collaborator are to improve data quality and to make the data more compact for a more efficient use. We exemplify the use of uncertainty quantification in a visual analytics tool for pre-processing of MVTs presented by Bernard et al. [BHR⁺19] to support analysis scenarios with uncertainty on different aggregation levels (Please be referred to this work for a detailed description of the interactive VA approach). Among others, it enables the assessment of (a) uncertainty introduced by a pre-processing step (compare Figure 7.4), (b) uncertainty influencing individual and multiple variables, and (c) uncertainty influenced by alternative pre-processing parameter values (compare Figure 7.5). For all steps and parameters used in the following examples, uncertainty is quantified as the normalized relative difference on a timestamp and individual variable level, $u_{rel}(z_{(t,v)})$.

First, we highlight how the collaborator applies a smoothing routine to remove noise and reduce the effect of outliers, i.e., to improve data quality. Figure 7.4 shows how the effect of the smoothing routine can be assessed for four dimensions and three different parametrizations (gray, blue, orange lineplots). Using aggregation by variable allows

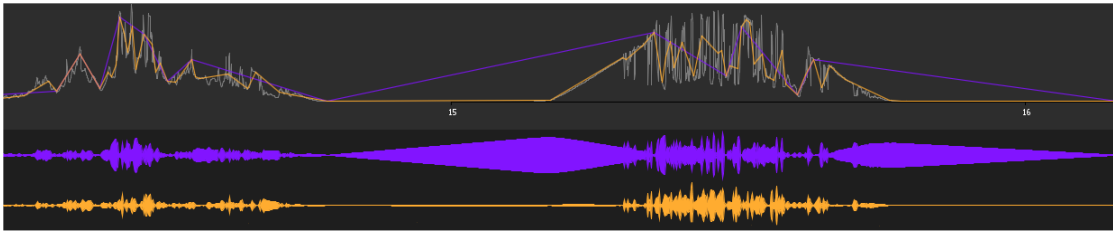


Figure 7.5: Assessment of uncertainty introduced by a sampling routine for one dimension, applied with two parameter values (purple, orange). The purple parametrization is too coarse, introducing a considerable amount of uncertainty [BHR⁺19].

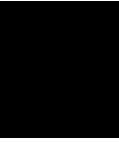
assessment of the average uncertainty across all selected dimensions, aggregation by time allows analysis of the uncertainty introduced for cyclic patterns observed in the first two dimensions. The orange boxplots on the bottom (compare Figure 7.4) indicate a considerably higher uncertainty with this parametrization and removes the cyclic patterns entirely. The collaborator proceeds by adding a sampling routine with two sampling window sizes, aiming for a more compact MVTS. To grasp the effect of the sampling routine at a fine-grained level, the collaborator inspects the sampling results (compare Figure 7.5) of one individual dimension of the MVTS (top purple and orange lineplots) and the corresponding uncertainties (bottom symmetric area charts), meaning we don't apply aggregation in the variable domain. It shows that the purple sampling routine introduces excessive uncertainty, due to a too coarse sampling kernel. Finally, the collaborator wants to validate the pipeline as a whole. Again, an adequate level of aggregation is used to exhibit the uncertainty of several routines. The uncertainties are aggregated over all variables, but shown for every pre-processing and timestamp individually. That way, the collaborator can identify which routines introduced the largest amount of uncertainty in comparison to the others.

7.3.1 Discussion

With the VA approach building upon the uncertainty quantification methodology, the collaborator was able to conduct the uncertainty-aware pre-processing of MVTS. She was able to make informed decisions in the creation as well as in the validation phase. The insights that quantified uncertainty from pre-processing provides show the extent to which MVTS were altered, if any signal was removed that was existing in the original data, or vice versa, i.e., signals are artificially introduced. Being provided composite and linked visualizations in an interactive environment lets the analyst assess these influences on different levels of granularity and aggregation. These varying levels of aggregation can, at first, give overview of the overall introduced level of uncertainty and signal intervals of the data that require further detailed investigation (compare Figure 7.5). Without a visual-interactive approach, selection adequate parameters would have required iterative comparison of intermittent processing results.

7.4 Lessons Learned

These case studies describe the usage of the developed VA solutions in real-world scenarios. They serve as prime examples how they are intended for analysts to solve their problems and in what way the implemented features facilitate common analysis. One drawback that has to be addressed is the controlled nature of these case studies. Even with extensive experience in their domain, analysts still face the challenge of getting familiar with the developed VA solution. Thus, visualization and VA experts also must validate if the employed visual encodings are appropriate for the tasks presented in these studies. It is difficult to guarantee the applicability of a VA solution based solely on the presentation of a concrete case study, so in the following chapters, I will also employ other evaluation techniques that can be used to validate visualizations and VA solutions with more rigor.



Iterative Design Process and Evaluation

The iterative design process was previously published in [BGK⁺18]. It was done in close collaboration with Simone Kriglstein and Margit Pohl, from the Human Computer-Interaction Group IGW at TU Wien.

Users' acceptance of VA approaches often depends on how well the approach considers users' tasks and needs. Thus, the interest in strategies from HCI to provide an iterative human-centered design process for VA approaches has increased over the last years [FPS14, KEM06, KW13, KKUFW06, SMM12, TM04]. The development of MetricDoc involved four iteration cycles (see Figure 8.1). This iterative process helped us to react to users' unexpected needs and expectations as well as to continually refine the design of the visual exploration environment based on well-known evaluation methods from HCI. The design and development of MetricDoc is based on the previously mentioned requirements and design rationales (see Section 4.1 and Section 4.2). In the following, I present the methods applied throughout the iterative design process and evaluation and give a retrospective summary of the insights gained in each iteration. These iterative design cycles were conducted and evaluated in collaboration with an HCI expert.

Methods

For the design and evaluation of our visual exploration environment, a combination of the following methods was used for the different iteration cycles:

M1 Prototyping [Gal07, HP12, MB02]. Prototyping is a popular method in HCI to collect feedback, to identify difficulties, and to refine the design already at a very early stage without losing too much time or money. During the design process of MetricDoc

varying fidelity levels of prototypes were prepared for heuristic evaluation and expert review sessions, as well as for a focus group session.

M2 Heuristic Evaluation and Expert Review [FJ10, Nie94, TFB⁺14, ZSN⁺06].

To detect a large number of basic design problems and to generate ideas for improving them, heuristic evaluation and expert reviews are advisable methods. For heuristic evaluation sessions I applied the visualization-specific heuristics developed by Forsell and Johansson [FJ10] and Tarrell et al. [TFB⁺14] which consider perception, cognition, usability, and interaction aspects. Furthermore, I conducted expert review sessions which were less formal than the heuristic evaluation sessions, focusing on the previously mentioned requirements and design rationales. The combination of heuristic evaluation and expert review sessions allowed us to get a holistic view in order to identify design problems. It gave us the flexibility to concentrate on specific problems or to discuss further design solutions.

M3 Focus Group [CB04, Kit95, MB02, PS96]. Focus groups are means to get a quick understanding of users' perception, experiences, expectations, impressions, and opinions about a design from multiple points of view. Based on our previous work (e.g., [KPS⁺14b]), I find that the dynamics and the open discussion in a group can stimulate new ideas and foster conversation about interesting design-relevant issues which would not happen in individual interviews. During the design process of MetricDoc, I conducted a focus group session with experts in the field of DQ, VA, and HCI in order to discuss and analyze the design from different points of view.

M4 MoSCoW Method [Bre09, CB94]. The heuristic evaluation, expert reviews, and focus group session were very constructive and many interesting design ideas were collected. To prioritize the findings I used the MoSCoW – *Must have, Should have, Could have, and Won't have* (but would like in future) – method. The benefit of the MoSCoW method is that it uses human language for prioritizing and not a specific scale which helps to quickly understand the concept of MoSCoW without prior knowledge or necessary training. This helped us to prioritize important design changes and in what order these changes should be implemented. It allowed us to pinpoint which features were missing but essential for the usage of MetricDoc, and what was least-critical but may be included in a future phase of the development.

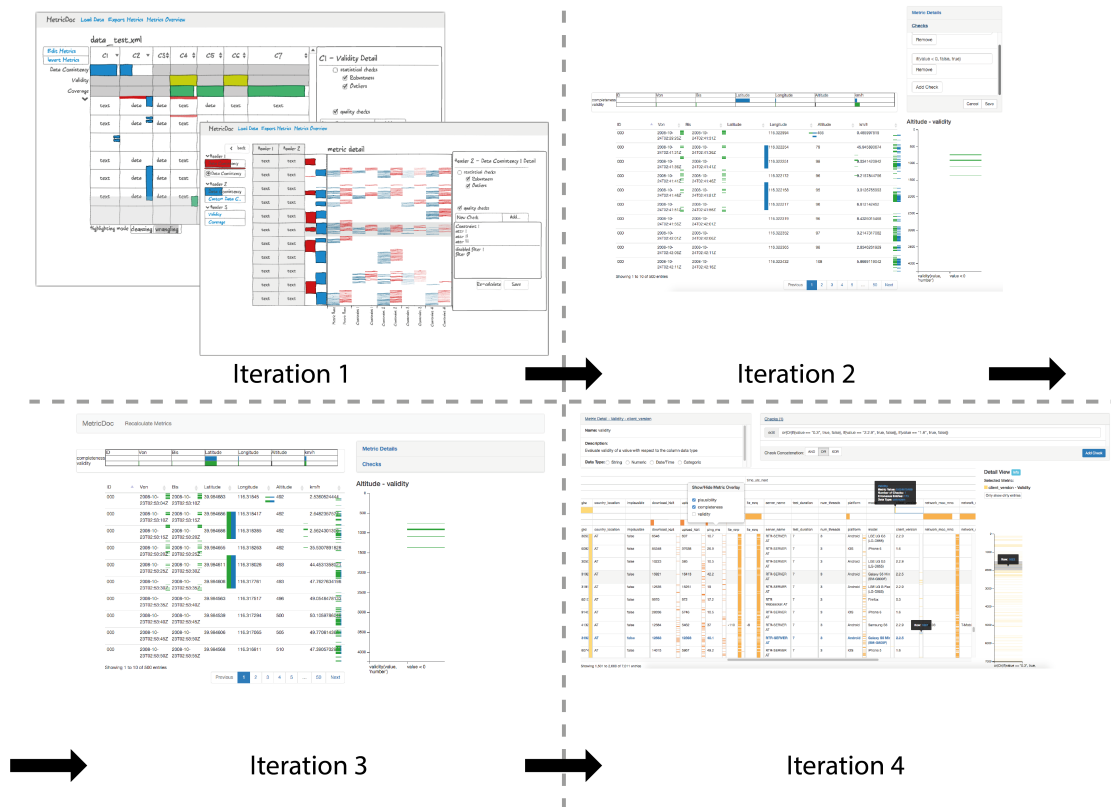


Figure 8.1: The different stages of the design process of MetricDoc.

8.1 Iteration One – Conceptual Design

In the first iteration cycle I concentrated on the creation of low-fidelity prototypes in consideration of the defined requirements (see Section 4.1). The goal of the prototypes was to explore different design ideas on how the DQ metrics for tabular datasets can be visualized in order to (1) provide an overview about the overall quality of a dataset and (2) offer detailed information about detected dirty entries and their position in the dataset. The concepts mainly differed in their arrangement of information and in the usage of different views (see Iteration 1 in Figure 8.1). For this purpose two different low-fidelity design concepts were created that differed mainly in their arrangement of information and in the usage of different views.

In an expert review session the different design concepts were analyzed and discussed by two experts in the field of HCI and VA. The experts went through each design concept to verify how well they support users in solving the tasks defined in Figure 4.1. Each of these two design concepts had their strengths and weaknesses. In the next step, both concepts were unified to have a foundation for the development of a high-fidelity prototype in the next iteration cycle. For example, an original idea of one approach was that the users

had to switch between the overview – showing the overall quality of a dataset – and the Metric Detail View – showing the Error Distribution Overview with respect to specific DQ metrics and self-defined checks. Thus, in this iteration I reached the following state:

- Features of the initial low-fidelity prototypes were carefully selected and consolidated into a conceptual design to build the **foundation of further high-fidelity prototype design**.
- Within this iteration cycle I had **not yet prioritized metric customization** as an integral part of our functionality design.

8.2 Iteration Two – Design Evaluation

Based on the conceptual design developed in the first iteration cycle, a first interactive prototype was developed. The focus of the first version of this prototype was to verify the interplay of the multiple views in order to ensure a good overview of detected dirty entries with respect to specific DQ metrics, the distribution of the detected dirty entries, the corresponding tabular representation, detail information about the DQ metrics and identified data types, and the creation of the quality checks. The prototype already included basic functionality, e.g., to create custom quality checks, to evaluate a specific quality metric, and to visualize the results of the checks.

In a two-round session, a heuristic evaluation and an expert review were conducted by two experts in the field of HCI and VA. The session started with the expert review part, which had the goal to analyze the functionality of the prototype and the interplay of the coordinated views. The prototype was furthermore reviewed in consideration of the tasks defined in Section 4.1 and the design rationales in Section 4.2 respectively, just as in the first iteration cycle. The second part of the session concentrated on the heuristic evaluation. For this purpose both experts assessed the prototype against visualization-specific heuristics [FJ10, TFB⁺14]. The output of the two-round session was twofold: On the one hand, the expert review revealed different suggestions for refining the functionality and the design of the Quality Metrics Overview, the Error Distribution Overview, and the Metric Detail View (e.g., interaction conceptualization and view ratios). The Error Distribution Overview visualization (see Section 4.3.3) in the tabular representation was not considered in this early stage of the prototype. On the other hand, the heuristic evaluation revealed a list of design and usability problems. For example, it revealed a violation of D1: different colors were used for the same DQ metrics to show the number of dirty entries and to visualize their distribution. In the next version of the prototype (developed in the next iteration cycle), I assigned a unique color to each quality metric to avoid confusion. This iteration led to the following outcomes:

- I conducted an expert evaluation according to established HCI heuristics, which led to a number of suggestions how to improve the design.
- The **Error Distribution Overview** visualization was **not yet considered** in the development of the environment.

- These suggestions were **prioritized** with the help of the *MoSCoW* method to identify which changes are essential and should thus be addressed in the next iteration cycle.
- Concrete **changes of the design were consolidated** for the next iteration cycle.

8.3 Iteration Three – Focus Group Evaluation

The main focus in this iteration was to (1) resolve the discovered design and usability problems and (2) implement the visualization of the distribution of dirty entries (with respect to the corresponding DQ metrics) in combination with the tabular representation. Since the developed prototype included sufficient basic functionality, a focus group evaluation was conducted with the goal to learn more about target users' opinions, their satisfaction with the current design, and to identify further directions. In order to get valuable discussions and ideas for the further development from multiple points of view, three DQ experts, one HCI expert and one VA expert (both were familiar with data profiling), and the developer of the prototype were invited. The focus group was held in a room with a live presentation of the prototype on a beamer setup, its duration was around two hours, and a skilled moderator, who was familiar with the domain, guided the discussion. The focus group was aimed at covering tasks derived from our requirements (Section 4.1):

- (1) To check a specific column with the help of a specific quality metric and to identify the resulting dirty entries in the tabular representation,
- (2) To customize an additional check for a specific quality metric and to apply the check to a specific column to identify which entities are affected,
- (3) To compare two DQ metrics and to identify dirty entries with respect to one or both DQ metrics.

Furthermore, a list of questions was prepared to find out participants' opinion about the design solutions.

The focus group session was free-flowing with interesting and valuable discussions about the design and possible improvements of the prototype. The DQ experts highlighted that the prototype was powerful for checking the different columns with respect to different DQ metrics and for developing custom-made checks and customized metrics respectively. Putting the Error Distribution Overview in a separate view was noted as helpful also in combination with the tabular representation. The experts commented that it would allow users to not only concentrate on the analysis of the distribution of dirty entries but also see the distribution in context with the table. All participants agreed on the benefits of retaining the Metric Detail View and Error Distribution Overview side-by-side instead of combined isolated visualizations. Other suggestions on visual presentation and design improvements included: avoiding the color green (see Figure 8.1 Iteration 2 & 3) as it confuses users due to the color being associated with positive feedback (unanimous among participants), adding a color legend, adding a heading to the Metric Detail View, and providing zoom functionality in this view (VA expert). With growing understanding

of the tool the DQ experts wished for more means to make changes to metrics, e.g., merging metrics, previewing customized metrics, saving and exporting metrics. Hence, the focus group session led to the conclusion that providing a comprehensive metric customization interface could enable DQ experts to develop metrics more efficiently. After the focus group session, the developer and the moderator discussed their notes to consolidate comments, improvements and design problems.

This list of suggestions was concluded and subsequently prioritized according to the *MoSCoW* method:

- The core feature set of the MetricDoc environment was shifted from exclusively **exploring DQ issues** with pre-defined metrics (with the ability to change parameters) to also **developing metrics**.
- Instead of implementing the suggested preview window for customizing metrics, I chose to provide **direct feedback to customizing metrics** by validating them syntactically during editing. Metric re-calculation is performed upon saving the metric.
- DQ experts' suggestions for more sophisticated validation methods were categorized as *Won't have* (they would be nice to have but could not be realized in the current state of the prototype, due to development costs).

8.4 Iteration Four – Final Development and Inspection

The goal of this iteration was to gather feedback from **DQ experts** on the design of the prototype (as in the second iteration, the revised prototype was analyzed by HCI and VA experts). The HCI and VA experts verified how suggestions for improvement brought up during the focus group discussion were realized. They checked if the noted design issues were addressed adequately and if visualization-specific HCI heuristics are satisfied [FJ10, TFB⁺14]. Furthermore, open questions which occurred during development were settled. For example, they discussed design ideas how the metric overview bar could be split into multiple rows, indicating not the overall quality but each quality check separately. It was also discussed how linking and brushing can be improved to emphasize the connection between Error Distribution Overview and the Metric Detail View. The resulting list of improvements as well as of design and usability issues from the heuristic evaluation and the expert review session were subsequently discussed and prioritized. The following changes were applied to the final prototype:

- Visual clarity was criticized during the expert review, so the prototype was adapted by **adding separators between views** and adequately **aligning the environment components**.
- It was hard to determine if the current data table was only showing filtered rows (the data could be toggled to only show erroneous entries), this was improved by adding a **visual cue** (grey background in the Metric Detail View) to **indicate non-dirty rows are hidden**.

- Linking and Brushing was improved by **highlighting the currently hovered row** of the Metric Detail View in the raw data table. **All dirty rows** can be **highlighted on demand** in the raw data table, to facilitate browsing and exploration with context information about dirty entries.
- During the Focus Group evaluation I discovered that DQ experts – while appreciating visual representations – also **expected information on numeric values of metrics**. Thus, contextual information was added for metric customization: number of checks, number of erroneous entries, and the actual quality metric value. Additionally, **notifications inform** the user about how the **last change** has influenced the metric (see Figure 4.1k).

8.5 Results

User preference is considered to be important to improve acceptance of MetricDoc among DQ experts. Design and development was focused on providing diverse interaction and exploration techniques to support users during data exploration based on DQ. Under this premise our environment supports different workflows for metrics customization: Both the creation of multiple simple metrics as well as the development of few highly sophisticated metrics for validation are possible and similarly expressive for data exploration. Simple metrics allow a more comprehensive overview and detailed information on syntactic issues. Metrics that feature multiple complex quality checks allow for swift data profiling of recurring datasets and determining semantic errors. I also note limitations in terms of the complexity of metrics that can be developed in MetricDoc. Time-oriented DQ metrics and checks are currently available with GREL scripts and probing functions. In order for users to take advantage of the entire GREL function set and to properly integrate these functions into sophisticated checks and metrics, a visual scripting engine would be required.

Throughout development visual support for exploration and customization tasks has been prioritized. Immediate visual feedback is provided when changes occur, e.g., due to metric re-calculations. Both Metric Detail View and Error Distribution Overviews were optimized to support exploration and comparison of errors: Mouseover tooltips give contextual entry information, scrolling informs the user about the current position in the dataset, and Error Distribution Overviews can be disabled separately, if the user prefers a more classical exploration style without additional visual information. Moreover, the heatmap columns supporting quality checks can be resized in width to facilitate comparison of the results of two or more quality checks, e.g., to check for error correlations between columns or between different metrics. There are potential scalability issues with larger datasets (e.g., >100.000 rows), but they can be circumvented in the tool's current state: While the development and customization of metrics can be done on a representative subset of the data, the resulting metrics can subsequently be used on the full original dataset for quality assessment. Additionally, users can swiftly re-apply existing metrics which have been created for older datasets to updated or new datasets – with the same or similar structure. With structural changes in the data, the tool allows

Iteration Cycles	I	II	III	IV
Environment Design				
Visualization Design				
Metrics Conceptualization				
Functionality				
Interaction Design				

Iteration Cycles	I	II	III	IV
Metric Detail Views				
Metric Overviews				
Raw Data Table				
Customization				
Interaction				
Brushing & Linking				

Table 8.1: Distribution of development (orange) and design (cyan) efforts over the course of the four iteration cycles and beyond. Color saturation corresponds to increased effort of development or design during a specific iteration cycle. The proportionate efforts were determined by qualitative content analysis [Sch12].

users to adapt metrics flexibly and assess the impact of the metrics on the dataset, supported by the employed visual feedback and visualizations.

8.6 Discussion & Lessons Learned

An iterative design process with short cycles of development and testing had the benefit that I was able to discuss and test different design ideas. Moreover, it allowed us to react flexibly to design changes without losing time and investing unnecessary resources. Time plays a very important role for companies and influences their decision to conduct an iterative human-centered design process (compare [KPS⁺14b]). For evaluating MetricDoc I intertwined iterative prototyping and development with heuristic evaluation, a focus group, as well as expert review sessions. One benefit of this iterative prototyping and development process is the possibility to quickly elaborate different design ideas and dynamically evaluate them throughout the entire design process. This also allows shifting design efforts to focus on specific issues which were discovered during evaluation and reviewing. Table 8.1 shows a juxtaposition of changes in all development stages, indicating shifts in development (see left table with orange highlighting) and design (see right table with cyan highlighting) as a result of feedback that was gathered in the prior cycle. This table was created retrospectively based on keywords gathered from notes, commits (from *git*), and the *MoSCoW* prioritization list, that have been counted and categorized to quantify development and design efforts throughout design. This method is named

'Qualitative content analysis' [Sch12]. It can be seen that after each of the design cycles development shifted to different areas, which is likely due to the the implementation of specific functionality (according to milestones specified for this iteration cycle), but it can also be observed that areas that had already been targeted in earlier cycles were re-visited, due to usability issues and suggestions by expert users.

In addition to the changes highlighted after each iteration cycle, I point out significant revisions of the final prototype that were concluded from insights gathered during this iterative design and evaluation process:

- To better support the comparison of dirty entries with respect to different DQ metrics, the Quality Metrics Overview, Raw Data View, and Metric Detail View were designed as multiple views, instead of the merged view that was initially planned.
- DQ experts repeatedly stressed the importance of adding additional interaction techniques to both metrics and exploration features (brushing and linking, highlighting, etc.) as well as contextual feedback during metric and quality checks editing. This led to a shift towards better supporting metric customization, rather than solely providing predefined metrics and checks. These predefined metrics and checks now only serve as starting points for more complex data validation and quality assessment indicators.
- I discovered scalability issues with the initial design of the Metric Detail View that resulted in over-plotting when dealing with datasets of high row counts. During the focus group this feature was overlooked due to the limited size of the demonstrated test dataset.

Especially with early low-fidelity prototypes I could observe that the ideas were discussed more critically and, therefore, it was possible to more easily identify interesting alternatives as with high-fidelity prototypes. The course of the focus group including scenarios, tasks, and questions was prepared before the session started. The structure was, however, maintained to be flexible to allow for deviations from the predefined schedule. This resulted in discussions about the prototype, unexpected suggestions for improvement, and useful ideas for the further development (e.g., to integrate the possibility to show or hide specific elements based on the user's preference). From this relaxed atmosphere new ideas sparked, also in terms of the environment's potential usage in different application fields: One expert noted that the prototype could be also valuable for developing a powerful visual search environment in order to find specific data entries in tabular datasets. This lead us to the conclusion that along with different application scenarios, users expect different features that complement their own workflows, which results in different functional requirements for MetricDoc.

8.6.1 Lessons Learned

Since I considered various perspectives from different domains of expertise during the different iterations of MetricDoc I not only had the possibility to assess progress and

get feedback from different points of view, but I could also identify differences in the analysts' background knowledge which resulted in diverse expectations regarding usage and interactions. It confirmed our emphasis on offering different interaction techniques to users based on the usage of the environment. However, I also encountered difficulties regarding further evaluation. The variety of approaches of assessing DQ implies that there are multiple valid practices towards determining quality issues, but also that experts of varying domains are satisfied with different levels and types of dirtiness in the data. Hence, designing a usage scenario that covers all functions of the environment, without forcing users to follow a particular workflow, is challenging. The development of our environment was focused on gaining insight into the state of a dataset's quality. This also poses a difficulty for evaluation, since the level of insights may vary greatly depending on user behavior and how adequately the usage scenario matches a user's personal approach of determining DQ. To construct a comprehensive usage scenario that covers different kinds of insights, usage, and customization of DQ metrics, as well as utilizing multiple views for exploration, and evaluating them towards other data profiling and quality metric tools, is out of scope of this paper and will be the subject of future work.

Qualitative User Experience Evaluation

After finishing prototype implementation of DQProv Explorer, I conducted a user experience study to determine if it enables users to analyzing provenance generated from data wrangling workflows. I recruited 6 participants (4 male, 2 female; 1 Master Student, 4 Doctoral Students, and 1 Post-Doctoral Researcher in Computer Science) with varying degrees of experience in both data quality assessment and visual data analysis. The self-assessed expertise (from (1) = *novice* to (5) = *expert*) of users ranged from intermediate (3) to expert (5) in data wrangling. Expertise in visual data analysis ranged from novice (1) to expert (5).

The goal of the study was to answer if the tasks defined in Section 5.3 are sufficiently supported by our prototype. I specifically formulated the questions: (1) 1. Can participants determine if quality has changed, and can they decide if the data is usable for subsequent analysis? 2. Are the participants able to compare branches to assess the difference in operations applied to the data, and decide which of the branches poses the most useful dataset for their analysis? 3. Does the prototype allow the users to derive which quality issues were inherent in the dataset and how they were resolved?

9.1 Procedure

Due to limited time with participants, I gave an introduction into the visual encodings and interaction features of the prototype. The investigator assigned them to complete prepared tasks. The participants were encouraged to think aloud while conducting the tasks. Important actions and comments during the tasks and participant feedback after the session were noted. The sessions took between 75 and 90 minutes and were structured as follows:

Introduction Session (10-15 Minutes) If necessary, the participants received an introduction into data wrangling and quality metrics to clarify the scope of analysis, specifically because participants had different expectations of a usable dataset. The investigator then exhibited the general functionality and visual encodings of the prototype.

Task Assignment (30-40 Minutes) Participants were instructed to conduct tasks that were oriented around our requirements analysis (compare Section 5.3). Questions were prepared for each task to guide iterative analysis. If the participant did not provide enough information, the investigator would ask intermittent questions and to suggest possible alternatives to exploring the provenance data. Specifically, questions were intermittently asked to determine what type of provenance participants relied on when conducting analysis.

Interview (10-20 Minutes) In the interview, the investigator asked for feedback about their experiences with the prototype. The participant should reflect on the usability and usefulness of DQProv Explorer. This was done to encourage participants to express difficulties they encountered during analysis and to collect suggestions how these could be resolved. The feedback was collected in an unstructured way, participants could express their comments and suggestions in any way they preferred.

Questions During each separate task the investigator asked participants a series of questions to stimulate iterative exploration and cover the tasks laid out in Section 5.3:

T_{act} & T_{pres} - Look at the first state of the dataset and identify the column with the most issues (Column *'weight'*). Now look at the end node of one transformation branch and determine how quality evolved for this column. You can see multiple transformation branches: How different are the two branch end nodes in terms of quality, do similar issues remain? Can you find out what transformation/operation impacted the quality of this column the most?

T_{meta} - If only the dataset of the second branch was available for analysis, what columns would you use for analysis. If you look at the three different branches and compare remaining quality issues, which one would you choose for analysis, and for what type of analysis? (The *'weight'* column was affected differently in different branches, compare Figure 7.2 Analysis Step 4)

T_{rec} & T_{rep} - How did a sequence of actions influence the data? Going back to the Weight column, which of the branches would you use for analysis?

T_{coll} - Can you determine the user's objective in the sequence of transformations shown in the branch at the bottom of the provenance graph?

9.2 Results

I summarize the results and provide an overview of feedback that was given by multiple participants (a detailed breakdown of the user study and summarized feedback from participants on the different views can be found in Appendix 12). The questions were solved by all participants, with the exception of one participant not being able to solve questions for \mathbf{T}_{coll} (the participant had the lowest self-assessed experience with data wrangling). In summary, two different methodologies could be observed for assessing quality issues, based on the participants' patterns of exploration. Two participants iteratively navigated the provenance graph in an detail-first, overview later approach (mainly exploring the Provenance Graph View, using the Quality Flow View for quality inspection). The remaining four participants pursued an overview-first, details on demand methodology (mainly using the Quality Flow View for exploration, and the Provenance Graph View was used only for selecting different branches, and for on-demand context information).

Furthermore, I could find implications that the trust in the employed data quality metrics and the trust towards the wrangled dataset depends on the participant's expertise in data wrangling. While two participants simply accepted the metrics as being accurate and subsequently found the Quality Flow View to be sufficient for determining the validity of the dataset, two participants would refuse to make a final statement on the data's quality without exploring the raw data. Specifically participants with higher data wrangling experience demanded for more brushing and linking features, which to us indicated that familiarity with these tools makes users more confident to use complex interaction techniques. Two users suggested to add filtering techniques and toggling techniques to enable more focused exploration on particular types of changes.

Feedback from participants on the different views was mixed. While generally the Quality Flow View and the Provenance Graph View have been well received, the usefulness of the Issue Distribution View was questioned by the majority of participants. This view extended the concept of a *schematic error view* presented in Section 4.3.2, which was adapted to show the distribution of errors across all columns. Participants showed no interest in this view. Two participants also noted that auditing the data wrangling process of someone else by exploring the provenance graph increased their confidence in the data. This implies the usefulness of DQProv Explorer for hand-off tasks in collaborative settings.

9.3 Discussion & Lessons Learned

The study results show that DQProv Explorer was well received, even though some features were not deemed as necessary by participants. Generalization of the feedback is questionable due to the small number of participants (6), and inappropriate participant expertise. This tool is unique in its ability to explore workflow and data provenance from data wrangling, hence it was not possible to use comparable tools in the evaluation. In

particular, the Quality Flow and Provenance Graph Views feature custom visualizations to display data provenance specific to wrangling, which is not possible to appropriately encode in general workflow provenance visualizations.

I can neither confirm nor deny the proposition that leveraging data quality metrics aids the user in understanding the quality of the dataset. However, one interesting observation from the user study was participants' different perception of quality: While some considered each entry of a dataset as valuable, preferring imputation of values over removal of entries, others solely depended on the quality metrics to signal quality issues and considered the absence of issues as sufficient. The participants' comments during task execution imply that DQProv Explorer supports users in making sense of the wrangling history and in estimating the usefulness of the resulting data, based on the user's subjective perception of quality. Particularly when participants were asked questions during executing the understanding task \mathbf{T}_{coll} . However, this means that data quality metrics must be carefully developed and adequately used, because it could also lead to perceiving low/high quality mistakenly.

9.3.1 Lessons Learned

Evaluating the DQProv Explorer in a qualitative user study could validate whether a provenance-driven approach would appropriately communicate qualitative aspects of the dataset to the users' content. The interview-like character of the evaluation sessions allowed providing feedback for the eventuality that participants ran into dead ends during analysis. Only a brief textual introduction to the complex environment could not have permitted participants to perform analysis without extensive practice. The study design effort could be kept to a minimum: A use case was defined that covered all features of the prototype, with the ability to lead the discussion to explore different features and answer all defined questions. That way, the evaluation sessions could be used to evaluate both analysis to simply convey general DQ to participants unexperienced with DQ assessment, but also receive detailed feedback from participants that had extensive data wrangling and cleansing experience. Participants' behavior lead to desirable features was not anticipated during design. In a controlled study, participants would have difficulty to express their desires because pre-fabricated examples only evaluate existing features.

Visualizing Uncertainty of Segmented Time Series

MVTS often are complex and high-dimensional, which renders inspection and comparison of individual variables difficult. Analysts apply pre-processing techniques to reduce dimensionality and segment the data in order to discover notable patterns that could otherwise not be found without automated segmentation. These techniques take advantage of inherent temporal characteristics to cope with the complexity of the data. Bernard et al. [BDB⁺16, BBB⁺18, BHR⁺19] presented numerous approaches that employ pre-processing and segmentation pipelines and workflows to facilitate the analysis of time series in various domains. I stress the importance of uncertainty-aware processing (compare Chapter 6). It is imperative to integrate uncertainty quantification techniques into segmentation pipeline and result analysis. Chapter 6 showed a conceptualization of uncertainty quantification techniques for pre-processing algorithms. Furthermore, segmentation and labeling algorithms associate probability values to their results, which allows designers to leverage these probabilities as uncertainty indicators in the result visualization [BBB⁺18]. Bernard et al. [BBB⁺18] provide a segmentation pipeline that allows running the workflow for multiple parameter settings, combining pre-processing and segmenting algorithms to find an optimal segmentation result and determine influencing parameters for such a successful segmentation. These methodologies yield different types of uncertainties in MVTS that can subsequently be leveraged in subsequent visualization and VA solutions.

Bögl et al. [BBGM18] characterized various types of uncertainty inevitably introduced or generated over an entire processing, segmenting, and labeling pipeline. They distinguish value uncertainty, result uncertainty, aggregation uncertainty, and cause & effect uncertainty. To assess the influence of these different sources of uncertainty in MVTS, it is necessary to provide uncertainty-aware visualizations during exploration of the results. Bernard et al. [BBB⁺18] presented initial designs for showing value and result

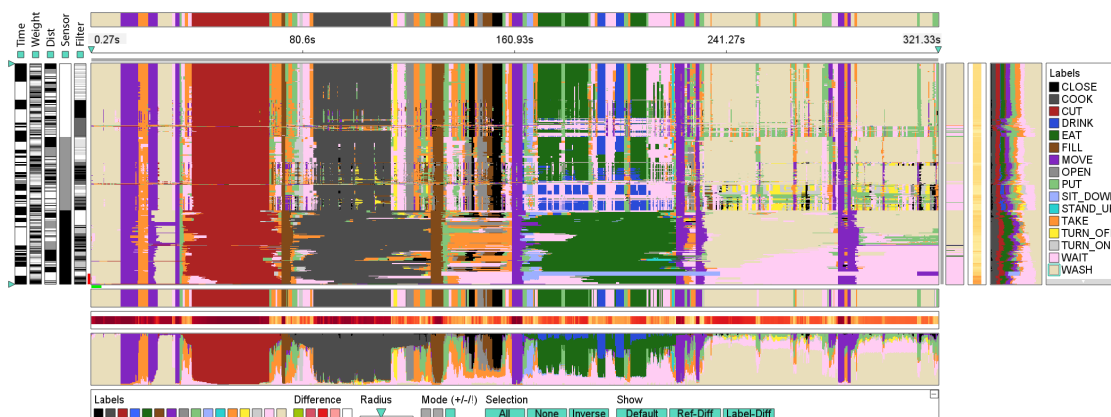


Figure 10.1: Schematic illustration of the VA approach for analyzing activity recognition algorithms by Röhlig et al. [RLK⁺15]. The central overview shows a large set of segmentation results encoded as a pixel-based visualization showing one pixel per segmentation result over time. The color encodes the assigned label for every timestamp. On the left hand side, employed parameter values are encoded as color coded stripes. The top view shows the ground truth of the current MVTs. In the bottom view, the currently selected segmentation result is shown with the label probabilities for every timestamp and a heatband denoting the probability of the dominating label.

uncertainties in different view modes. However, it is unclear if this design is scalable for exploring a large number of results, for parameter selection and uncertainty assessment. In the upcoming section I will discuss the requirements and design aspects a scalable designs and evaluate it in a user study testing the effectiveness of different uncertainty visualization techniques.

10.1 Design Goals

One challenge for analyzing VA is visualizing large sets of time series for segmenting and labeling tasks. Röhlig et al. [RLK⁺15] defined appropriate visual encodings for exploring large segmentation result sets (compare Figure 10.1), and Bernard et al. [BDB⁺16] identified obstacles for designing visualizations and VA solutions. Keeping these considerations in mind, I define the following design goals that ensure the uncertainty-aware visualization design does not interfere with current encodings but rather complements it.

D1 Faithfully represent label encodings for segments. The visual encodings of uncertainty should not interfere with colors of labels.

D2 De-/Emphasize segment or interval uncertainty (compare [OJS⁺11]). Depending on analysts' goal, the visual encodings of the visualized segments should highlight or de-emphasize regions of high uncertainty.

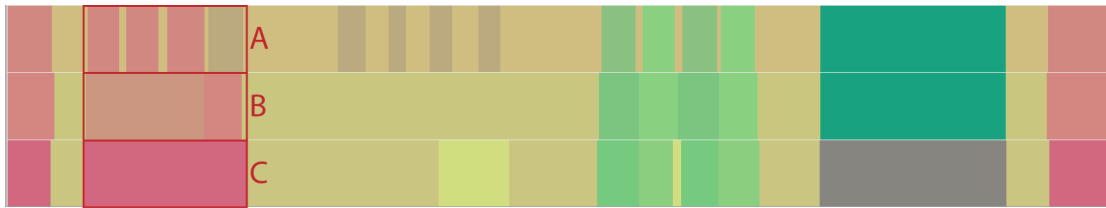


Figure 10.2: Example visualization showing three segmentation results computed by the same segmentation pipeline, with different parameters: The colors indicate differently labeled segments, with each color corresponding to a certain label. The highlighted areas (A, B, and C) show a very different segmentations of the same time interval. But how certain are these different segmentations?

D3 Support uncertainties with varying dimensionality and scale. The dimensionality of uncertainty depends on different factors (compare Section 3.2.2), hence, the visualization techniques should accurately represent uncertainty at any aggregation level.

10.2 Visualization Design

Following the visual design from existing approaches [RLK⁺15, BDB⁺16], and based on numerous studies of uncertainty visualization for time series [GBFM16, WBFL17, FWM⁺18] I designed visualization designs for encoding uncertainty in a label-based segmentation result visualization. These visualization designs satisfy design goals D1–D3, while still supporting integration into the aforementioned visual exploration interface for segmenting and labeling. Multiple types of uncertainty are inherent to the data or are introduced at various steps in the pipeline. As a result, one single view is insufficient for assessing the influence of uncertainty on the segmentation result or other uncertainties, respectively. I distinguish between (1) the Overview and (2) the Detail View. The Overview is designed to show the entire set of segmentation results and associated uncertainties, allowing the analyst to explore patterns and local phenomena. On demand, the Detail View shows multiple types of uncertainty juxtaposed, to allow comparison and detailed inspection. One important goal of this study is to determine the most appropriate visual encodings for value, aggregation and result uncertainty [BBGM18]. This is done by conducting a comparison task: The overall aim of our visualization design is to make a qualitative comparison of the inherent uncertainties between results, individual segments, or sequences of segments.

Comparison Task. The task for the analyst is to determine areas that exhibit either high or low uncertainty. For example, Figure 10.2 shows three segmentation results. The segments in the red frames (A, B, and C) show different results for the same time interval, however each result segmented this interval differently. Without showing the uncertainty associated with these segments, the analyst can not assess the trustworthiness

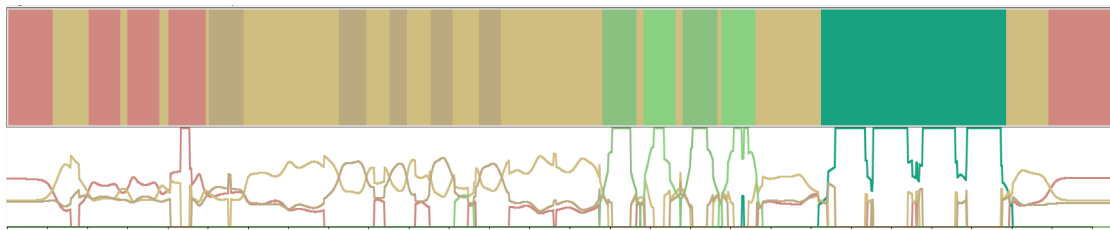
of the results. Consequently, the visualization design should allow direct comparison to determine the most certain/uncertain area.

Visualizing Probability-based Uncertainty. The design guidelines to develop alternative overview visualization designs were obtained from the results presented by Gschwandtner et al. [GBFM16]: The segmented time series' result uncertainty is caused by the classifier detecting a different segment to be more likely, causing a transitions between segments. However, the end of one segment marks the start of the next (except for the last segmented interval of a time series), which differs from a single temporal interval being displayed. Due to the potentially large number of time series shown in the Overview, a pixel-based visualization technique is required. Reviewing the designs evaluated in [GBFM16], I found the gradient plot and the disambiguation plot to be appropriate for displaying decreasing and increasing probabilities simultaneously, as well as being used in a pixel-based visualization. In the Detail View, the remaining visualization techniques (i.e., error bars, centered error bars, violin plots, and accumulated probability plots) could be employed, but changing the visual encoding in different views is not recommended and thus was avoided.

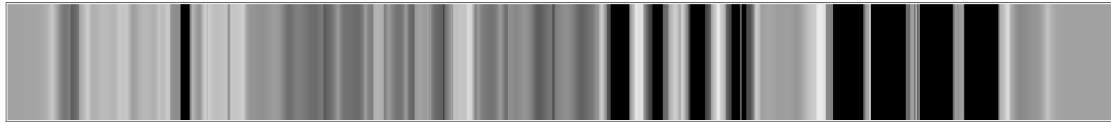
To provide points of reference to compare our visualization design against, we used two additional visualization techniques for encoding uncertainty: an heatmap that encodes uncertainty as grey values, omitting label colors, and a composite visualization showing segments in the top view and result uncertainty as a line plot (compare Figure 10.3).

Composite view and Heatmap designs are expected to allow users to accurately determine uncertainty values: The Composite view encodes uncertainty as location information in line charts, which permits the most appropriate encoding of quantitative values [Mac86]. The Heatmap view encodes uncertainty as grey values, omitting label information, so there is no additional comparison of colors to be made by the user. For the Gradient Uncertainty plot I expect similar performance to the Heatmap view, but adding color encoding for labels could lead to falsely perceived uncertainty. To mitigate for problems to distinguish between two similarly looking color and saturation combinations, the view can be interactively toggled between showing the Gradient Uncertainty plot, or a plot showing only the segmentation result. Using a Threshold Uncertainty plot reduces information to uncertainty of an interval being below or above a specified level of uncertainty. To make it possible to analyze different thresholds of uncertainty, this view features a slider that changes the disambiguation threshold.

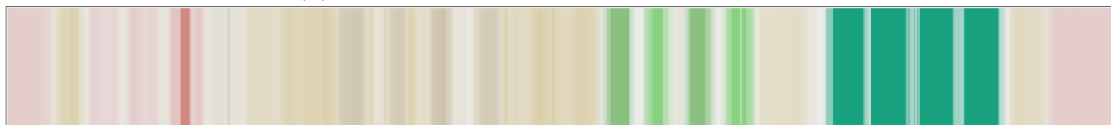
Visualizing Distribution-based Uncertainty. Value uncertainty is quantified either directly from the input data or is externalized from the employed pre-processing pipeline, and usually stored alongside the time series. While some algorithms and procedures may alter individual timestamps (e.g., outlier removal or imputation), others can also affect the entire time series (e.g., sampling), it could be more appropriately treated, like an additional dimension of the time series (compare Section 3.1). This difference in dimensions and variables requires encoding such uncertainty in a separate view. As a



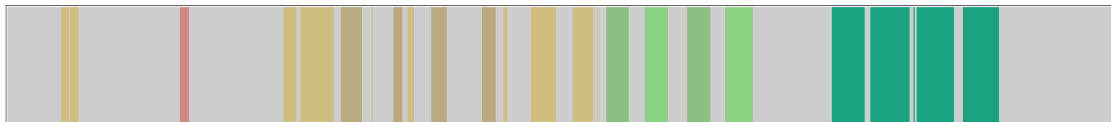
(a) Composite segmentation result label and uncertainty lineplot view.



(b) Heatmap view only encoding uncertainty.



(c) Gradient plot showing segmentation result labels and encoding uncertainty as saturation.



(d) Disambiguation plot showing segmentation result labels only if uncertainty is below a specified threshold (current threshold: 42% result uncertainty).

Figure 10.3: Visualization Designs for showing uncertainty in segmentation results.



(a) Area Uncertainty plot encoding value uncertainty.



(b) Uncertainty Heatmap view encoding value uncertainty.

Figure 10.4: Visualization Designs for showing value uncertainty.

result, the difference in uncertainty type should also be indicated by using a different visual encoding. To encode distribution-based uncertainty we use Area Uncertainty plot and Uncertainty Heatmap designs (compare Figure 10.4).

10.3 Study Design

The following subsections will give insights into how the visualization designs were generated, which hypotheses were formulated, and what study design was chosen, including a rundown of the questions in the study.

10.3.1 Data

The data used in the study has been generated using the segmentation pipeline from Bernard et al. [BDB⁺16]. The employed pipeline consisted of a sequence of four pre-processing steps, i.e., (1) missing value removal, (2) applying a moving average, (3) outlier treatment, and (4) data sampling, followed by a *k-means*-based segmentation and similarity-based labeling. The result uncertainties employed in the probability-based uncertainty designs were obtained from the *k-means*-based segmentation. The value uncertainties used for the distribution-based uncertainty designs were derived from the pre-processing steps of the segmentation pipeline. The input dataset that generated the segmented time series is a 120 seconds time series from the human MoCap database [MRC⁺07]. To create different results based on different parameter settings, I varied moving average window and sampling sizes to smooth out the data and remove notable patterns.

10.3.2 Hypotheses

Based on preceding findings, previous designs of segmented time series, and the defined design goals, I formulate the following hypotheses for evaluating uncertainty visualization designs for MVTS segmentation results:

H0 The Gradient Uncertainty plot **does not perform worse** than the Composite visualization showing segmentation results as colored bars and probability-based uncertainty as line plots.

H1 The Gradient Uncertainty plot **does not perform worse** than a Heatmap view showing only probability-based uncertainty for comparing assessing uncertainties of multiple segmented time series.

H2 The Gradient Uncertainty plot **is more effective** for assessing probability-based uncertainties of multiple segmented time series than a Threshold Uncertainty plot, **H2a** especially if vertical space is limited.

H3 The Uncertainty Heatmap view **does not perform worse** than the Area Uncertainty plot showing distribution-based uncertainty of a time series.

10.3.3 Participants and Questionnaire

In total, the study consisted of 111 participants (30 female), with the participants being undergraduate students, participating in a course in information design and visualization which implies basic knowledge about visual representations. The user study was provided in an online survey tool, SurveyJS [O⁺19], providing interactive visualizations for toggling the Gradient Uncertainty plot and adjusting threshold uncertainty for the Uncertainty Threshold plot. The study was designed as a within-subject study, meaning every participant had to answer all questions for each visualization design. To eliminate learning effects, all participants were distributed into groups, with each group receiving differently permuted sequences of questions (e.g., group A: [design a, design b, design c, design d], group B: [design d, design a, design b, design c], etc.). The participants

8 . Out of the highlighted areas (red frames), which is the most certain?

- Area A
 Area B

You can toggle between the regular result view and the uncertainty transparency view.

Show Uncertainty



You have spent 30 sec in total.

Next

Figure 10.5: Screenshot of a survey question evaluating the Gradient Uncertainty plot (Frame B has been cropped for improved readability).

27 . Out of the highlighted areas (red frames), which area has the least overall uncertainty?

- Area A
 Area B



Figure 10.6: Screenshot of a survey question evaluating the Area Uncertainty plot (Frame B has been cropped for improved readability).

received an overall introduction at the start of the study and another introduction to each of the visualization designs, which described how to interpret the visualization.

Task 1 - Comparison of Probability-based Uncertainty. The first six questions were designed to evaluate the Comparison task for probability-based uncertainty, which resulted in a total of 666 answers for each of the four visualization designs. Each of the questions showed one or more segmentation results with marked areas (compare Figure 10.5 – Questions 1 and 2 contained 2 marked areas, Questions 3 to 6 contained 3 marked areas), where the participant had to determine the most certain area (Questions 1 to 5), or sort from highest to lowest uncertainty (Question 6).

Task 2 - Comparison of Distribution-based Uncertainty. Another three questions were created to evaluate the Comparison task for distribution-based uncertainty, resulting in 333 answers for each of the two visualization designs. Each of the questions showed one or more segmentation results with marked areas (compare Figure 10.6 – 2 marked areas), where the participant had to determine the area with the least overall

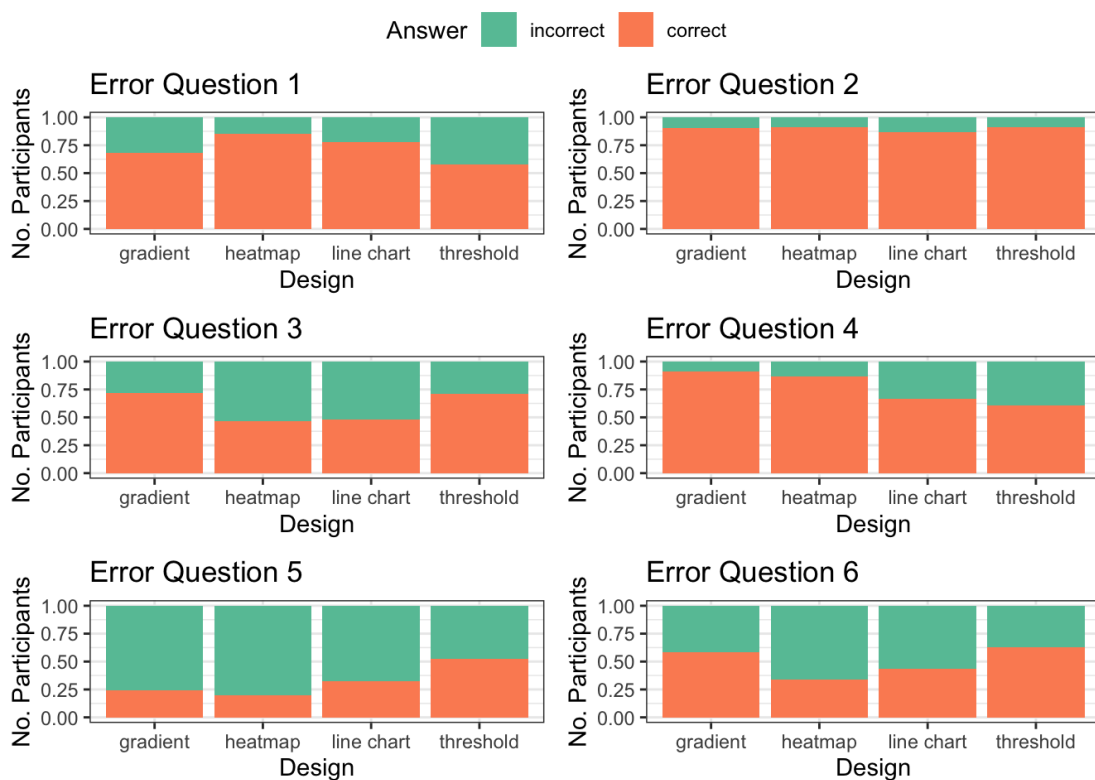


Figure 10.7: Participants' error rates for each question.

uncertainty.

10.4 Results

From the 111 participants of the study, we received 111 sequences of results, missing answers (14) were rated as wrong, instead of omitting the participant from the test (complete block design requires equal answers for each group). For each design the tested variables were participants' error (right or wrong answer) and completion times (in seconds).

10.4.1 Statistical Tests

After initial data pre-processing (converting from `json` to `csv` format, and determining correctness of the answers), the results were loaded into the R statistical computing environment [R C19]. To test for significance (compare Section 10.3.2, **H2**), we computed a Friedman rank sum test [CI81] for the test error and completion times in an *unreplicated complete block design*, i.e., the design has two variables, one *group* variable, and one *block* variable: The groups were the various visualization designs and the block was each

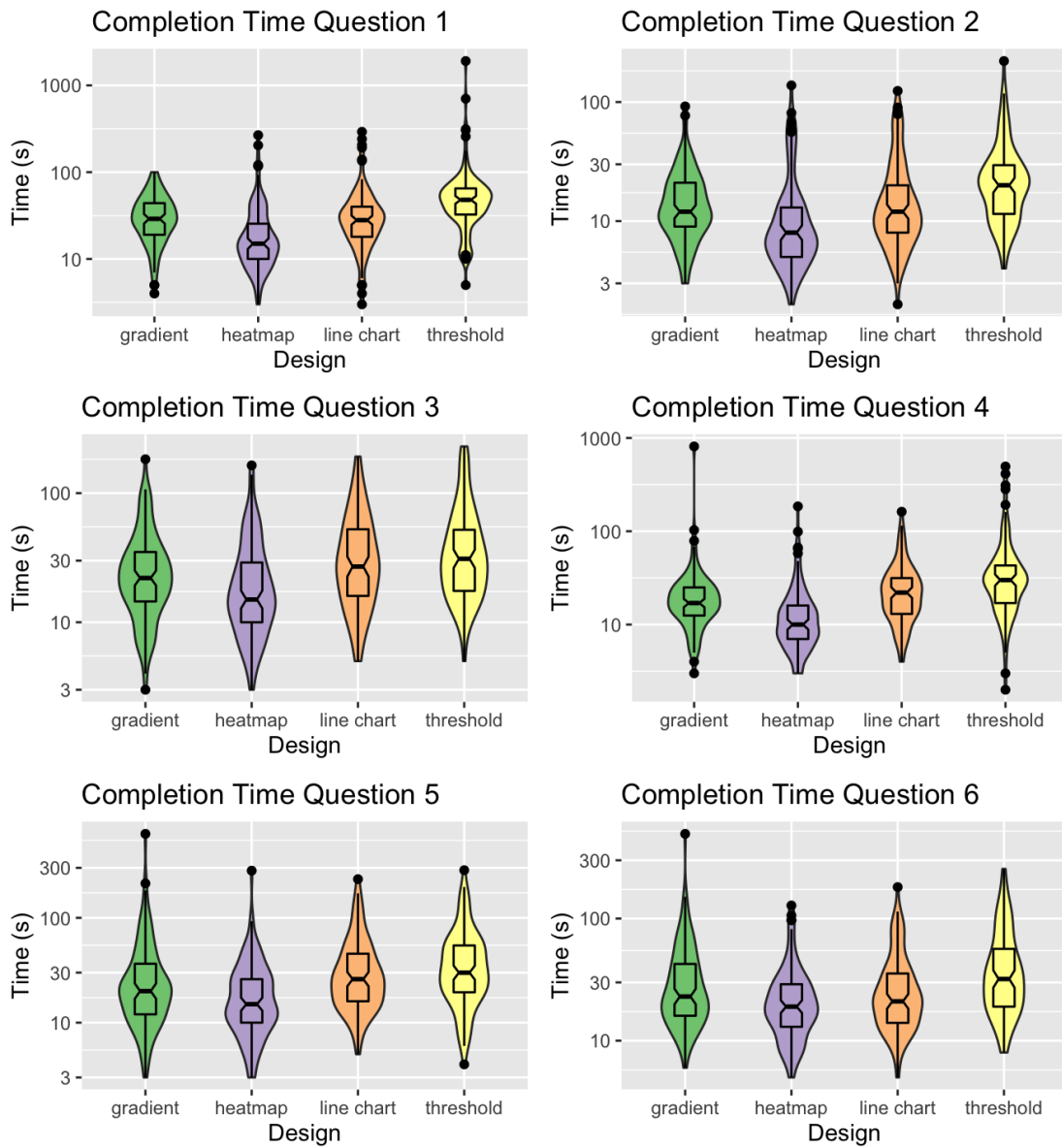


Figure 10.8: Participants' completion times for each question.

participant of the study. The significance was tested for Questions 1-6 combined, to test for significance of **H2**, and for Questions 4-5 combined, to test for significance of **H2a**, i.e., when vertical space is limited. Since the Friedman test result was significant for both the error and the completion times, we ran a post-hoc Nemenyi all-pairs comparisons test [Nem63] to determine design pairs that would be significantly different. The post-hoc Nemenyi test showed that no pairs were significantly different for error, but all completion time results were significantly different for questions 1-6, as well as questions 4-5 (compare Figure 10.8).

To test the remaining hypotheses (compare Section 10.3.2, **H0**, **H1**, and H3), we tested the results for equivalence and non-inferiority [WN11, Wel10], with a significance level $\alpha = 0.05$, and type II error probability $\beta = 0.05$. To achieve results across questions, the tests were run on the combined results (Questions 1-6 for **H0**, **H1**, and Questions 7-9 for H3), testing for non-inferiority pairs: **H0** – Gradient Uncertainty plot and Composite view, **H1** – Gradient Uncertainty plot and Uncertainty Heatmap, and H3 – Uncertainty Heatmap and Area Uncertainty plot. The non-inferiority tests confirmed **H0**, **H1**, and H3, and, furthermore, showed that the Gradient Uncertainty plot **error rate is lower** compared to the Composite view (**H0**), and that the Uncertainty Heatmap and Area Uncertainty plot **error rates are equivalent**.

Questions	Test Variable	Friedman p-value
1-6	Error	0.00023
1-6	Completion Time	<0.0001
4-5	Error	0.1705
4-5	Completion Time	<0.0001

Table 10.1: Friedman Test results for comparison of probability-based uncertainty.

10.4.2 Problems

After extensively exploring and evaluating the results, I found some problems and notable observations about both the results as well as the overall study and question design.

Inconsistent Question Difficulty. Two questions proved to be more difficult to solve for participants, questions 1 and 5. The error rates were different from the rest and increased the overall error rate for all questions. But surprisingly, these two questions showed that for difficult cases, the Gradient Uncertainty plot seems to perform worse. Compared to similar questions, error rates and completion times were higher.

- **question 1:** $\mu_{error} = 27.70\%$, $\tilde{t}_{completion} = 29s$, question 2: $\mu_{error} = 10.36\%$, $\tilde{t}_{completion} = 12s$
- question 4: $\mu_{error} = 23.87\%$, $\tilde{t}_{completion} = 18s$, **question 5:** $\mu_{error} = 67.79\%$, $\tilde{t}_{completion} = 23s$

Questions 1-6 – Error (p-values)			
	<i>Gradient</i>	<i>Heatmap</i>	<i>Composite</i>
<i>Heatmap</i>	0.224		
<i>Composite</i>	0.082	0.966	
<i>Threshold</i>	0.974	0.446	0.206
Questions 1-6 – Completion Time (p-values)			
	<i>Gradient</i>	<i>Heatmap</i>	<i>Composite</i>
<i>Heatmap</i>	<0.0001		
<i>Composite</i>	0.04	<0.0001	
<i>Threshold</i>	<0.0001	<0.0001	<0.0001
Questions 4-5 – Completion Time (p-values)			
	<i>Gradient</i>	<i>Heatmap</i>	<i>Composite</i>
<i>Heatmap</i>	<0.0001		
<i>Composite</i>	0.0085	<0.0001	
<i>Threshold</i>	<0.0001	<0.0001	0.0035

Table 10.2: Post-hoc Nemenyi all pairs comparison test.

Due to the low number of overall questions, it could be that the increased error rates of these two questions skewed the test results for all questions. For question 5, the Uncertainty Threshold plot outperformed any other design (p-value – *Threshold-Gradient*: 0.007, *Threshold-Heatmap*: 0.001, *Threshold-Composite*: 0.1001). This shows that specifically when uncertainty values are extremely similar, analysts could benefit using an interactive Uncertainty Threshold plot.

Insufficient Testing of Hypothesis H2a. I could not determine a significant difference in error rates w.r.t. **H2a**, testing if the Uncertainty Gradient plot performs better with limited vertical space available for visualizing segmentation results. In combination with the problem of question difficulties, only two questions were included in the study design that would evaluate this hypothesis. Question 4 showed significantly better error rate for the Gradient Uncertainty plot. But in question 5 the Threshold Uncertainty plot showed better results, which I attribute to the difficulty of the question, rather than the issue of limited vertical space (compare Figure 10.9). With the evaluated questions, this study cannot answer **H2a** because more diverse evaluation is necessary.

However, it had also the overall worst completion times because participants had to adjust threshold values in order to solve the questions.

10.4.3 Outcome

H0 – Comparing Gradient Uncertainty vs. Composite visualization. Hypothesis 0 **can be confirmed**. The non-equivalence test even showed superiority of the Gradient Uncertainty plot for error rate (p-value lower bound: 0.340, p-value upper

bound: <0.001), and equality for completion time (p-value lower bound: <0.001 , p-value upper bound: <0.001). The superiority of the Gradient Uncertainty plot is surprising, because the Composite view encodes uncertainty in location information, which supposedly outperforms the transparency encoding for quantitative data (according to Card et al. [CMS99]).

H1 – Comparing Gradient Uncertainty vs. Uncertainty Heatmap. Hypothesis 1 **can be confirmed**, the Gradient Uncertainty plot is superior over the Uncertainty Heatmap for error rate (p-value lower bound: 0.151, p-value upper bound: <0.001), however the Uncertainty Heatmap is superior to the Gradient uncertainty plot for completion time (p-value lower bound: <0.001 , p-value upper bound: 0.061). Again, the superiority of the Gradient Uncertainty plot is surprising, because uncertainty is encoded in the same visual channel, but the Uncertainty Heatmap does not encode additional label information as color, which should allow participants to assess uncertainty more accurately.

H2 – Gradient vs. Threshold Uncertainty performance. Hypothesis 2 **can not be confirmed for error rates**, there is no significant difference between both designs, according to the Post-hoc Nemenyi test ($p=0.974$). However, it **can be confirmed for completion time** ($p<0.0001$), but this is due to the fact that completion times with the Threshold Uncertainty plot were the highest overall.

H2a – Limited vertical space. In use cases with limited vertical space, the hypothesis can **neither be confirmed nor denied**. As discussed in the Problems Section, more evaluation is required to validate this hypothesis. In detail, question 4 revealed a significant difference: Gradient Uncertainty plot error rates were better than for the Threshold Uncertainty plot, which indicates that this could be confirmed in future studies.

H3 – Distribution-based Uncertainty Heatmap performance. Hypothesis 3 **can be confirmed**. The non-equivalence evaluation tested equivalence of both visualization designs for both error rate (p-value lower bound: <0.001 , p-value upper bound: <0.001) and completion times (p-value lower bound: <0.001 , p-value upper bound: <0.001).

10.5 Discussion & Lessons Learned

I will now discuss the hypothesis outcomes and other implications that I discovered from exploring the study results. In general, the study design evaluated performance of uncertainty comparison using Gradient and Threshold Uncertainty plots. I found the following results: (i) The hypotheses prove that segmentation results enhanced with probability-based uncertainty do not perform worse than designs only encoding uncertainty, which can be confirmed (**H0, H1**). (ii) However, the results did not confirm that views with limited vertical space or showing a large number of segmentation results do not benefit from using the Gradient Uncertainty plots (**H2, H2a**). (iii) There are no disadvantages using Uncertainty Heatmap for visualizing distribution-based uncertainty over Area Uncertainty plots (**H3**).

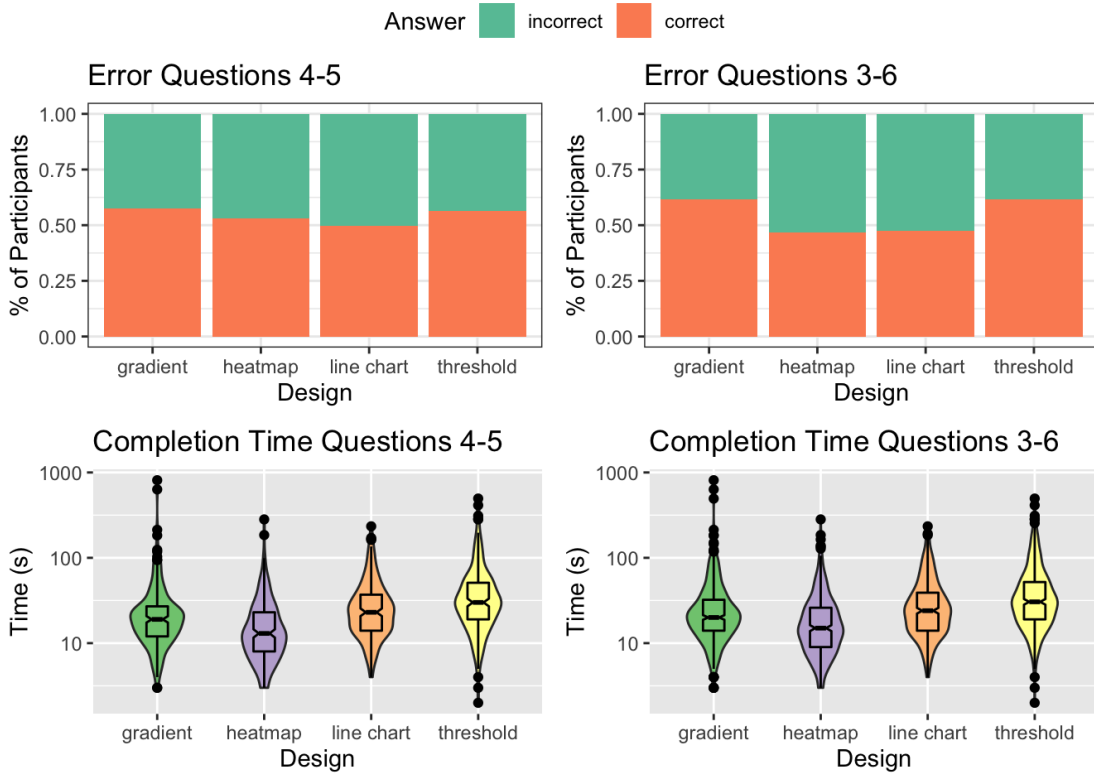


Figure 10.9: Participants’ error rates, aggregated for questions 4 and 5 (limited vertical space available) and questions 3, 4, 5, and 6 (vertical comparison examples).

H0 showed that the Gradient Uncertainty plot is superior to the Composite visualization, which implies that uncertainty comparison in segmentation results is more difficult using the Composite visualization design. One reason for this is the task of comparing uncertainties distributed vertically. Questions 3 to 6 contained examples where comparison had to be made across segmentation results, so participants had to perform comparison across vertical space (compare Figure 10.9). Tests for these questions showed significant differences in error rates (Gradient Uncertainty- Uncertainty Heatmap, Gradient Uncertainty- Composite, Threshold Uncertainty- Uncertainty Heatmap, Threshold Uncertainty- Composite), with the Gradient Uncertainty and Threshold Uncertainty plots outperforming the remaining two designs. This implies that when showing a large number of segmentation results, vertical comparison must also be done across results, and analysts would not benefit from views showing only uncertainty (e.g., Uncertainty Heatmap or Composite visualization). Task completion times are highest for Threshold Uncertainty plots, so for vertical comparison of uncertainties in large sets of segmentation results, Gradient Uncertainty plots are recommended.

Distribution-based uncertainty can vary in dimensionality and aggregation, with the finding that Area Uncertainty and Uncertainty Heatmap can be used interchangeably.

This extends designers' possibilities to more appropriately visualize this type of uncertainty. For example, Area Uncertainty plots can be used for stacking uncertainties if the overall level of uncertainty should be analyzed. Employing distribution-based uncertainty designs in detail views allows for comparison between

The hypotheses I tested showed that composite visualizations using both disambiguation and gradient uncertainty encodings work equally well as – and in some cases better than – simple visualizations only encoding uncertainty. Designers can integrate these designs into interactive views to allow analysts perform uncertainty-aware analysis and exploration without increasing error rate and only slightly increasing completion time. However, the study results also showed that for difficult scenarios with barely distinguishable uncertainties, participants had problems to determine the areas with the lowest/highest uncertainty, with the exception for the Threshold Uncertainty design. As a result, it is recommendable to employ different visualization designs that can be toggled, to support swift exploration using Gradient Uncertainty plots, and allow analysts to switch to Threshold Uncertainty plots to determine small differences and determine the most appropriate result. The difficulties in resolving barely distinguishable uncertainties indicates that mentally aggregating uncertainty values is challenging and should be supported with interactive techniques, for example, employing brushing techniques to show the aggregated uncertainty on-demand.

10.5.1 Lessons Learned

Reflecting on the user study design methodology, it was already been discussed that some aspects were not adequately evaluated due to the limited number of questions testing **H2a**. However, in general the employed samples and examples have served the purpose of finding the most appropriate visual encoding for conducting a comparison task in a MVTTS segmentation result. Another drawback for the overall evaluation was the influence of difficult questions affecting the overall distribution of scores in an unpredicted way. The latter issue could have been prevented with a more thoroughly tested set of questions presented to the participants. Appropriate user study design should be accompanied by extensive testing, most ideally with users that have little to no experience with the tested topic.

Part IV

The Conclusion

Conclusions & Limitations

In this chapter, I summarize my contributions in the research areas of VA, DQ, provenance, and uncertainty and how they influenced the answers to my research questions. I also list my scientific dissemination in the form of publications, and give an outlook to possible follow-up research and future work directions.

11.1 Summary of Contributions

From the previously presented visualization techniques and VA approaches I summarize the following contributions from the proposed solutions in Part II and the evaluation results shown in Part III. The contributions involved the conceptual design of visualization techniques and VA approaches, the development of visual-interactive prototypes to support data quality assessment and uncertainty analysis. I employed various evaluation types to determine the appropriateness, effectiveness, and expressiveness of these designs and prototypes.

Visual-Interactive Creation and Customization of Metrics. MetricDoc presents a visual-interactive environment for assessing data quality. It provides customizable, reusable DQ metrics in combination with immediate visual feedback, featuring an overview visualization of employed metrics, and an error visualization that facilitates exploration and navigation of quality issues present in the data. During the iterative design evaluation of the MetricDoc environment (compare Chapter 4) experts reviewed the features to be powerful for customizing DQ metrics in tabular datasets. There was general agreement that MetricDoc is appropriate to interactively assess the data quality and identify dirty entries. DQ experts also stated that supporting the creation and customization of metrics with interactions and visual feedback allows discovering error patterns, data property highlighting, and other contextual insights. This work laid the foundation to validate Research Sub-Question 1.

Capturing and Visualizing Provenance from Data Wrangling. DQProv Explorer was developed to capture and visualize provenance from data wrangling operations. It combines this provenance with descriptive annotations, like DQ metrics and summary information, to enable users to comprehend how changes to the dataset affected DQ. To accomplish that, the prototype features a provenance graph of operations and the data stream, a Quality Flow View to analyze the development of quality over time, and a Issue Distribution View to assess the distribution of DQ errors across the dataset. DQProv Explorer helps analysts understand the development of quality throughout data wrangling tasks, which makes it unique in its ability to explore workflow and data provenance from data wrangling. In a case study, I demonstrated the practical application of the system and discussed implications for extending the set of available quality metrics for more general use. A user experience study on DQProv Explorer showed that participants were capable of successfully completing various tasks associated with understanding and tracing the development of DQ in a dataset over time. Most interactive features were well understood and subjectively well received by participants, even though some features were not deemed as necessary (i.e., the Issue Distribution View). The results from the evaluation gave insights into analysts' trust in the dataset based on the provided DQ information and the personal data wrangling and cleansing experience. This was helpful in finding answers to Research Sub-Question 3.

Quantifying Uncertainty from Time Series Pre-Processing. In both a concrete (rastering) scenario (compare Section 6.1) and a more generalized approach (compare Section 6.2), I identified important aspects towards deriving DQ and uncertainty measures from time series pre-processing. While for the concrete case of time series rastering, I defined measures for interactively assessing the appropriateness of the rastering parameters in a VA approach. However, these measures were not generalizable to other pre-processing algorithms and procedures. I presented a methodology to quantify uncertainty in a generic way that enables designers visualize uncertainty more appropriately for the analysis of pre-processing steps and entire pipelines, depending on the use case, available uncertainty, and the task pursued by the analyst. This methodology helped to partly address Research Sub-Question 2. The case study presented (see Section 7.3) various visualization designs that enhanced existing analysis views with uncertainty information to derive insights on the influence of uncertainty on downstream analysis.

Visualizing Uncertainty of Time Series Segmentation Results. These visual designs were translated and further developed to be used for analyzing uncertainty of large numbers of segmented time series. Different visualization designs were evaluated in a quantitative user study to determine if a Gradient Uncertainty plot is appropriate for encoding uncertainty complementing colored segmentation result view. I found that the Gradient Uncertainty design is superior to using dedicated uncertainty visualization designs and can be employed for segmentation result analysis. However, a threshold uncertainty visualization should be available to identify subtle differences in uncertainty values when comparing different segmentation results. These results allowed me to determine further aspects for answering Research Sub-Question 2.

11.2 Answering My Research Questions

After shortly summarizing the work I presented in this thesis and describing the main takeaways from the different results, I want to contextualize these results within my research questions. The presented techniques and prototypical implementations have been tested within a range of different validation scenarios, ranging from case studies and expert reviews to qualitative and quantitative user studies, from which I derive empirical answer to the following questions:

Sub-Question 1 Can DQ metrics be utilized in a data wrangling and cleansing application as **measures of quality for various types of data** to give a visual overview of the **overall** amount of issues as well as a **detailed information** about the errors in the dataset? And how can VA methods be utilized to support **identifying, understanding,** and **correcting quality issues**?

- The MetricDoc environment allows creation and customization of DQ metrics and using them for gaining both overview information of DQ in datasets, as well as the detailed inspection of errors in the dataset is useful. In expert reviews I found implications that the interactive DQ metrics customization allows the discovery of error patterns and gain insights into the properties of tabular data.
- By defining re-usable quality checks and adding them to DQ metrics in updated datasets, analysts can swiftly search for errors and correct them accordingly based on which checks identified the dirty entries.

Sub-Question 2 How can **uncertainty** be **quantified from data wrangling and cleansing** and how can it be visualized to assess the influence of the pre-processing steps on downstream analysis?

- To address how uncertainty can be quantified from data wrangling and cleansing, I first used a concrete approach for deriving uncertainty and DQ metrics from rastering univariate time series. Subsequently, I developed an uncertainty quantification methodology that allows developers to estimate the impact of a pre-processing algorithm on the overall uncertainty in a MVTS.
- The uncertainty quantification cube is applied in two approaches for (1) pre-processing MVTS [BHR⁺19] and (2) segmenting and labeling MVTS in a segmentation pipeline [BBB⁺18]. It substantiates the general applicability of this methodology to communicate uncertainty inevitably introduced into the processed MVTS and allows analysts to both assess the influence of individual operations on uncertainty and perform uncertainty-aware analysis of the processed time series.
- To show how different dimensions of uncertainty could be communicated most appropriately to analysts, I conducted a quantitative study to evaluate various uncertainty design alternatives.

Sub-Question 3 What kind of **DQ information** can be **stored as data provenance** and used by analysts to **comprehend the history of data wrangling and cleansing steps** and assess the qualitative condition of the dataset to judge the data's usability?

- The Data Quality Provenance Explorer was developed to store DQ metrics and data summary information as provenance during data wrangling. It continuously captures and visualizes provenance to allow users explore the provenance graph of operations, the development of quality over time for available exploration paths.
- A user experience study showed implications that DQ metrics are a valid form of meta-information for allowing analysts understand the development of quality. Study participants were successfully able to complete tasks associated with comprehending the quality of a dataset.
- Addressing the question if data is usable: If participants' experience with data quality assessment was low, users were more willing to accept DQ overview visualizations as a form for validating the usability of a dataset. But with increasing experience, participants demanded more comprehensive and detailed methods for assessing data quality, and were not satisfied with one dedicated overview visualization.

The answers from my sub-questions can be summarized to provide a comprehensive answer for our main question:

Main Question: Which VA methods can be found as appropriate to explore and identify DQ issues in time-oriented data leveraging metrics, provenance, and uncertainty?

- By employing custom DQ metrics in tabular datasets and quantifying uncertainty from pre-processing operations and capturing them as provenance during data wrangling and cleansing, we can employ visualization techniques and VA systems that provide both overview of the DQ issues in a data set, as well as allow the analyst to further investigate the cause of errors using exploration of detail visualizations.
- Ultimately, analysts' perceived usability of a dataset depends on user experience, and the comprehensiveness of the employed DQ metrics depends on analysts' domain expertise and experience with data quality assessment. However, VA methods facilitate navigating detected quality issues and the employed error detection and quantification methods (DQ metrics and uncertainty quantification).
- In the domain of time series analysis, I investigated if uncertainty is beneficial to determine the impact of individual pre-processing operations on data. In particular, by employing interactive uncertainty-aware analysis and integrating visualizations showing various types of uncertainty, analysts can make more informed decisions to apply appropriate pre-processing operations in a processing and segmentation workflow.

11.3 Publications and Dissemination

The main scientific results were contributed to the field of computer science, and more specifically in visualization and VA research. However, research in data quality was first fostered through applied research in collaboration with companies processing time series data. These ventures resulted in interdisciplinary research in the fields of data quality and provenance. In another basic research project on time series pre-processing and

segmenting, further investigation was done into uncertainty quantification and analysis, more specifically the analysis of parameter influence in machine learning processes.

11.3.1 Data Quality and Provenance

- Christian Bors, Theresia Gschwandtner, Silvia Miksch, Johannes Gärtner, “QualityTrails: Data Quality Provenance as a Basis for Sensemaking”, *Proceedings of the IEEE VIS Workshop on Provenance for Sensemaking*, pp. 1–2, 2014.
- Christian Bors, Theresia Gschwandtner, Silvia Miksch, “QualityFlow: Provenance Generation from Data Quality”, *Proceedings of the Eurographics Conference on Visualization (EuroVis) - Posters 2015*, pp. 3, 2015.
- Christian Bors, Theresia Gschwandtner, Simone Kriglstein, Silvia Miksch, Margit Pohl, “Visual Interactive Creation, Customization, and Analysis of Data Quality Metrics”, *Journal of Data and Information Quality (JDIQ)*, vol. 10, pp. 3:1–3:26, 2018.
- Christian Bors, Simon Attfield, Leilani Battle, Michelle Dowling, Alex Endert, Steffen Koch, Olga A. Kulyk, Robert S. Laramee, Melanie Tory, John Wenskovitch, “A Novel Approach to Task Abstraction to Make Better Sense of Provenance Data”, *Provenance and Logging for Sense Making (Dagstuhl Seminar 18462)*. *Dagstuhl Reports*, 8(11):35–62, 2019. *Not peer reviewed*.
- Christian Bors, Theresia Gschwandtner, Silvia Miksch, “Capturing and Visualizing Provenance from Data Wrangling”, *Computer Graphics & Applications*, vol. 36-6, pp. 61–75, 2019.
- Christian Bors, John Wenskovitch, Michelle Dowling, Simon Attfield, Leilani Battle, Alex Endert, Olga Kulyk, Robert S. Laramee, “A Provenance Task Abstraction Framework”, *Computer Graphics & Applications*, vol. 36-6, pp. 46–60, 2019.

11.3.2 Uncertainty in Time Series Processing and Analysis

- Christian Bors, Markus Bögl, Theresia Gschwandtner, Silvia Miksch, “Visual Support for Rastering of Unequally Spaced Time Series”, *Data Science, Statistics & Visualisation Conference (DSSV)*, p. 1, 2017.
- Christian Bors, Markus Bögl, Theresia Gschwandtner, Silvia Miksch, “Visual Support for Rastering of Unequally Spaced Time Series”, *10th International Symposium on Visual Information Communication and Interaction (VINCI)*, pp. 53-57, 2017. *Best Short Paper Award*.

- Jürgen Bernard, Christian Bors, Markus Bögl, Christian Eichner, Theresia Gschwandtner, Silvia Miksch, Heidrun Schumann, Jörn Kohlhammer, “Combining the Automated Segmentation and Visual Analysis of Multivariate Time Series”, *EuroVis Workshop on Visual Analytics (EuroVA) 2018*, pp. 49–53, 2018.
- Markus Bögl, Christian Bors, Theresia Gschwandtner, Silvia Miksch, “Categorizing Uncertainties in the Process of Segmenting and Labeling Time Series Data”, *Proceedings of the Eurographics Conference on Visualization (EuroVis) - Posters 2018*, pp. 45-47, 2018.
- Christian Bors, Theresia Gschwandtner, Silvia Miksch, “Visually Exploring Data Provenance and Quality of Open Data”, *Proceedings of the Eurographics Conference on Visualization (EuroVis 2018) - Posters*, pp. 9–11, 2018.
- Markus Bögl, Christian Bors, Theresia Gschwandtner, Silvia Miksch, “Uncertainty types in segmenting and labeling time series data”, *Data Science, Statistics & Visualisation Conference (DSSV)*, p. 1, 2018.
- Christian Bors, Markus Bögl, Theresia Gschwandtner, Jürgen Bernard, Silvia Miksch, “Quantifying Uncertainty in Time Series Data Processing”, *Vis In Practice Mini-Symposium on Visualizing Uncertainty – VIS 2018*, p. 1, 2018.
- Jürgen Bernard, Marco Hutter, Heiko Reinemuth, Hendrik Pfeifer, Christian Bors, Jörn Kohlhammer, "Visual-Interactive Preprocessing of Multivariate Time Series Data", *Computer Graphics Forum*, Volume 38, Number 3, pp. 401–412, 2019.
- Christian Bors, Jürgen Bernard, Markus Bögl, Theresia Gschwandtner, Jörn Kohlhammer, Silvia Miksch, "Quantifying Uncertainty in Multivariate Time Series Pre-Processing", *EuroVis Workshop on Visual Analytics (EuroVA)*, pp. 31-35, 2019.

11.4 Future Directions

Based on the findings, results, but also shortcomings in the presented works, I identify the following future directions that were beyond the scope of this thesis but could significantly advance research for visual-interactive data quality assessment and uncertainty analysis from data pre-processing.

DQ metric complexity. I shortly discussed in the results of the expert reviews in MetricDoc that analysts’ expected application scenarios of DQ metrics ranged from simplistic evaluation schemes to very complex validation scenarios containing multiple quality checks. Modularity would allow this, however, constructing such complex validation functions requires appropriate coding support. For example, employing visual scripting to create checks and metrics could significantly empower analysts developing new metrics. Another aspect is employing statistical and machine learning techniques to recommend and create more expressive metrics automatically, and using external source

validation, like linked or semantic data. Improving the expressiveness of DQ metrics could also be beneficial for analysts using DQProv Explorer.

Scalability of DQProv Explorer. The provenance graph used in the case study (compare Section 7.2) and the user experience study of the DQProv Explorer was average in size. Both the Quality Flow View and Provenance Graph View showed no issues w.r.t. scalability, also with provenance graph twice the size. However, either very large provenance graphs, or a graphs with excessively long wrangling sequences could render both views ineffective due to overplotting. To address this, merging similar branches and sequential patterns in the provenance graph could resolve scalability issues. Additionally, the currently statically drawn Provenance Graph View could be replaced by a dynamic and interactive graph structure that allows panning, zooming, and disabling/hiding uninteresting branches.

Combining uncertainty and DQ metrics. The uncertainty quantification and visualization techniques presented in this thesis were specifically developed for time series analysis. Analysts and developers alike could benefit from a general framework for quantifying uncertainty from pre-processing operations. Such a general framework allows more consistent integration of uncertainty in visualization design and makes the influence of pre-processing on downstream analysis more explicit to the user.

Support collaboration With a continued increase of the amount of data we generate and the necessity of data pre-processing and analysis to be comprehensive and comprehensible, the aspect of collaboration and sharing work between analysts is important. Provenance can facilitate collaboration by allowing analysts to observe and review prior work done by colleagues. The methods and VA solutions presented in this thesis could be extended by collaborative features to allow analysts annotate the development of DQ and uncertainty over time to explicitly introduce *insight provenance* into the provenance graph, and make the changes they applied better understandable.

Part V
Appendix

DQProv Explorer – Qualitative User Study

12.1 Evaluation Structure

12.1.1 Introduction

- 5-10 Minutes of introduction into the field of Data Wrangling
- Introduction into OpenRefine transformations and filters
- Introduction into the Explorer prototype
 - Overview of quality metrics used within the prototype (completeness, validity, numeric plausibility)
 - Used encodings
 - General functions of the three different components
 - * Quality Flow View (QF)
 - * Issue Distribution View (ID)
 - * Provenance Graph View (PG)
 - * Comparison View/Mode (CV)
 - Interactions available in the components (I)
- Introduction into the used data set, with short overview of particular data columns.

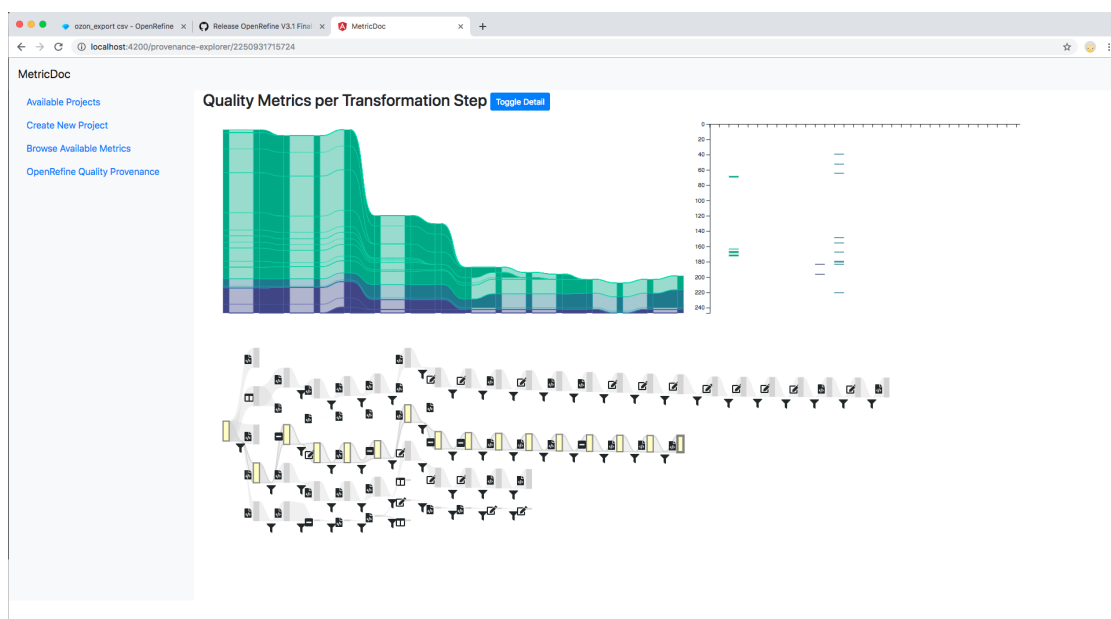


Figure 12.1: Overview of the *DQProv Explorer*. In the Provenance Graph View, branch 2 of 4 major branches is selected for analysis of quality development over time (in the quality flow view in the top left, and detailed analysis of remaining issues in the Issue Distribution View (top right).

12.1.2 Dataset

The dataset used during the user study is a slightly modified version (purposefully removed single cells) of the *TopGear* dataset obtained from the *R*-package *robustHD* [Alf08]¹. The data quality metrics employed alongside the dataset were deliberately chosen to be simplistic, so that participants need not to require further information on specific quality checks and validation schemata. The dataset exhibited quality issues in terms of validity (invalid data types), completeness (missing values), and plausibility (implausibly high/low values).

12.1.3 Tasks

The following describes the tasks given to the study participants, to see if the design of the prototype allowed conducting the tasks of confirming quality changes, discriminating between different changes in quality, validating if a dataset is usable in its current state, and understanding the sequence of transformations conducted by a different user.

T_{act} & T_{pres} - Look at the first state of the dataset and identify the column with the most issues (Column ‘*weight*’). Now look at the end node of one transformation branch and determine how quality evolved for this column. You can see multiple transformation

¹<https://www.rdocumentation.org/packages/robustHD/versions/0.5.1>

branches: How different are the two branch end nodes in terms of quality, do similar issues remain? Can you find out what transformation/operation impacted the quality of this column the most?

T_{meta} - If only the dataset of the second branch was available for analysis, what columns would you use for analysis. If you look at the three different branches and compare remaining quality issues, which one would you choose for analysis, and for what type of analysis?

T_{rec} & T_{rep} - How did a sequence of actions influence the data? Going back to the Weight column, which of the branches would you use for analysis?

T_{coll} & T_{meta} - Can you determine the user's objective in the sequence of transformations shown in the branch at the bottom of the provenance graph?

12.1.4 Participant Expertise

Participants were asked about profession, and self-assessment of experiences in the fields of:

Data Wrangling, Data Profiling, or Data Cleansing

If Experienced, on what type of data

Information Visualization, Visual Analytics

12.2 Summarized Results

We have summarized the feedback from participants by views and interactions of DQProv Explorer. That way the usefulness of each part of the system could be assessed on its own, and how the interactions combined them. We also note how many participants used which view for which task.

12.2.1 Quality Flow View

This view was received well, 5 of 6 participants noted the usefulness for assessing the development of quality. The view was used in all tasks by all participants.

Two users initially had issues mapping the stacked bars to columns.

12.2.2 Provenance Graph View

The Provenance Graph View was used in all tasks by all users. But in terms of usability, the node size was deemed as not useful by 3 of 6 participants. 3 participants could not understand the encoding of data flow along the edges, hence they wondered why edges were seemingly bundled or disappeared (when the data was filtered to only very few values, the edge would become very thin). 1 participant noted to add a filter function to

highlight nodes that affected particular columns, and adding a delta function to more effectively find changes in row size of the data. Participants explored the graph in different ways: while 2 participants iteratively navigated each node of the graph, the remaining participants were only interested in the end nodes (*“I would like to have all end nodes highlighted”*).

12.2.3 Issue Distribution View

4 of 6 Participants questioned the usefulness of the Issue Distribution View, using them as part to solve tasks \mathbf{T}_{meta} (2 of 6), \mathbf{T}_{coll} (2 of 6), which *“takes up whitespace”*, and *“I haven’t used the detail view, and for the current selection it does not even give me useful information”*. We attribute these critical comments to the application scenario employed in our study design, and the assigned tasks not being specifically tailored to assessing error distribution in the dataset. 3 of 6 Participants criticized that the change of content in the difference view when switching to comparison mode is unclear and must be signaled accordingly.

12.2.4 Comparison Mode

The comparison mode was appreciated to compare branches, participants used them in tasks \mathbf{T}_{rec} & \mathbf{T}_{rep} (6 of 6) and \mathbf{T}_{coll} & \mathbf{T}_{meta} (3 of 6).

3 participants noted that the mirroring initially posed confusion. Even though the participants were explicitly instructed about the mirroring, 2 participants still mixed up the branches during detailed inspection. It was noted to signal the mirroring more clearly (the colored nodes were not sufficiently indicative), and one participant suggested to mirror the Provenance Flow View vertically to compare the selected data revisions.

12.2.5 Interactions

Other critical feedback could be traced back to limited interaction possibilities, and we determined that some approaches pursued by participants during task execution would have required a more extensive set of interactions, such as metric selection to brush nodes affecting the metric in the Provenance Graph View, provenance graph node filtering, or highlighting techniques.

Evaluation – Participant 1

Participant

- Gender: male.
- Profession: MA Student.
- Expertise:

- Data Wrangling, Data Profiling, or Data Cleansing: Yes, No, Yes. Advanced. Tools: LoD Refine² (OpenRefine³).
- Data: Multiple data source harmonization.
- Information Visualization: Entry level, data analysis plots and statistics plots.

Performance on T_{act} & T_{pres}

- Could find the column with maximum error in the quality flow visualization (QF). But only assessed validity metric as maximum, even though also a second metric (completeness) signaled issues in this column.
- By selecting the last node of the top branch (PG), and observing the flow of the metric developing over time (QF), he could find that one operation reduced quality.

Performance on T_{meta}

- Comparing the two top branches (PG) lead the participant to the conclusion that the top branch yielded more valid data, with the lower branch removing entries unnecessarily (I).

Performance on T_{rec} & T_{rep}

- Looking at the second branch (CV, PG, I), quality was improved, but he found that this corresponds to changes of other problems as well (QF) (this is due to rows being removed, affecting the ratio of errors across all metrics).
- He would not use the dataset due to these transformations affecting all rows (CV, PG, I) (*rows are being deleted*).

Performance on T_{coll} & T_{meta}

- The participant tried to focus on quality (QF) and try to comprehend what happened when multiple entries were edited but could not due to [*self-assessed*] missing info (I) (*[info is available on mouseover, but is limited that edit action was performed, but detailed information is missing]*).

Critical Feedback

- The participant could not see what impact an action had on the data (I), due to the edges (PG) not being clearly recognizable to him. The visual encoding of edge width corresponding to filtering rows of the dataset was not understood.
- More highlighting (I) was demanded, e.g., highlighting columns, searching for nodes that were changed in the provenance graph.
- The QF when comparing two branches should be scrollable (I).

²<https://sourceforge.net/projects/lodrefine/>

³<http://openrefine.org/>

- Raw data should be comparable on demand in the DV.
- differentiation in the CV between the two paths is unclear, a different linking should be employed to show the differences in metrics between the two end-nodes.

Positive Feedback

- The visualization of quality across data transformations (QF) was marked as very useful, especially when the scale of the dataset is larger.

Evaluation – Participant 2

Participant

- Gender: male.
- Profession: PhD Student.
- Expertise:
 - Data Wrangling, Data Profiling, or Data Cleansing: Yes, Yes, Yes. Expert, > 1 year. Tools: self developed tools.
 - Data: Text data, retrieval data.
 - Information Visualization: Expert, > 1 year.

Performance on T_{act} & T_{pres}

- Could successfully determine the column *Weight* (QF), but immediately noted that height is sub-optimal encoding for lack of quality – he would rather prefer height maps to high quality.
- He noted the necessity for mouse-over trial-and-error (I) for finding the column with the highest number of issues.
- He did not understand alignment of nodes (PG) and QF bars at first, so searched for nodes that affected column *Weight* individually (PG, I), and noted that it could be beneficial to highlight nodes that affect column *Weight* on demand (I), to have insight how this column changed across all branches.

Performance on T_{meta}

- Participant wants to see the changes rather than the overall quality development (PG).
- Noted that the diff view (ID) is not very helpful in detecting the differences between two views. Alternatively the overall number of rows and quality issues could give better way of determining a difference. Also a link into the data could help.
- By highlighting what nodes were affecting certain selected columns, exploration would be more easy, and to guide users towards relevant branches.

Performance on T_{rec} & T_{rep}

- He used the provenance graph nodes to determine how many rows remain in the dataset for the top two branches of the provenance graph, determined that the second contained less data and that overall quality was not significantly lower, so preferred branch number 1 (from top).
- Within this process he noted that he would like to see all branches' end-points highlighted.
- Also it was noted that the diff did not help enough, because both the overall number of entries *and* the quality are key measures for high quality in the dataset.
- Validation would require inspecting the data – wants a link back into the data state.

Performance on T_{coll} & T_{meta}

- Understood the transformation operations, in which a subset of the data was selected to conduct cleansing only on that data. But noted that the filter indication is not very expressive without the possibility to observe the content of the column.
- Single cell operations do not tell any information what happened – needs to be addressed to trace actions. - If overall error is increased, information without signaling the number of rows is rather ambiguous and needs to be determine in a separate step.
- The participant understood that the decision what transformation path to choose depends on the subsequent analysis, based on if high accuracy of the available data is favoured, or if more entries with imputed values are beneficial (e.g., for model building) [*it should be added that the used dataset provides data without significant outliers, but certain entries are incomplete*]

Observations

The user mainly utilized the quality flow visualization for determining changes in the data, and only used the provenance graph mouseover information if necessary.

Critical Feedback

- In the provenance graph, the filter analogy can be overlooked easily if filtering only yields a small number of entries. A delta of changes, or numeric values for total size and number of changes in the data (for each state of the dataset) would signal changes more effectively. - Mirroring the second quality flow visualization should be signaled.

Positive Feedback

- The prototype provides the ability to conduct collaborative cleansing by allowing users to see the branches that are created by different approaches.
- The quality flow visualization is very effective for signaling the overall quality.
- Linking of the prototype components is really smooth and helps with exploration.

Evaluation – Participant 3

Participant

- Gender: male.
- Profession: PhD Student.
- Expertise:
 - Data Wrangling, Data Profiling, or Data Cleansing: Yes, Yes, Yes. Entry to advanced level, < 1 year. Tools: scripting.
 - Data: Databases: relational/structured data.
 - Information Visualization: Expert, > 1 year.

Performance on T_{act} & T_{pres}

- Easily found weight column by using mouseover (QFV), could determine the operation responsible for the change. However the filter operation was not clearly understood at first.

Performance on T_{meta}

- Participant is iteratively navigating nodes in comparison view, and trying to understand alternate path column changes.
- Could determine that second path solved quality issues at the same steps, but in a different way. But determined that the second branch is more beneficial for solving problems.
- Not clear that dark colored paths signal a change in metric (suggestion to use a different texture).

Performance on T_{rec} & T_{rep}

- The participant noticed that branch two reduced the data size while branch one retained the data and concluded that selecting between those branches came down to preference.
- First branch more complete, second branch removed data.
- To decide the user wants to know more information on what changed in the first branch to reconstruct (missing information what was changed in the single cell operations).
- If the dataset is unknown, and operations are provided in detail: The user preferred retaining information, under the assumption of knowing that issues were solved. This requires trust in the dataset and the user conducting the wrangling process

Performance on T_{coll} & T_{meta}

- Participant could find out that cars running fossil fuels were removed and found that quality degraded.
- He concluded correctly by comparing two branches that the ratios of problems increased due to the removal of more correct data, and retaining dirty ones.

Observations

The user used iterative selection of the PG nodes to exactly retrace changes done to the dataset. Hence, the participant could understand the used wrangling workflow quite effectively. Subsequently, the participant saw the need for improving graph interactions, like a focus+context technique, or grouping operations. In contrast, the Issue Distribution View was not used at all. For him it was difficult to distinguish changes of quality in the quality flow visualization, color coding transitions like states added to this problem, could be addressed by employing a different coloring schema.

Critical Feedback

- differentiation in the comparison view between the two paths is unclear, a different linking should be employed to show the differences in metrics between the two end-nodes.
- Information on filters should be more intuitive and clear (range indicators, and condensing information)
- Connection to the dataset should allow more detailed analysis.
- Issue Distribution View adds to much white space, and does not resolve the question where the issues are, apart from position, but this is irrelevant for retrospective analysis.

Positive Feedback

- Adding signals to highlight nodes that have already been explored.
- The prototype allows for finding leaks and modifications more easily, if done in the tool.
- Use of icons for operations.
- Collaborative efforts can be explored.

Evaluation – Participant 4

Participant

- Gender: male.
- Profession: Post-doctoral researcher.
- Expertise:

- Data Wrangling, Data Profiling, or Data Cleansing: Yes, Yes, No. Expert, < 1 year. Tools: Alteryx⁴.
- Data: Text data, retrieval data.
- Information Visualization: Expert, > 1 year.

Performance on T_{act} & T_{pres}

- Participant could easily find column *Weight*. But noticed that it disappeared, by attempting to click the transformations.

Performance on T_{meta}

- Participant valued branches with less operations to accomplish similar quality, but still determined branch one to be the best quality dataset, he also interpreted the completeness metric as the most worrying.

Performance on T_{rec} & T_{rep}

- Inspection of individual changes in quality.
- Mostly focusing on filtering icons and mouse-over information, rather than filters and operations
- Participant preferred dropping columns (what's the least amount of columns to conduct an analysis on the entire dataset?)
- Trust in imputed values is only accepted if knowledge about who conducted the operations is available, otherwise dropping these entries is preferred.

Performance on T_{coll} & T_{meta}

- Participant noticed worse amount of errors based on the row removal

Critical Feedback

- Participant suggested the ability to filter for changes in specific columns, to find transformations more quickly (T1)
- Single cell operations require more information.
- Operations icons should be encoded by a glyph.
- Quality flow should also encode information about number of rows/entries in the dataset.

Positive Feedback

- Quality flow was appreciated, but the participant suggested vertical mirroring instead of horizontal.
- Using a different set of metrics for determining a dataset's appropriateness for machine

⁴www.alteryx.com

learning training.

- Usefulness is tied to the objective quality functions – the more expressive they are, the better the analysis can be.

Evaluation – Participant 5

Participant

- Gender: female.
- Profession: PhD student.
- Expertise:
 - Data Wrangling, Data Profiling, or Data Cleansing: Yes, Yes, No. Beginner.
 - Data: Scientific data, spatial data
 - Information Visualization: Expert, > 3 years.

Performance on T_{act} & T_{pres}

- Participant wanted to use click interaction (QF, I) to find column *Weight*. But after all found out to use on demand mouseover information to find the column.

- Participant struggled to find context information to determine the corresponding transformation, alignment could not help adequately.

Performance on T_{meta}

- The participant did value lower quality over availability of the data. Upon asking the metrics were seen as trustworthy by the user.

Performance on T_{rec} & T_{rep}

- Upon inspection (CV) the participant expressed that the columns are rather unclear to her. It did not help her to comprehend what happened in the data (the miles per gallon column exhibited excessive amounts of implausible values), but she did not associate the change with the transformation (PG, QF).

Performance on T_{coll} & T_{meta}

- Participant could not associate the changes in quality to the nodes/edges in the provenance graph (PG).

- She did not find out for what purpose the sequence of actions/operations was executed, hence a distinction between the branches was not achieved by her (CV).

- The inspection only led to single insights, that certain actions caused a decrease in quality issues.

Observations

The participant did not try to explore all different modes of interaction, and hence also did not leverage them to determine the source of changes in quality or compare the differences between two selected quality flows.

Critical Feedback

- Difficult to see where data are filtered, suggested to use different encoding, only show filter information on demand.
- The number of data in a data state is not clearly visible, and rarely comparable.
- Data to ink ratio low.
- Change metric representation to resemble columns – vertical scaling of the visualization.

Positive Feedback

- Quality encoding makes sense intuitively.
- Comparison view works well.

Evaluation – Participant 6

Participant

- Gender: female.
- Profession: PhD student.
- Expertise:
 - Data Wrangling, Data Profiling, or Data Cleansing: Yes, Yes, Yes. Advanced, > 1 year. Tools: Excel.
 - Data: tabular data, relational data.
 - Information Visualization: Expert, > 3 years.

Performance on T_{act} & T_{pres}

- Participant could easily find column *Weight*. Attempted to click the metric paths.
- She could find minimal changes in quality for the selected branch and could determine the operation responsible for the change, by iterating through all operations until she found the change in the column.

Performance on T_{meta}

- For comparison the participant disabled the detail view.
- Noted that differences were not significant (for the selected branches).
- Using the comparison mode, she decided for the branch with higher quality/lower amount of quality issues.
- Participant wanted to use node toggling to determine the changes in quality between two changes.

Performance on T_{rec} & T_{rep}

- Would prefer the dataset with higher quality, when confronted with the branch that removed rows, she preferred the other, valuing data size as well.
- Did not trust the dataset enough to decide on a branch, without ability to look into the raw data.

Performance on T_{coll} & T_{meta}

- Participant noticed that operations had different implications, which came down to the observation that she used an iterative approach towards understanding the wrangling/cleansing process.
- Could distinguish and understand differences in operation types and their impact on quality.

Critical Feedback

- Demanded more interaction and linking abilities, in particular quality flow to provenance graph.
- Path highlighting was not sufficient to link the branches to the flow views.
- Legend missing.
- Graph structure changes during exploration, makes navigating harder.
- Demanded column labeling.

Positive Feedback

- Liked use of whitespace
- Participant stressed the importance of the Issue Distribution View, and would only provide a toggle to remove "empty" columns.
- Liked the use of color that make the elements distinguishable.

DQProv Explorer – Usability Inspection

13.1 Evaluation Structure

The evaluation was split into two different studies, interviews and a focus group. Both groups got an introduction. The participants received paper prototypes and had to solve questions and tasks, as well as give feedback on the usability in the end.

13.1.1 Introduction

- Introduction on course of interview
- General questions (age, experience, etc.)

13.1.2 Goal

The evaluation had the following goals:

- Validate if design an used symbols (add, delete, merge icons) are understandable by participants.
- Discovery of possible improvements to the design.

The following questions were formulated to be used for determining tasks:

- Is the prototype intuitive?

- Are the visualizations for (1) quality metrics, (2) detail view, (3) column edit operations understandable?
- Can the individual steps be followed (e.g. change path, show detail view, ...)

13.1.3 Questions

The subjects received questions and tasks to solve with the paper prototypes on the topics:

- Quality metrics,
- Analysis of metric changes,
- Detail views,
- Alternate provenance graph paths,
- Column removal, creation, and merging, and
- Annotations

The investigator continuously guided the participants through the experiment, asking questions for different tasks and consecutive operations. Below some exemplary figures are shown, which were used as paper prototypes during the experiment, so participants could use pencils to add notes.

13.1.4 Feedback

The participants were asked to give feedback on the three most and least favorite design aspects, as well as general remarks.

- Less colors means the dataset is cleaner (also noted in focus group)
- Operations are shown prominently and pleasantly
- Clean, without aid lines
- Provenance graph also shows alternate paths
- Provenance graph is similar to git

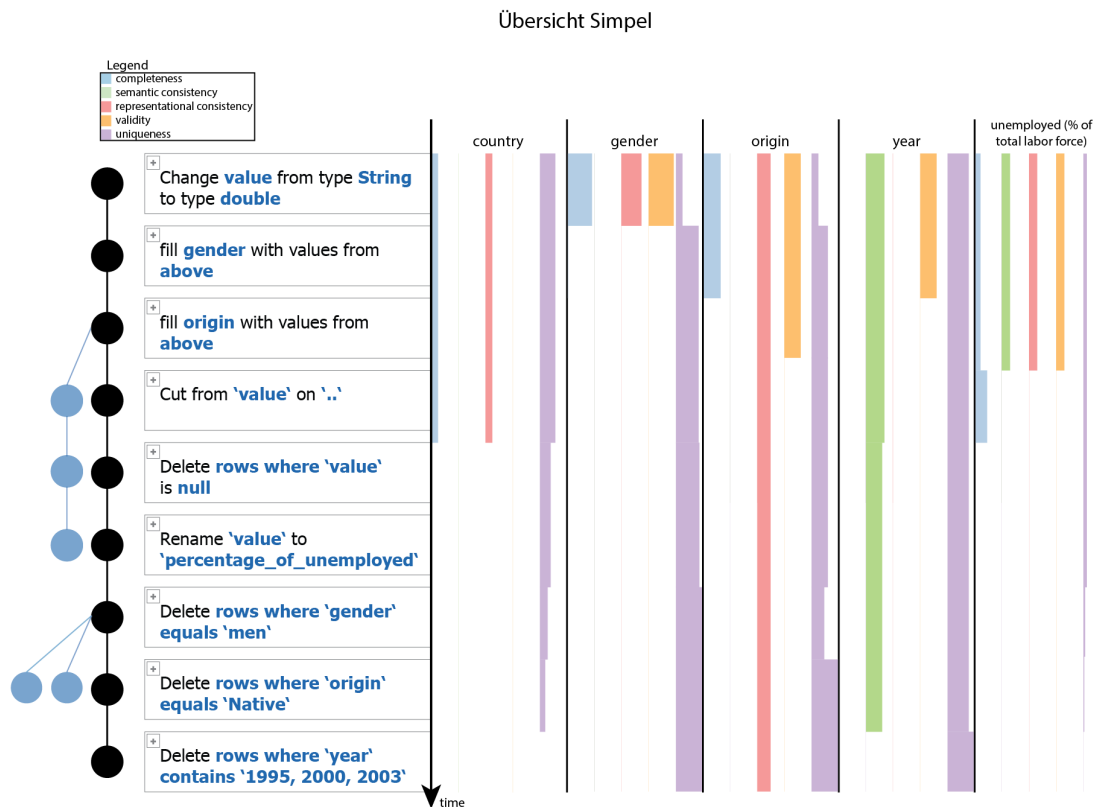


Figure 13.1: Overview of the paper prototype the participants had to solve questions and task with.

Interview

- Parameters are highlighted in operations
- Delete, insert, and merge visualizations
- Effects of operations can be seen in the bar chart visualization
- Difference of data in detail view
- The plus symbol that shows the ability to open a detail view
- the quality bars can be compared easily

Focus Group

- It can be seen how the dataset looked before and after the operation,
- The detail view is helpful to follow changes,

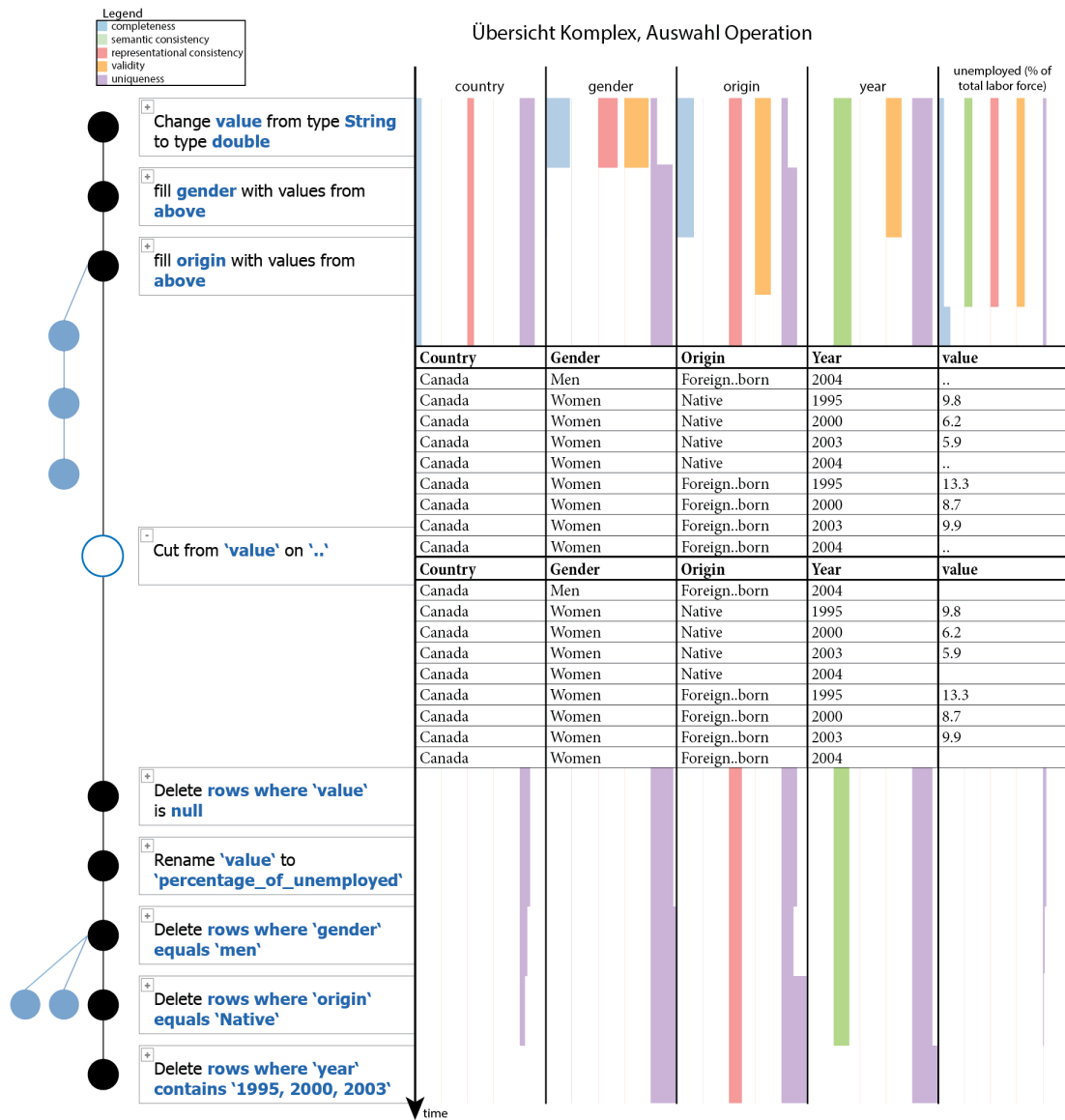


Figure 13.2: In the detail view the raw data is shown, with the data showing the state directly before and after an operation.

- Structure is well understandable,
- Overview of the changes,
- Effect of operations can be followed for the most parts.

Visualizing Uncertainty in Time Series Processing

14.1 Questions and Results per Question

14.1.1 Questions

Questions 1 to 6 are used for testing hypotheses H_0 , H_1 , and H_2 . Questions 7 are used for testing hypothesis H_3 .

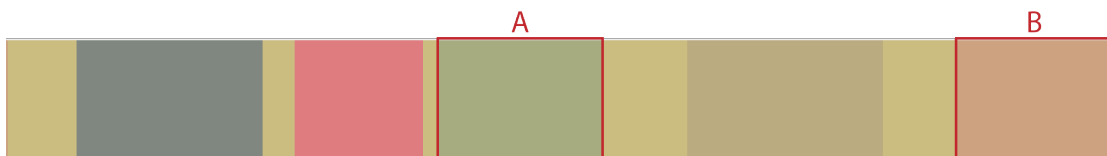


Figure 14.1: Question 1: Out of the highlighted areas (red frames), which is the most certain?

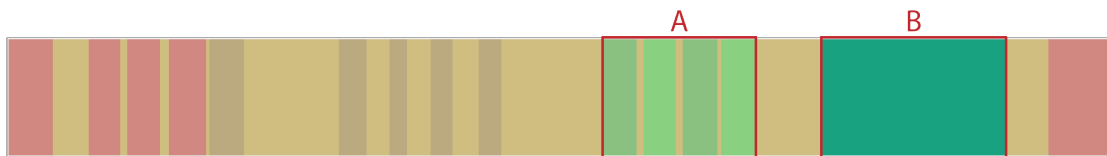


Figure 14.2: Question 2: Out of the highlighted segments (red frames), which is the most certain?

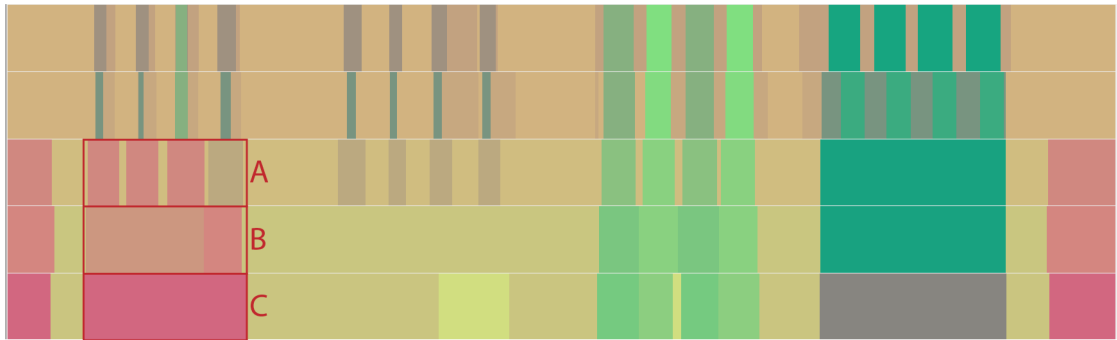


Figure 14.3: Question 3: Out of the highlighted areas (red frames), which is the most certain?



Figure 14.4: Question 4: Out of the highlighted segments (red frames), which is the most certain?

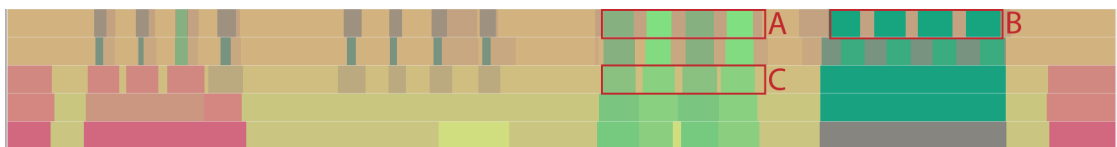


Figure 14.5: Question 5: Out of the highlighted areas (red frames), which is the most certain?

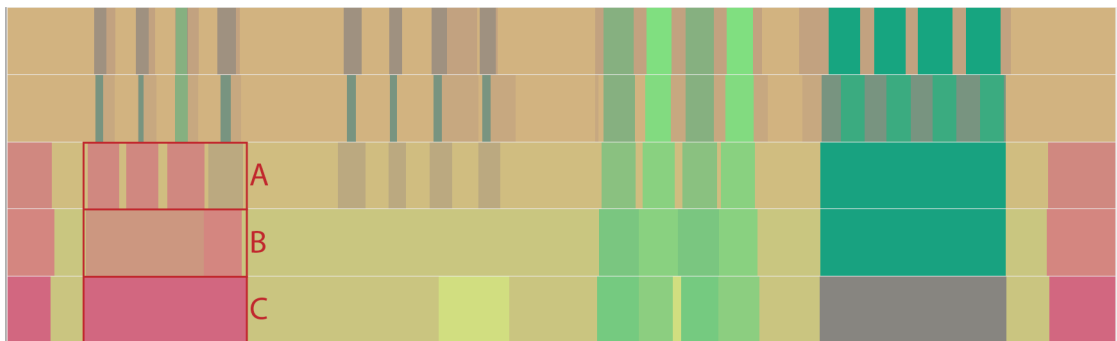


Figure 14.6: Question 6: Please sort the following highlighted Segments from Most Certain to Least Certain.



Figure 14.7: Question 7: Out of the highlighted areas (red frames), which has less uncertainty (Area Chart Variant)?



Figure 14.8: Question 8: Out of the highlighted areas (red frames), which has less uncertainty (Area Chart Variant)?



Figure 14.9: Question 9: Out of the highlighted areas (red frames), which area has the least overall uncertainty (Area Chart Variant)?

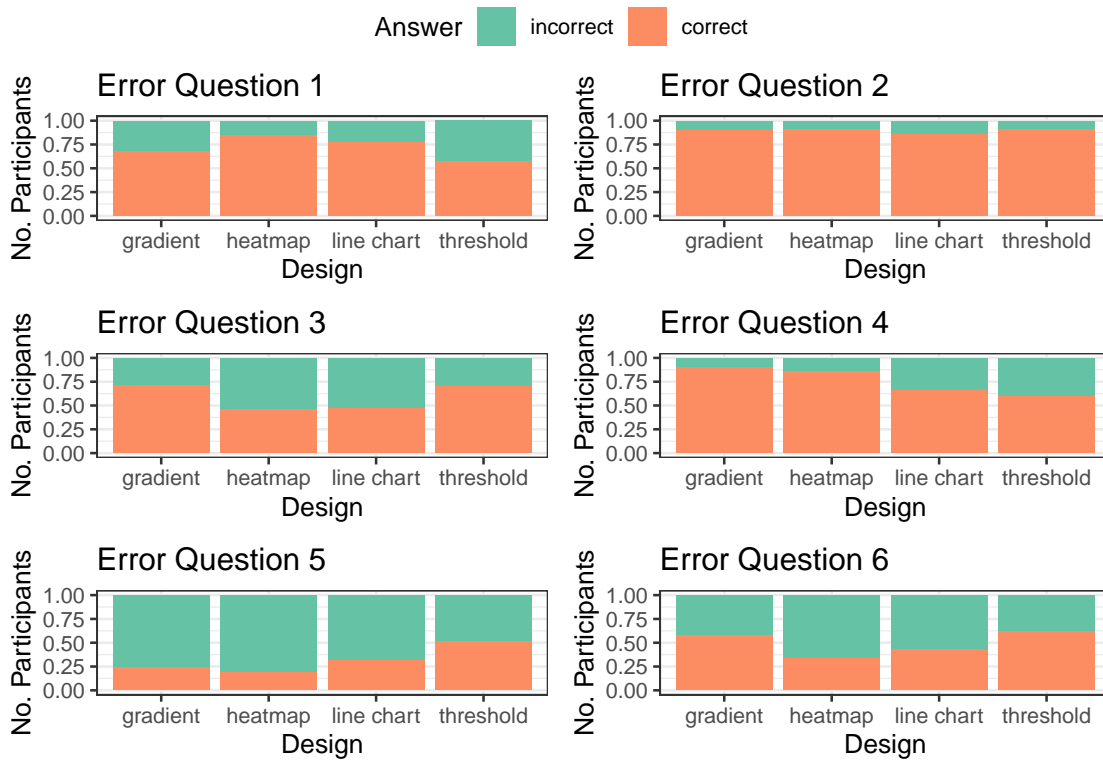


Figure 14.10: Results – Error Rates per question.

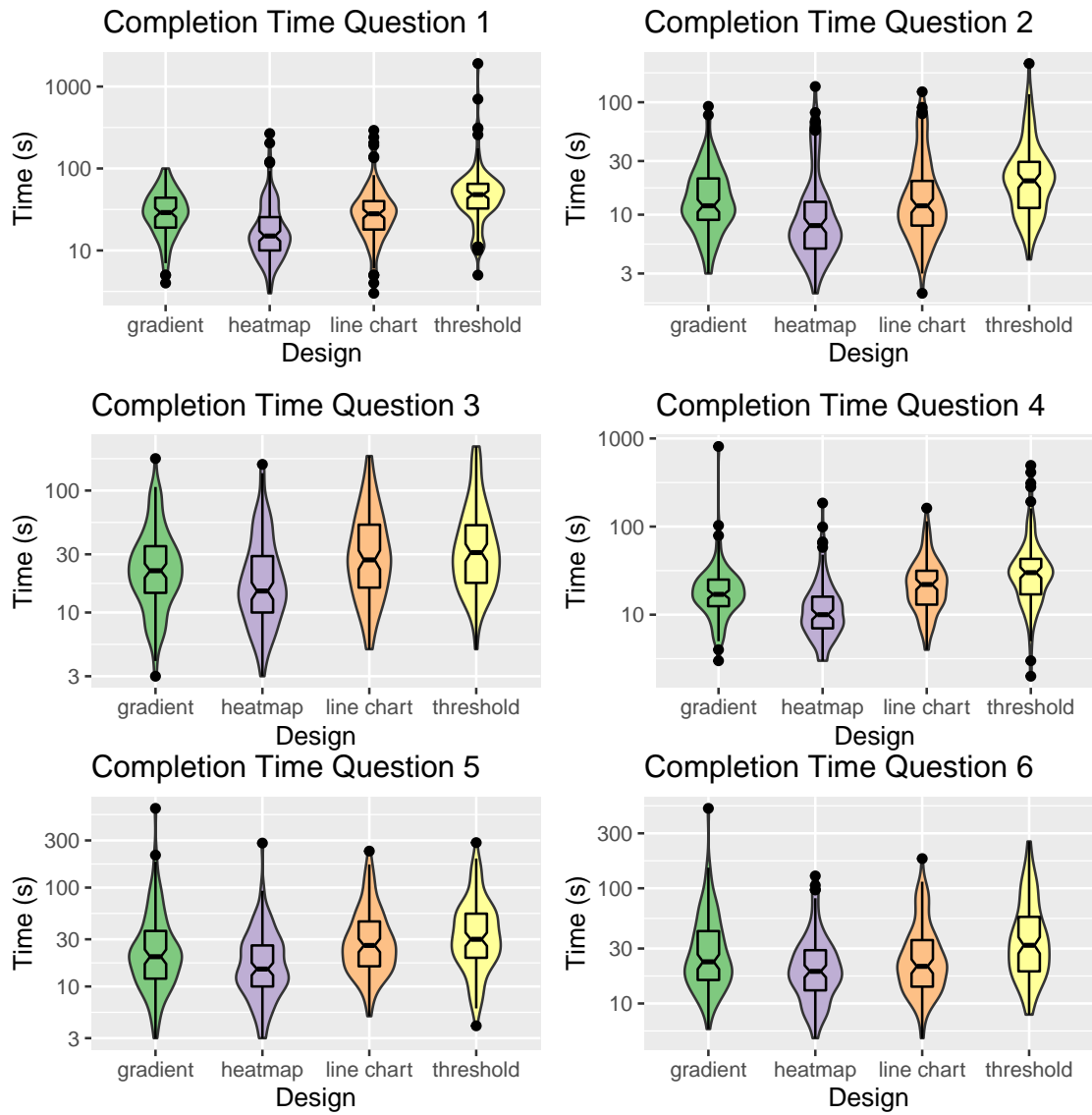


Figure 14.11: Results – Completion times per question.

14.2 User Study Results - Uncertainty in Time Series Segmentation Results

14.2.1 Hypotheses

- H_0 The *Gradient Uncertainty Plot* does not perform significantly worse than a *Composite Uncertainty and Segmentation Result Plot* for showing segment probabilities in segmented time series.
- H_1 The *Gradient Uncertainty Plot* that can be toggled does not perform worse when assessing uncertainties in segmented time series than an *Uncertainty Heatmap* showing only uncertainty.
- H_2 The *Gradient Uncertainty Plot* is more effective in conveying certainties of a segmented time series than an interactive *Threshold Uncertainty Plot*, especially if vertical space is insufficiently available
- H_3 The *Heatband Uncertainty Plot* is not inferior to the *Area Uncertainty Plot* for conveying uncertainty effectively over time.

Hypothesis Testing

H_2 will be tested using a Friedman test to calculate statistical significance, and a post-hoc Nemenyi test determining if the design pair in question, i.e., **gradient - threshold**, are significantly different, *followed by a superiority test*.

H_0 , H_0 , and H_0 will be tested using a non-inferiority test, evaluating if one used method is not significantly inferior to another. Using an equivalence test and only observing the *lower bound* will yield the test for non-inferiority (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3019319/>)

The bounds are calculated based on the statistical power of 0.95, the number of study participants $n = 111$, and the Significance level $\alpha = 0.05$, yielding the upper and lower bounds, of which only the **lower bound** will be of interest:

14.2.2 Significance Tests

Tests for significant differences between designs. Here we try to find significance particularly between the pair gradient and threshold, which would confirm H_2 with a significant pair **Gradient Uncertainty plot - threshold plot**.

Friedman Test - Error and Completion Time over all questions

Questions 1 to 6 error and Completion Time, including post-hoc Nemenyi test:

```
##
## Friedman rank sum test
##
```

14. VISUALIZING UNCERTAINTY IN TIME SERIES PROCESSING

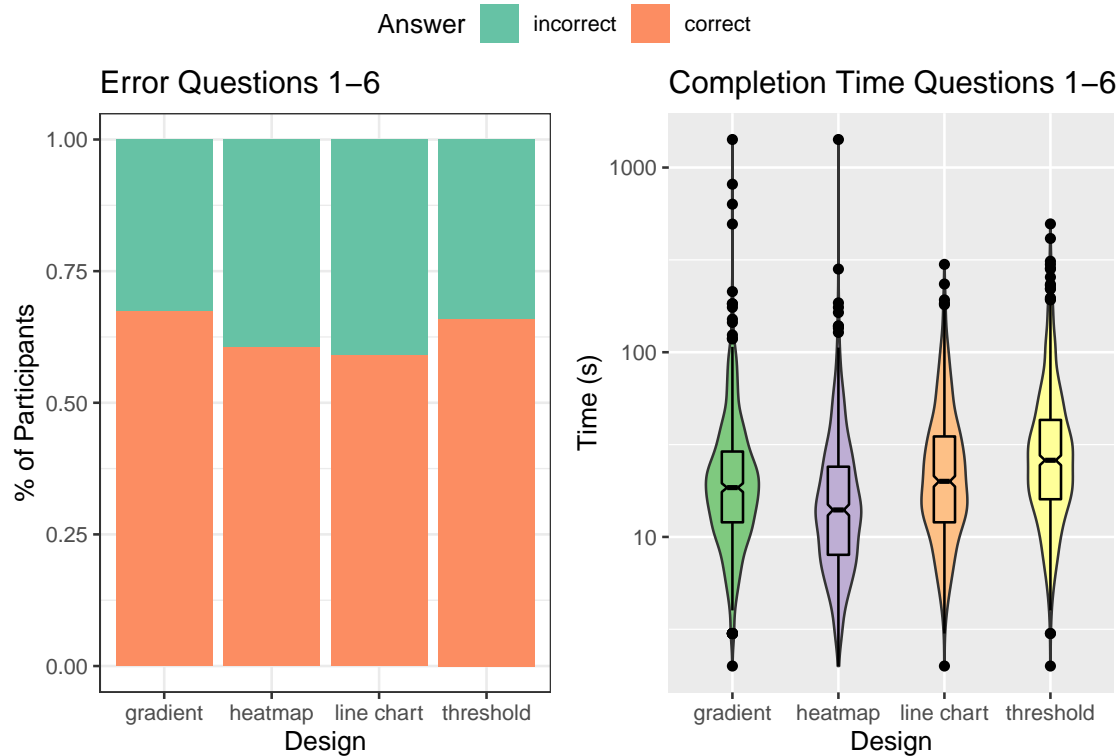
```
## data: u_scores_combined$question , u_scores_combined$design
## and u_scores_combined$id
## Friedman chi-squared = 19.341, df = 3, p-value = 0.0002324

##
## Friedman rank sum test
##
## data: u_scores_combined$time , u_scores_combined$design
## and u_scores_combined$id
## Friedman chi-squared = 286.03, df = 3, p-value < 2.2e-16

##
## Pairwise comparisons using Nemenyi multiple comparison test
## with q approximation for unreplicated blocked data
##
## data: question and design.f and id
##
## gradient heatmap line chart
## heatmap 0.224 - -
## line chart 0.082 0.966 -
## threshold 0.974 0.446 0.206
##
## P value adjustment method: none

##
## Pairwise comparisons using Nemenyi multiple comparison test
## with q approximation for unreplicated blocked data
##
## data: time and design.f and id
##
## gradient heatmap line chart
## heatmap 1.9e-12 - -
## line chart 0.04 3.4e-14 -
## threshold 2.9e-14 < 2e-16 2.8e-09
##
## P value adjustment method: none
```

Plots for Error and Completion Time over All Questions



Result

No significant pairs for scores, however, significance for Completion Time.

Friedman Test - Error and Completion Time for Questions 4 and 5

Error rate significantly lower especially for questions 4 and 5 would confirm that **Gradient Uncertainty** plot performs better than **threshold** plot for use cases where vertical space is limited.

```
##
## Friedman rank sum test
##
## data:  u_scores_q45$question , u_scores_q45$design
##        and u_scores_q45$id
## Friedman chi-squared = 5.0174, df = 3, p-value = 0.1705
##
##
## Friedman rank sum test
##
```

```
## data:  u_scores_q45$time , u_scores_q45$design
##          and u_scores_q45$id
## Friedman chi-squared = 160.9, df = 3, p-value < 2.2e-16

##
## Pairwise comparisons using Nemenyi multiple comparison test
##          with q approximation for unreplicated blocked data
##
## data:  time and design.f and id
##
##          gradient heatmap line chart
## heatmap    2.6e-07  -      -
## line chart  0.0085   3.5e-14 -
## threshold  2.8e-10  < 2e-16 0.0035
##
## P value adjustment method: none
```

Plots for Error and Completion Time over Questions 4 and 5

Error

Error Rate: No Significance.

Completion Time Result: Significant differences between all designs. Order: 1.**Uncertainty Heatmap**, 2.**Gradient Uncertainty plot**, 3.**composite line chart**, 4.**threshold plot**.

Friedman Test - Error and Completion Time for Questions 3 - 6 (Vertical Comparison)

Error rate significantly different especially for questions 3 - 6 would confirm that **Gradient Uncertainty plot** performs better than **threshold plot** for use cases where vertical space is limited.

```
##
## Friedman rank sum test
##
## data:  u_scores_q3456$question , u_scores_q3456$design
##          and u_scores_q3456$id
## Friedman chi-squared = 49.709, df = 3, p-value = 9.214e-11

##
## Friedman rank sum test
##
```

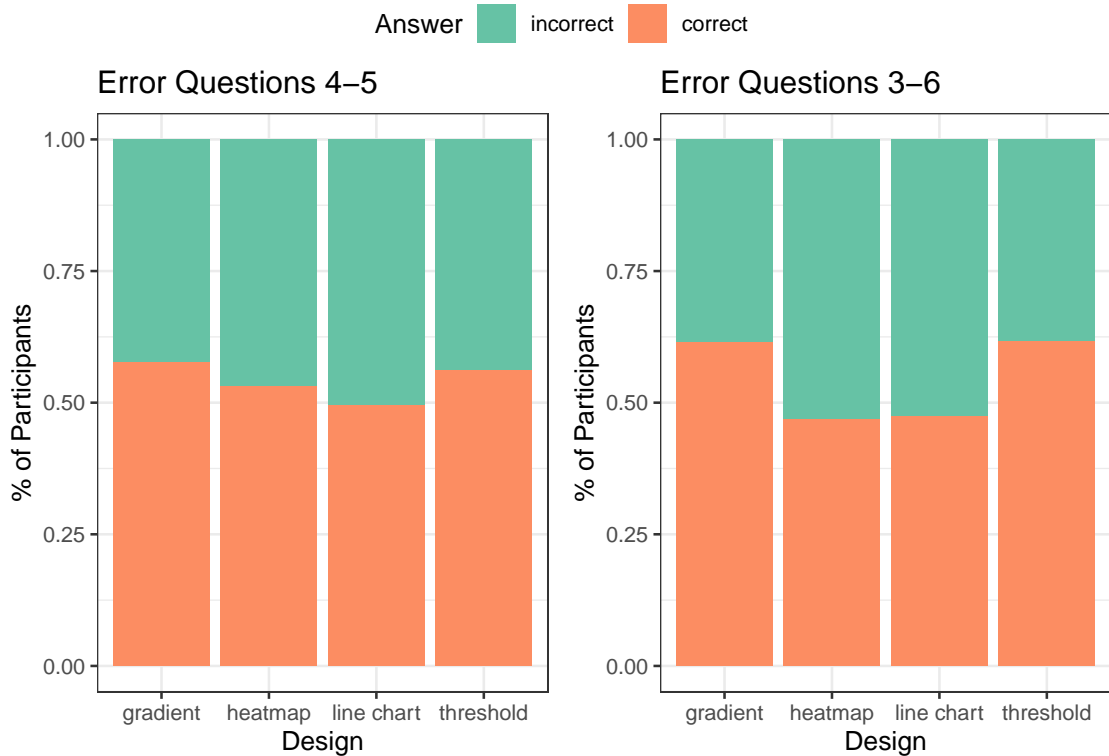
14.2. User Study Results - Uncertainty in Time Series Segmentation Results

```
## data: u_scores_q3456$time , u_scores_q3456$design
##          and u_scores_q3456$id
## Friedman chi-squared = 243.87, df = 3, p-value < 2.2e-16

##
## Pairwise comparisons using Nemenyi multiple comparison test
##          with q approximation for unreplicated blocked data
##
## data: question and design.f and id
##
##          gradient heatmap line chart
## heatmap    0.0041  -      -
## line chart 0.0069  0.9986  -
## threshold  0.9999  0.0034  0.0058
##
## P value adjustment method: none

##
## Pairwise comparisons using Nemenyi multiple comparison test
##          with q approximation for unreplicated blocked data
##
## data: time and design.f and id
##
##          gradient heatmap line chart
## heatmap    1.2e-10  -      -
## line chart 0.009    3.9e-14  -
## threshold  4.1e-14  < 2e-16  9.1e-07
##
## P value adjustment method: none
```

Plots for Error and Completion Time over Questions 3-6



Results

Error Rate - Significance between pairs:

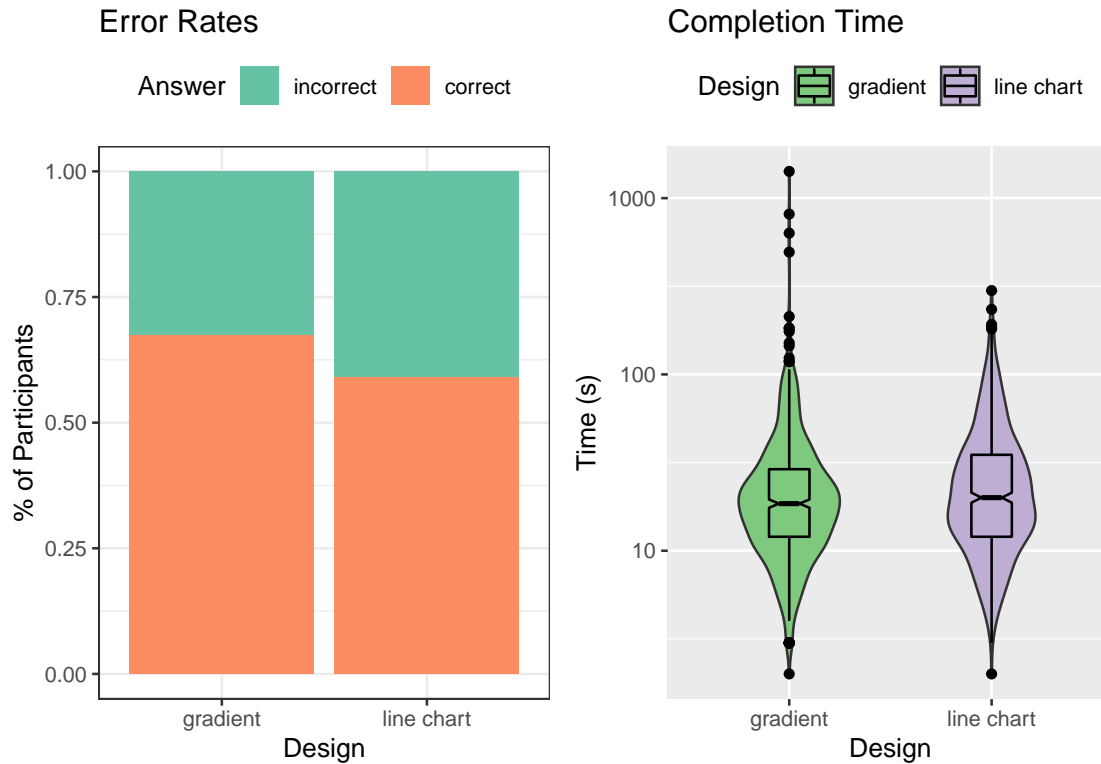
- **Gradient Uncertainty plot and Uncertainty Heatmap (0.0041)**
 - Gradient Uncertainty plot performed significantly better
- **Gradient Uncertainty plot and line plot (0.0069)**
 - Gradient Uncertainty plot performed significantly better
- **threshold plot and Uncertainty Heatmap (0.0034)**
 - Threshold Uncertainty plot performed significantly better
- **threshold plot and line plot (0.0058)**
 - Threshold Uncertainty plot performed significantly better

Completion Time Result: Significant differences between all designs. Order: 1.**Uncertainty Heatmap**, 2.**Gradient Uncertainty plot**, 3.**composite line chart**, 4.**Threshold Uncertainty plot**.

14.2.3 Non-Equivalence Test of Gradient Uncertainty Plot vs Composite Uncertainty and Segmentation Result Plot (H_0)

Testing for non-inferiority (error is lower) of Error ($q_1 - q_6$) and completion times ($t_{q1} - t_{q6}$) between **Gradient Uncertainty plot - line plot** (H_0).

##



Loading required namespace: jmvcore

TOST INDEPENDENT SAMPLES T-TEST

TOST Results

		t	df	p

question	t-test	3.192	1330	0.001
	TOST Upper	-0.413	1330	0.340
	TOST Lower	6.80	1330	< .001

##

```

##      time          t-test      0.228    1330     0.819
##              TOST Upper    -3.376    1330     < .001
##              TOST Lower     3.83     1330     < .001
## -----
##
##
## Equivalence Bounds
## -----
##              Low           High           Lower           Upper
## -----
## question    Cohen's d      -0.198     0.198
##              Raw           -0.0950    0.0950     0.0407     0.127
##
## time        Cohen's d      -0.198     0.198
##              Raw           -11.0433   11.0433    -4.3428     5.742
## -----

```

Result

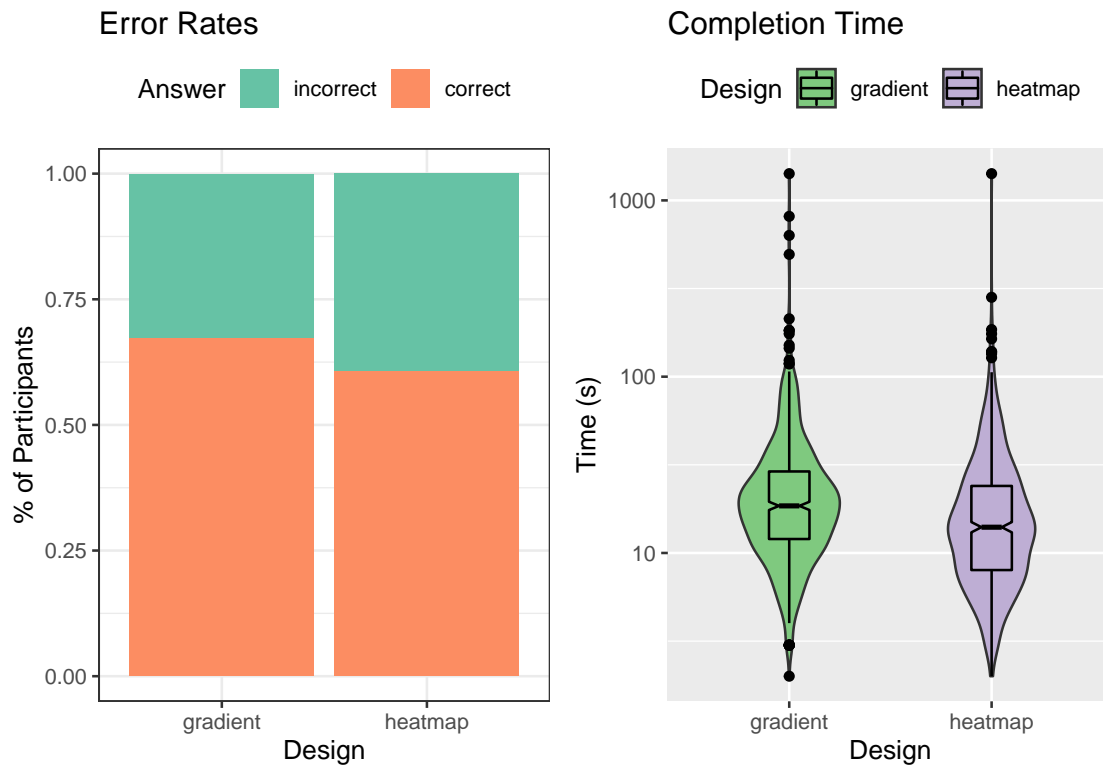
Score:

Completion Time:

14.2.4 Non-Equivalence Test of Gradient Uncertainty Plot vs Uncertainty Heatmap (H_1)

Testing for non-inferiority (error is lower) of Error ($q_1 - q_6$) and completion times ($t_{q1} - t_{q6}$) between **Gradient Uncertainty plot - Uncertainty Heatmap (H_1)**.

##



```
##
## TOST INDEPENDENT SAMPLES T-TEST
##
## TOST Results
## -----
##                t          df          p
## -----
## question      t-test      2.57      1330      0.010
##               TOST Upper  -1.03      1330      0.151
##               TOST Lower   6.18      1330      < .001
##
## time          t-test      2.06      1330      0.040
##               TOST Upper  -1.55      1330      0.061
##               TOST Lower   5.66      1330      < .001
## -----
##
##
## Equivalence Bounds
## -----
##                Low          High          Lower          Upper
## -----
```

```
## question    Cohen's d    -0.198    0.198
##            Raw        -0.0946   0.0946   0.0244   0.111
##
## time        Cohen's d    -0.198    0.198
##            Raw        -13.1132  13.1132  1.5003  13.476
## -----
```

Result

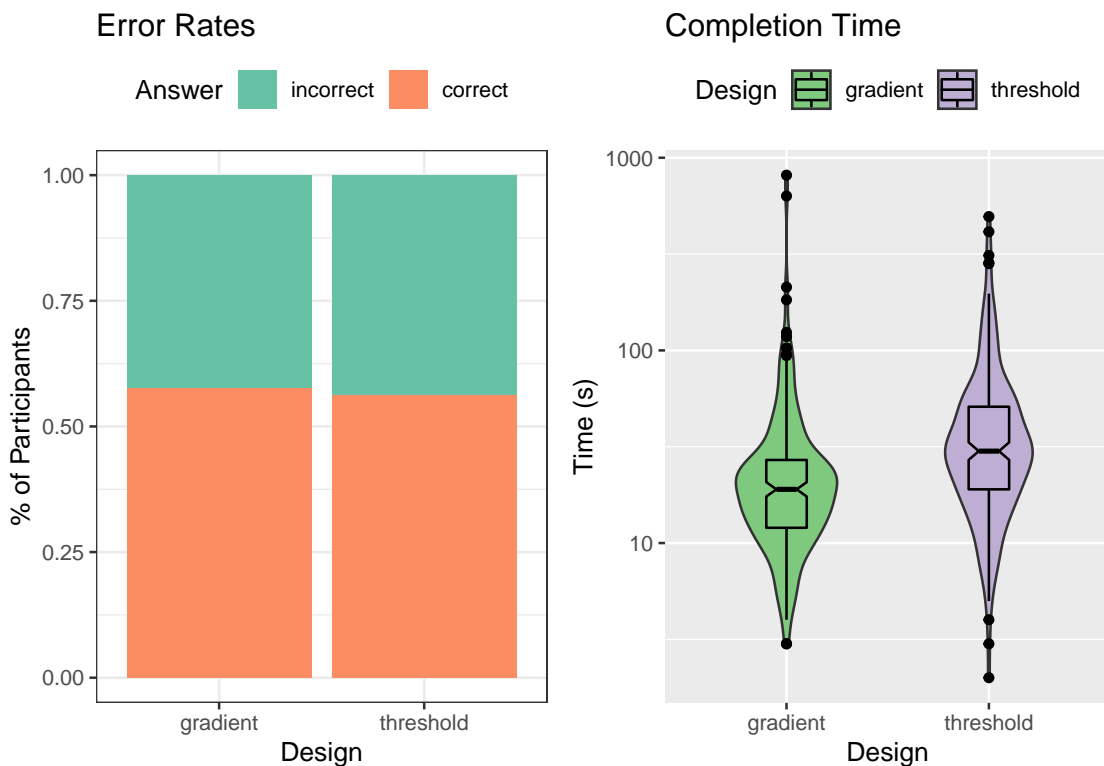
Score:

Completion Time:

14.2.5 Non-Equivalence Test of Gradient Uncertainty Plot vs Threshold Uncertainty Plot (H_2)

Testing for non-inferiority (error is lower) of Error ($q_1 - q_6$) and completion times ($t_{q1} - t_{q6}$) between **Gradient Uncertainty plot - threshold (H_2)**

##



##

14.2. User Study Results - Uncertainty in Time Series Segmentation Results

```
## TOST INDEPENDENT SAMPLES T-TEST
##
## TOST Results
## -----
##                t                df                p
## -----
## question      t-test              0.287          442          0.774
##               TOST Upper          -3.32          442          < .001
##               TOST Lower           3.89          442          < .001
##
## time          t-test             -2.355          442          0.019
##               TOST Upper          -5.96          442          < .001
##               TOST Lower           1.25          442          0.106
## -----
##
##
## Equivalence Bounds
## -----
##                Low                High                Lower                Upper
## -----
## question      Cohen's d          -0.342          0.342
##               Raw                -0.170          0.170          -0.0641          0.0911
##
## time          Cohen's d          -0.342          0.342
##               Raw                -22.510         22.510          -24.9997         -4.4147
## -----
```

Result

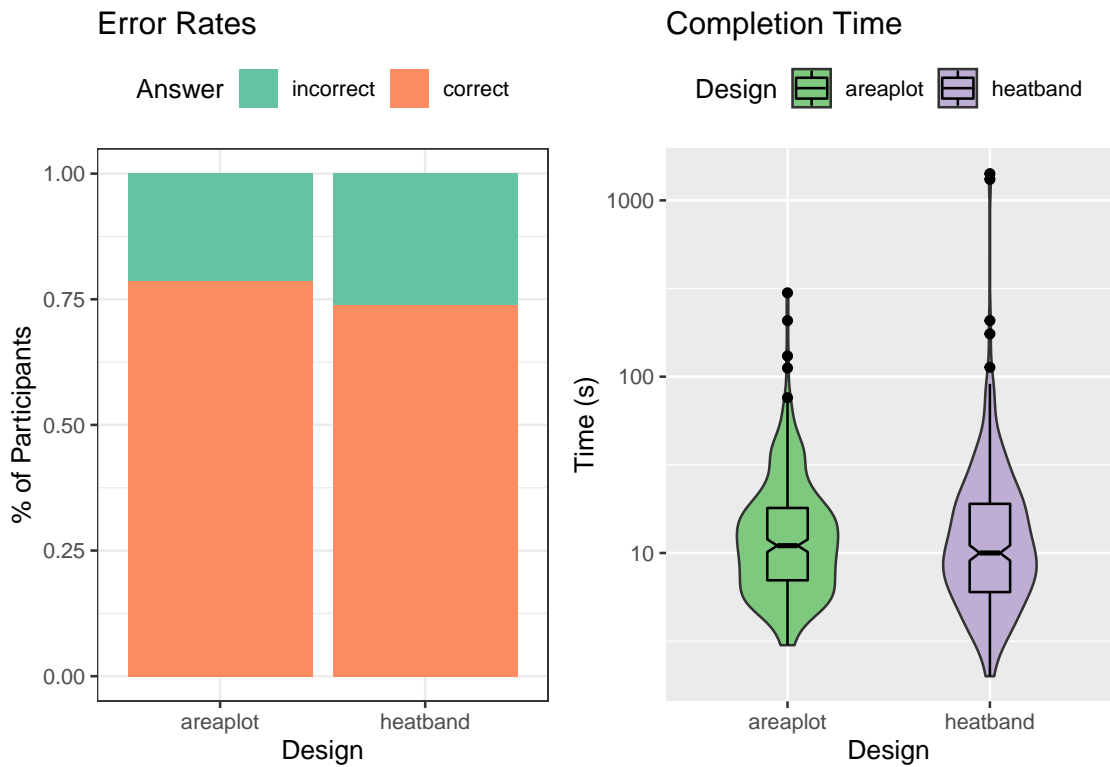
Score:

Completion Time:

14.2.6 Non-Equivalence Test of Area Plot vs. Heat Bands (H_3)

Testing for non-inferiority (error is lower) of Error ($q_1 - q_3$) and completion times ($t_{q1} - t_{q3}$) between **area plot - heat bands** (H_3).

##



```
##
## TOST INDEPENDENT SAMPLES T-TEST
##
## TOST Results
## -----
##          t          df          p
## -----
## question  t-test      1.46      664      0.145
##           TOST Upper  -2.15      664      0.016
##           TOST Lower   5.06      664      < .001
##
## time      t-test     -1.29      664      0.197
##           TOST Upper  -4.90      664      < .001
##           TOST Lower   2.31      664      0.010
## -----
##
##
## Equivalence Bounds
## -----
##          Low          High          Lower          Upper
## -----
```

14.2. User Study Results - Uncertainty in Time Series Segmentation Results

## question	Cohen's d	-0.279	0.279		
##	Raw	-0.119	0.119	-0.00625	0.102
##					
## time	Cohen's d	-0.279	0.279		
##	Raw	-21.581	21.581	-17.58762	2.134
##	-----				

error

- Non-inferiority confirmed in q_1 , q_2 , and q_3 .
- Equality confirmed in q_2 and q_3 .
- Area plot is superior in q_1 .

Completion Time

- Equality (and subsequently non-inferiority) confirmed in q_1 , q_2 , and q_3 .

14.2.7 Hypotheses Tested

H_0 Gradient Uncertainty Plot vs. Composite Uncertainty Visualization

Errors: Gradient Plot is superior to Composite Uncertainty Visualization

Completion Time: Equality confirmed.

H_0 non-inferiority **confirmed**, even **superiority** of gradient plot for errors.

H_1 Gradient Uncertainty Plot vs. Uncertainty Heatmap

Errors: Gradient Plot is superior to Uncertainty Heatmap

Completion Time: Heatmap is superior to Gradient Plot.

H_1 non-inferiority **confirmed**.

H_2 Gradient Uncertainty Plot vs. Threshold Uncertainty Plot

Errors: Gradient Plot is not significantly better than Threshold Uncertainty Plot, pairs not significant according to post-hoc Nemenyi test ($p=0.974$).

Completion Time: Gradient Plot is significantly better than Threshold Uncertainty Plot.

H_2 can only be **confirmed for completion times**.

H_{2a} - Limited Vertical Space Errors: Friedman Test non-significant

Completion Time: Gradient Plot is significantly better than Threshold Uncertainty Plot.

H_{2a} is **not confirmed** for errors, but can again be **confirmed for completion times**.

H_3 Difference between Heatband and Area Charts Uncertainty

Errors: Equivalence confirmed.

Completion Time: Equivalence confirmed.

H_3 can be **confirmed** with equivalence.

14.2.8 Implications

For Question 1 and 2 comparisons had to be made between segments from one result, meaning that horizontally comparisons could be made well using line charts or heatmaps. However, in Questions 3 to 6, comparison had to be made across segmentation results visualized as rows, which seems to be more difficult when using the Composite Visualization: There were noticeable differences in results for Question 3, 4, and 6 where the Gradient Uncertainty Plot outperformed the Composite Visualization (H_0), while times employed using the Gradient Uncertainty Plot were not significantly longer.

Question 4 was aimed to test the effectiveness of uncertainty visualization designs for limited vertical space, in which the Gradient Uncertainty Plot had significantly higher error than the Composite (H_0) and Threshold Uncertainty Visualization (H_2) and Completion Time not inferior to other designs, except for the Uncertainty Heatmap (H_1).

Question 5 had the overall worst error rate, which we infer was due to the difficulty of the question being two very similar segment uncertainties. In this case, the Threshold Uncertainty Plot significantly outperformed the Gradient Uncertainty Plot (H_2) and Uncertainty Heatmap. However, the Completion Time was still significantly worse than both of these designs. Error were also low for the Gradient Uncertainty Plot, which was out of line with other questions with multiple segmentation results visualized (Question 3-6).

Two questions in the test were more difficult to answer (Q1, Q5): differences between uncertainty in the segments and areas were smaller than in other questions. Participants took longer to answer these questions, and had worse error compared to similar questions:

- Question 1 and 2 are similar, horizontal intervals must be compared:
 - Mean Error **Q1: 0.277027**, Q2: 0.1036036
 - Median Completion Time **Q1: 29**, Q2: 12
- Question 4 and 5 are similar, horizontal and vertical comparison with vertical space available.
 - Mean Error Q4: 0.2387387, **Q5: 0.6779279**
 - Median Completion Time Q4: 18, **Q5: 23**

(Question 5 even had error rates above 50%, except for the Uncertainty Threshold Plot).

This implies that the aggregated uncertainty of an interval is hard to judge mentally and without visual support. We suggest employing an aggregated uncertainty

Glossary

DQProv Explorer Data Quality Provenance Explorer is a VA approach to visualizing provenance that was captured by our data wrangling provenance model.. 107, 108, 124, 125, 141–144, 162, 167, 173

MetricDoc An environment for the visual-interactive customization of data quality metrics. 14, 15, 90–93, 100, 101, 109, 121, 131–133, 136–139, 161, 163, 166

Acronyms

DoI degree-of-interest. 70

DQ Data Quality. 3, 4, 12, 14, 17–26, 28, 37, 38, 40, 41, 43–45, 49, 50, 52–54, 59, 71, 72, 75–78, 81, 85, 89–94, 96–98, 100, 101, 111–116, 121, 122, 132–137, 139, 140, 144, 161–164, 166, 167

F+C Focus+Context. 50

MVTS multivariate time series. 15, 38, 61, 81–84, 111, 116, 117, 127, 128, 145, 146, 150, 158, 163

PDF probability density function. 30, 31

VA Visual Analytics. 4–14, 17, 28, 30–32, 36, 37, 50, 53, 54, 59, 61–64, 67–69, 72, 75, 86, 87, 100, 101, 111, 116, 121, 128, 129, 145, 146, 161–164, 167

Bibliography

- [AK06] B.M. Ayyub and G.J. Klir. *Uncertainty Modeling and Analysis in Engineering and the Sciences*. Taylor & Francis, 2006.
- [ALA⁺18] N. Andrienko, T. Lammarsch, G. Andrienko, G. Fuchs, D. Keim, S. Miksch, and A. Rind. Viewing Visual Analytics as Model Building. *Computer Graphics Forum*, 37(6):275–299, 2018.
- [Alf08] Andreas Alfons. *Robust Methods for High-Dimensional Data*, 2016-01-08. TopGear Dataset. Robust methods for high-dimensional data, in particular linear model selection techniques based on least angle regression and sparse regression. (v0.5.1).
- [AMA05] S. Adelman, L.T. Moss, and M. Abai. *Data Strategy*. Addison-Wesley, 2005.
- [AMST11] Wolfgang Aigner, Silvia Miksch, Heidrun Schumann, and Christian Tominski. Time & Time-Oriented Data. In *Visualization of Time-oriented Data*. Springer, 2011.
- [AMTB05] W. Aigner, S. Miksch, B. Thurnher, and S. Biffl. PlanningLines: Novel Glyphs for Representing Temporal Uncertainties and their Evaluation. In *Ninth International Conference on Information Visualisation (IV'05)*, pages 457–463, July 2005.
- [Ans73] F. J. Anscombe. Graphs in Statistical Analysis. *The American Statistician*, 27(1):17–21, February 1973.
- [ASMP17] C. Arbesser, F. Spechtenhauser, T. Mühlbacher, and H. Piringer. Visplause: Visual Data Quality Assessment of Many Time Series Using Plausibility Checks. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):641–650, January 2017.
- [BAL12] Ken Brodlie, Rodolfo Allendes Osorio, and Adriano Lopes. A Review of Uncertainty in Data Visualization. In John Dill, Rae Earnshaw, David Kasik, John Vince, and Pak Chung Wong, editors, *Expanding the Frontiers of Visual Analytics and Visualization*, pages 81–109. Springer London, 2012.

- [BBB⁺18] Jürgen Bernard, Christian Bors, Markus Bögl, Christian Eichner, Theresia Gschwandtner, Silvia Miksch, Heidrun Schumann, and Jörn Kohlhammer. Combining the Automated Segmentation and Visual Analysis of Multivariate Time Series. In *EuroVis Workshop on Visual Analytics (EuroVA)*, pages 49–53. The Eurographics Association, 2018.
- [BBB⁺19] Christian Bors, Jürgen Bernard, Markus Bögl, Theresia Gschwandtner, Jörn Kohlhammer, and Silvia Miksch. Quantifying Uncertainty in Multivariate Time Series Pre-Processing. In *EuroVis Workshop on Visual Analytics (EuroVA)*, pages 31–35. The Eurographics Association, 2019.
- [BBGM17] Christian Bors, Markus Bögl, Theresia Gschwandtner, and Silvia Miksch. Visual Support for Rastering of Unequally Spaced Time Series. In *Proceedings of the 10th International Symposium on Visual Information Communication and Interaction, VINCI '17*, pages 53–57, New York, NY, USA, 2017. ACM. Best Short Paper.
- [BBGM18] Markus Bögl, Christian Bors, Theresia Gschwandtner, and Silvia Miksch. *Uncertainty Types in Segmenting and Labeling Time Series Data*. Data Science, Statistics & Visualisation Conference (DSSV). 2018.
- [BCC⁺05] L. Bavoil, S. P. Callahan, P. J. Crossno, J. Freire, C. E. Scheidegger, C. T. Silva, and H. T. Vo. VisTrails: Enabling Interactive Multiple-view Visualizations. In *IEEE Visualization, 2005.*, pages 135–142, October 2005.
- [BCEH05] Jan Van den Broeck, Solveig Argeseanu Cunningham, Roger Eeckels, and Kobus Herbst. Data Cleaning: Detecting, Diagnosing, and Editing Data Abnormalities. *PLOS Medicine*, 2(10):e267, June 2005.
- [BCFM09] Carlo Batini, Cinzia Cappiello, Chiara Francalanci, and Andrea Maurino. Methodologies for Data Quality Assessment and Improvement. *ACM Comput. Surv.*, 41(3):16:1–16:52, July 2009.
- [BDB⁺16] Jürgen Bernard, Eduard Dobermann, Markus Bögl, Martin Röhlig, Anna Vögele, and Jörn Kohlhammer. Visual-Interactive Segmentation of Multivariate Time Series. In *EuroVis Workshop on Visual Analytics (EuroVA)*, pages 31–35. The Eurographics Association, 2016.
- [BG05] José Barateiro and Helena Galhardas. A Survey of Data Quality Tools. *Datenbank-Spektrum*, 14(15–21):48, 2005.
- [BGK⁺18] Christian Bors, Theresia Gschwandtner, Simone Kriglstein, Silvia Miksch, and Margit Pohl. Visual Interactive Creation, Customization, and Analysis of Data Quality Metrics. *Journal of Data and Information Quality*, 10(1):3:1–3:26, May 2018.

- [BGM19] C. Bors, T. Gschwandtner, and S. Miksch. Capturing and Visualizing Provenance from Data Wrangling. *IEEE Computer Graphics and Applications*, 39:61–75, 2019.
- [BH19] Leilani Battle and Jeffrey Heer. Characterizing Exploratory Visual Analysis: A Literature Review and Evaluation of Analytic Provenance in Tableau. *Computer Graphics Forum*, 38(3):145–159, 2019.
- [BHJ⁺14] Georges-Pierre Bonneau, Hans-Christian Hege, Chris R. Johnson, Manuel M. Oliveira, Kristin Potter, Penny Rheingans, and Thomas Schultz. Overview and State-of-the-Art of Uncertainty Visualization. In *Scientific Visualization, Mathematics and Visualization*, pages 3–27. Springer, London, 2014.
- [BHR⁺19] Jürgen Bernard, Marco Hutter, Heiko Reinemuth, Hendrik Pfeifer, Christian Bors, and Jörn Kohlhammer. Visual-Interactive Preprocessing of Multivariate Time Series Data. *Computer Graphics Forum*, 38(3):401–412, 2019.
- [BKT01] Peter Buneman, Sanjeev Khanna, and Wang Chiew Tan. Why and Where: A Characterization of Data Provenance. In Jan Van den Bussche and Victor Vianu, editors, *International Conference on Database Theory*, pages 316–330. Springer, LNCS 1973, 2001.
- [BLBC12] E. T. Brown, J. Liu, C. E. Brodley, and R. Chang. Dis-function: Learning Distance Functions Interactively. In *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 83–92, October 2012.
- [BM13] M. Brehmer and T. Munzner. A Multi-Level Typology of Abstract Visualization Tasks. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2376–2385, December 2013.
- [BOZ⁺14] Eli T. Brown, Alvitta Ottley, Helen Zhao, null Quan Lin, Richard Souvenir, Alex Endert, and Remco Chang. Finding Waldo: Learning about Users from their Interactions. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1663–1672, December 2014.
- [BPHE17] Nadia Boukhelifa, Marc-Emmanuel Perrin, Samuel Huron, and James Eagan. How Data Workers Cope with Uncertainty: A Task Characterisation Study. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, CHI '17*, pages 3645–3656, New York, NY, USA, 2017. ACM.
- [Bre09] K. Brennan. *A Guide to the Business Analysis Body of Knowledge*. International Institute of Business Analysis, 2009.

- [BRG⁺12] Jürgen Bernard, Tobias Ruppert, Oliver Goroll, Thorsten May, and Jörn Kohlhammer. Visual-Interactive Preprocessing of Time Series Data. In Andreas Kerren and Stefan Seipel, editors, *SIGRAD*, volume 81 of *Linköping Electronic Conference Proceedings*, pages 39–48. Linköping University Electronic Press, 2012.
- [BS06] Carlo Batini and Monica Scannapieco. *Data Quality: Concepts, Methodologies and Techniques (Data-Centric Systems and Applications)*. Springer Verlag New York, Inc., Secaucus, NJ, USA, 2006.
- [BS16] Carlo Batini and Monica Scannapieco. *Data and Information Quality - Dimensions, Principles and Techniques*. Springer International Publishing, 1st edition, 2016.
- [BW88] Barbara P. Buttenfield and Robert Weibel. Visualizing the Quality of Cartographic Data. In *Proc. Third International Geographic Information Systems Symposium (GIS/LIS)*, 1988.
- [BYB⁺13] Michelle A. Borkin, Chelsea S. Yeh, Madelaine Boyd, Peter Macko, Krzysztof Z. Gajos, Margo Seltzer, and Hanspeter Pfister. Evaluation of Filesystem Provenance Visualization Tools. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2476–2485, December 2013.
- [CAB⁺14] Lucian Carata, Sherif Akoush, Nikilesh Balakrishnan, Thomas Bytheway, Ripduman Sohan, Margo Seltzer, and Andy Hopper. A Primer on Provenance. *Queue*, 12(3):10:10–10:23, March 2014.
- [CAFG12] Michael Correll, Danielle Albers, Steven Franconeri, and Michael Gleicher. Comparing Averages in Time Series Data. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 1095–1104, New York, NY, USA, 2012. ACM. event-place: Austin, Texas, USA.
- [CB94] Dai Clegg and Richard Barker. *Case Method Fast-Track: A Rad Approach*. Addison-Wesley Longman Publishing Co., Inc., 1994.
- [CB04] Catherine Courage and Kathy Baxter. *Understanding Your Users: A Practical Guide to User Requirements Methods, Tools, and Techniques*. Morgan Kaufmann Publishers Inc., 2004.
- [CBK09] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly Detection: A Survey. *ACM Comput. Surv.*, 41(3):15:1–15:58, July 2009.
- [CCM09] C. Correa, Yu-Hsuan Chan, and Kwan-Liu Ma. A Framework for Uncertainty-aware Visual Analytics. In *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on*, pages 51–58. IEEE, 2009.

- [CFS⁺06] Steven P. Callahan, Juliana Freire, Emanuele Santos, Carlos E. Scheidegger, Cláudio T. Silva, and Huy T. Vo. VisTrails: Visualization Meets Data Management. In *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data*, SIGMOD '06, pages 745–747, New York, NY, USA, 2006. ACM.
- [CGOG11] M. Correll, S. Ghosh, D. O'Connor, and M. Gleicher. Visualizing Virus Population Variability from Next Generation Sequencing Data. In *2011 IEEE Symposium on Biological Data Visualization (BioVis)*, pages 135–142, October 2011.
- [CH17] Michael Correll and Jeffrey Heer. Regression by Eye: Estimating Trends in Bivariate Visualizations. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 1387–1396, New York, NY, USA, 2017. ACM.
- [Che13] Reynold Cheng. Managing Quality of Probabilistic Databases. In Shazia Sadiq, editor, *Handbook of Data Quality*, pages 271–291. Springer Berlin Heidelberg, 2013.
- [CI81] W. J. Conover and Ronald L. Iman. Rank Transformations as a Bridge between Parametric and Nonparametric Statistics. *The American Statistician*, 35(3):124–129, 1981.
- [CLKS18] M. Correll, M. Li, G. Kindlmann, and C. Scheidegger. Looks Good To Me: Visualizations As Sanity Checks. *IEEE Transactions on Visualization and Computer Graphics*, pages 830–839, 2018.
- [CLNL87] Daniel B. Carr, Richard J. Littlefield, W. L. Nicholson, and J. S. Littlefield. Scatterplot matrix techniques for large N. *Journal of the American Statistical Association*, 82(398):424–436, 1987.
- [CMH18] Michael Correll, Dominik Moritz, and Jeffrey Heer. Value-Suppressing Uncertainty Palettes. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 642:1–642:11, New York, NY, USA, 2018. ACM.
- [CMS99] Stuart K. Card, Jock D. Mackinlay, and Ben Shneiderman, editors. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.
- [CWR14] Lei Cao, Qingyang Wang, and Elke A. Rundensteiner. Interactive Outlier Exploration in Big Data Streams. *Proc. VLDB Endow.*, 7(13):1621–1624, August 2014.

- [CZC⁺15] Haidong Chen, Song Zhang, Wei Chen, Honghui Mei, Jiawei Zhang, Andrew Mercer, Ronghua Liang, and Huamin Qu. Uncertainty-Aware Multidimensional Ensemble Data Visualization and Exploration. *IEEE Transactions on Visualization and Computer Graphics*, 21(9):1072–1086, September 2015.
- [Das13] Tamraparni Dasu. Data Glitches: Monsters in Your Data. In Shazia Sadiq, editor, *Handbook of Data Quality*, pages 163–178. Springer Berlin Heidelberg, 2013.
- [dCCM09] S.M.S. da Cruz, M.L.M. Campos, and M. Mattoso. Towards a Taxonomy of Provenance in Scientific Workflow Management Systems. In *2009 World Conference on Services - I*, pages 259–266, July 2009.
- [DDG⁺16] Jeremy Debattista, Makx Dekkers, Christophe Guéret, Deirdre Lee, Nandana Mihindukulasooriya, and Amrapali Zaveri. Data on the Web Best Practices: Data Quality Vocabulary, December 2016.
- [DF08] Susan B. Davidson and Juliana Freire. Provenance and Scientific Workflows: Challenges and Opportunities. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1345–1350. ACM, 2008.
- [DHRL⁺12] Cody Dunne, Nathalie Henry Riche, Bongshin Lee, Ron Metoyer, and George Robertson. GraphTrail: Analyzing Large Multivariate, Heterogeneous Networks while Supporting Exploration History. In *In Proceedings of ACM SIGCHI 2012*. ACM, May 2012.
- [DJ03] Tamraparni Dasu and Theodore Johnson. *Exploratory Data Mining and Data Cleaning*. John Wiley & Sons, Inc., New York, NY, USA, 1 edition, 2003.
- [DLW⁺17] A. Dasgupta, J. Y. Lee, R. Wilson, R. A. Lafrance, N. Cramer, K. Cook, and S. Payne. Familiarity Vs Trust: A Comparative Study of Domain Scientists & Trust in Visual Analytics and Conventional Analysis Methods. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):271–280, January 2017.
- [DRdS07] Nicholas Del Rio and Paulo Pinheiro da Silva. Probe-It! Visualization Support for Provenance. In *Advances in Visual Computing*, Lecture Notes in Computer Science, pages 732–741. Springer Berlin Heidelberg, 2007.
- [EFN12] Alex Endert, Patrick Fiaux, and Chris North. Semantic Interaction for Visual Text Analytics. pages 473–482. ACM, May 2012.
- [EPD05] Cyntrica Eaton, Catherine Plaisant, and Terence Drisd. Visualizing Missing Data: Graph Interpretation User Study. In Maria Francesca Costabile and Fabio Paternò, editors, *Human-Computer Interaction - INTERACT*

2005, Lecture Notes in Computer Science, pages 861–872. Springer Berlin Heidelberg, 2005.

- [FAAM16] Paolo Federico, Albert Amor-Amorós, and Silvia Miksch. A Nested Workflow Model for Visual Analytics Design and Validation. In *Proc. of the Sixth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualisation (BELIV '16)*. ACM, 2016.
- [Fer18] Sara Johansson Fernstad. To Identify What is Not There: A Definition of Missingness Patterns and Evaluation of Missing Value Visualization. *Information Visualization*, page 1473871618785387, July 2018.
- [FJ10] Camilla Forsell and Jimmy Johansson. An Heuristic Set for Evaluation in Information Visualization. In *Proceedings of the International Conference on Advanced Visual Interfaces, AVI '10*, pages 199–206. ACM, 2010.
- [FKSS08] J. Freire, D. Koop, E. Santos, and C. T. Silva. Provenance for Computational Tasks: A Survey. *Computing in Science Engineering*, 10(3):11–21, May 2008.
- [FPS14] Carla M.D.S. Freitas, Marcelo S. Pimenta, and Dominique L. Scapin. User-centered Evaluation of Information Visualization Techniques: Making the HCI-InfoVis Connection Explicit. In Weidong Huang, editor, *Handbook of Human Centric Visualization*, pages 315–336. Springer, 2014.
- [FSC⁺06] Juliana Freire, Cláudio T. Silva, Steven P. Callahan, Emanuele Santos, Carlos E. Scheidegger, and Huy T. Vo. Managing Rapidly-Evolving Scientific Workflows. In Luc Moreau and Ian Foster, editors, *Provenance and Annotation of Data*, Lecture Notes in Computer Science, pages 10–18. Springer Berlin Heidelberg, 2006.
- [FWM⁺18] Michael Fernandes, Logan Walls, Sean Munson, Jessica Hullman, and Matthew Kay. Uncertainty Displays Using Quantile Dotplots or CDFs Improve Transit Decision-Making. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, CHI '18*, pages 144:1–144:12, New York, NY, USA, 2018. ACM.
- [FWR⁺17] P. Federico, M. Wagner, A. Rind, A. Amor-Amorós, S. Miksch, and W. Aigner. The Role of Explicit Knowledge: A Conceptual Model of Knowledge-Assisted Visual Analytics. In *2017 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 92–103, October 2017.
- [Gal07] Wilbert O. Galitz. *The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques*. Wiley & Sons, 2007.

- [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, i-KNOW '14*, pages 18:1–18:8. ACM, New York, NY, USA, 2014.
- [GBFM16] T. Gschwandtner, M. Bögl, P. Federico, and S. Miksch. Visual Encodings of Temporal Uncertainty: A Comparative User Study. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):539–548, January 2016.
- [GDK⁺07] B. Glavic, K. R. Dittrich, A. Kemper, H. Schöning, T. Rose, M. Jarke, T. Seidl, C. Quix, and C. Brochhaus. Data provenance: A Categorization of existing approaches. *BTW '07: Datenbanksysteme in Business, Technologie und Web*, (103):227–241, March 2007.
- [GE18] T. Gschwandtner and O. Erhart. Know Your Enemy: Identifying Quality Problems of Time Series Data. In *2018 IEEE Pacific Visualization Symposium (PacificVis)*, pages 205–214, April 2018.
- [GFSS00] Helena Galhardas, Daniela Florescu, Dennis Shasha, and Eric Simon. AJAX: An Extensible Data Cleaning Tool. In *ACM Sigmod Record*, volume 29, page 590. ACM, 2000.
- [GGAM12] Theresia Gschwandtner, Johannes Gärtner, Wolfgang Aigner, and Silvia Miksch. A Taxonomy of Dirty Time-Oriented Data. In Gerald Quirchmayr, Josef Basl, Ilsun You, Lida Xu, and Edgar Weippl, editors, *Lecture Notes in Computer Science (LNCS 7465): Multidisciplinary Research and Practice for Information Systems (Proceedings of the CD-ARES 2012)*, pages 58–72, Prague, Czech Republic, 2012. Springer, Berlin / Heidelberg.
- [GKHH11] Philip J. Guo, Sean Kandel, Joseph M. Hellerstein, and Jeffrey Heer. Proactive Wrangling: Mixed-initiative End-user Programming of Data Transformation Scripts. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pages 65–74, New York, NY, USA, 2011. ACM.
- [GLG⁺16] S. Gratzl, A. Lex, N. Gehlenborg, N. Cosgrove, and M. Streit. From Visual Exploration to Storytelling and Back Again. *Computer Graphics Forum*, 35(3):491–500, 2016.
- [Gol13] Lukasz Golab. Data Warehouse Quality: Summary and Outlook. In Shazia Sadiq, editor, *Handbook of Data Quality*, pages 121–140. Springer Berlin Heidelberg, 2013.

- [GS06a] Henning Griethe and Heidrun Schumann. The Visualization of Uncertain Data: Methods and Problems. In *Proceedings of SimVis '06*, pages 143–156. SCS Publishing House, 2006.
- [GS06b] D. P. Groth and K. Streefkerk. Provenance and Annotation for Visual Exploration Systems. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1500–1510, November 2006.
- [GZ09] David Gotz and Michelle X. Zhou. Characterizing Users’ Visual Analytic Activity for Insight Provenance. *Information Visualization*, 8(1):42–55, January 2009.
- [Har09] Olaf Hartig. Provenance Information in the Web of Data. *Proceedings of the WWW2009 Workshop on Linked Data on the Web*, 538, April 2009.
- [HDL17] Melanie Herschel, Ralf Diestelkämper, and Houssem Ben Lahmar. A Survey on Provenance: What For? What Form? What From? *The VLDB Journal*, 26(6):881–906, December 2017.
- [Hel08] Joseph M. Hellerstein. *Quantitative Data Cleaning for Large Databases*. United Nations Economic Commission for Europe, technical report edition, 2008.
- [HG15] Rinke Hoekstra and Paul Groth. PROV-O-Viz - Understanding the Role of Activities in Provenance. In Bertram Ludäscher and Beth Plale, editors, *Provenance and Annotation of Data and Processes*, Lecture Notes in Computer Science, pages 215–220. Springer International Publishing, 2015.
- [HGR94] G. J. Hunter, M. F. Goodchild, and M. Robey. A Toolbox for Assessing Uncertainty in Spatial Databases. In *Proceedings of the AURISA '94 Conference (Sydney, Australia)*, 1994.
- [HHK15] Jeffrey Heer, Joseph M. Hellerstein, and Sean Kandel. Predictive Interaction for Data Transformation. In *CIDR 2015, Seventh Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 4-7, 2015, Online Proceedings*, 2015.
- [HLS⁺12] Clemens Holzhüter, Alexander Lex, Dieter Schmalstieg, Hans-Jörg Schulz, Heidrun Schumann, and Marc Streit. Visualizing Uncertainty in Biological Expression Data. volume 8294, pages 251–261, 2012.
- [HMSA08] J. Heer, J. Mackinlay, C. Stolte, and M. Agrawala. Graphical Histories for Visualization: Supporting Analysis, Communication, and Evaluation. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1189–1196, November 2008.
- [HP12] Rex Hartson and Pardha A. Pyla. *The UX Book: Process and Guidelines for Ensuring a Quality User Experience*. Morgan Kaufmann, 2012.

- [HQC⁺19] J. Hullman, X. Qiao, M. Correll, A. Kale, and M. Kay. In Pursuit of Error: A Survey of Uncertainty Visualization Evaluation. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):903–913, January 2019.
- [HRA15] Jessica Hullman, Paul Resnick, and Eytan Adar. Hypothetical Outcome Plots Outperform Error Bars and Violin Plots for Inferences about Reliability of Variable Ordering. *PLOS ONE*, 10(11):e0142444, November 2015.
- [HSN13] Zachary Hensley, Jibonananda Sanyal, and Joshua New. Provenance in Sensor Data Management. *Queue*, 11(12):50:50–50:63, December 2013.
- [IIC⁺13] Tobias Isenberg, Petra Isenberg, Jian Chen, Michael Sedlmair, and Torsten Möller. A systematic review on the practice of evaluating visualization. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2818–2827, December 2013.
- [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, February 2014.
- [KAF⁺08] Daniel Keim, Gennady Andrienko, Jean-Daniel Fekete, Carsten Görg, Jörn Kohlhammer, and Guy Melançon. Visual Analytics: Definition, Process, and Challenges. In Andreas Kerren, John T. Stasko, Jean-Daniel Fekete, and Chris North, editors, *Information Visualization*, number 4950 in Lecture Notes in Computer Science, pages 154–175. Springer Berlin Heidelberg, January 2008.
- [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, January 2002.
- [KEM06] Andreas Kerren, Achim Ebert, and Jörg Meyer. Introduction to Human-centered Visualization environments. In Andreas Kerren, Achim Ebert, and Jörg Meyer, editors, *Human-Centered Visualization Environments*, Lecture Notes in Computer Science, pages 1–9. Springer, 2006.
- [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, October 2011.

- [Kin10] R. Kincaid. SignalLens: Focus+Context Applied to Electronic Time Series. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):900–907, November 2010.
- [Kit95] Jenny Kitzinger. Qualitative Research: Introducing Focus Groups. *BMJ*, 311(7000):299–302, 1995.
- [KKEM10] Daniel Keim, Jörn Kohlhammer, Geoffrey Ellis, and Florian Mansmann. *Mastering the Information Age Solving Problems with Visual Analytics*. Eurographics Association, 2010.
- [KKUFW06] Olga A. Kulyk, Robert Kosara, Jaime Urquiza-Fuentes, and Ingo H. C. Wassink. Human-centered Aspects. In Andreas Kerren, Achim Ebert, and Jörg Meyer, editors, *Human-Centered Visualization Environments*, Lecture Notes in Computer Science, pages 13–75. Springer, 2006.
- [KMH06] G. Klein, B. Moon, and R. R. Hoffman. Making Sense of Sensemaking 2: A Macrocognitive Model. *IEEE Intelligent Systems*, 21(5):88–92, September 2006.
- [KMRS17] Christoph Kinkeldey, Alan M. MacEachren, Maria Riveiro, and Jochen Schiewe. Evaluating the Effect of Visually Represented Geodata Uncertainty on Decision-making: Systematic Review, Lessons Learned, and Recommendations. *Cartography and Geographic Information Science*, 44(1):1–21, January 2017.
- [KMS⁺08] Daniel A. Keim, Florian Mansmann, Jörn Schneidewind, Jim Thomas, and Hartmut Ziegler. Visual Analytics: Scope and Challenges. In Simeon J. Simoff, Michael H. Böhlen, and Arturas Mazeika, editors, *Visual Data Mining*, number 4404 in Lecture Notes in Computer Science, pages 76–90. Springer Berlin Heidelberg, January 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*, CHI ’11, pages 3363–3372, New York, NY, USA, 2011. ACM.
- [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*, AVI ’12, pages 547–554, New York, NY, USA, 2012. ACM.
- [KPS14a] Simone Kriglstein, Margit Pohl, and Michael Smuc. Pep Up Your Time Machine: Recommendations for the Design of Information Visualizations of Time-Dependent Data. In Weidong Huang, editor, *Handbook of Human Centric Visualization*, pages 203–225. Springer New York, 2014.

- [KPS⁺14b] Simone Kriglstein, Margit Pohl, Nikolaus Suchy, Johannes Gärtner, Theresia Gschwandtner, and Silvia Miksch. Experiences and Challenges with Evaluation Methods in Practice: A Case Study. In *Proceedings of the Fifth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization*, BELIV '14, pages 118–125. ACM, 2014.
- [KS01] George J. Klir and Richard M. Smith. On Measuring Uncertainty and Uncertainty-Based Information: Recent Developments. *Annals of Mathematics and Artificial Intelligence*, 32(1):5–33, August 2001.
- [KW13] Simone Kriglstein and Günter Wallner. Human Centered Design in Practice: A Case Study with the Ontology Visualization Tool Knoocks. In Gabriela Csurka, Martin Kraus, Leonid Mestetskiy, Paul Richard, and José Braz, editors, *Computer Vision, Imaging and Computer Graphics. Theory and Applications*, pages 123–141. Springer, 2013.
- [LAW⁺18] Shixia Liu, Gennady Andrienko, Yingcai Wu, Nan Cao, Liu Jiang, Conglei Shi, Yu-Shuen Wang, and Seokhee Hong. Steering Data Quality with Visual Analytics: The Complexity Challenge. *Visual Informatics*, 2(4):191–197, December 2018.
- [LBI⁺12] H. Lam, E. Bertini, P. Isenberg, C. Plaisant, and S. Carpendale. Empirical Studies in Information Visualization: Seven Scenarios. *IEEE Transactions on Visualization and Computer Graphics*, 18(9):1520–1536, September 2012.
- [LFR17] T. von Landesberger, D. W. Fellner, and R. A. Ruddle. Visualization System Requirements for Data Processing Pipeline Design and Optimization. *IEEE Transactions on Visualization and Computer Graphics*, 23(8):2028–2041, August 2017.
- [LFS⁺12] Jason Jingshi Li, Boi Faltings, Olga Saukh, David Hasenfratz, and Jan Beutel. Sensing the Air We Breathe: The Opensense Zurich Dataset. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, pages 323–325. AAAI Press, 2012.
- [LMK⁺15] L. Liu, M. Mirzargar, R.m. Kirby, R. Whitaker, and D. H. House. Visualizing Time-Specific Hurricane Predictions, with Uncertainty, from Storm Path Ensembles. *Computer Graphics Forum*, 34(3):371–380, June 2015.
- [LMW⁺17] S. Liu, D. Maljovec, B. Wang, P. T. Bremer, and V. Pascucci. Visualizing High-Dimensional Data: Advances in the Past Decade. *IEEE Transactions on Visualization and Computer Graphics*, 23(3):1249–1268, March 2017.
- [LPK10] Lin Li, Taoxin Peng, and Jessie Kennedy. Improving Data Quality in Data warehousing Applications. 2010.

- [LTM18] H. Lam, M. Tory, and T. Munzner. Bridging from Goals to Tasks with Design Study Analysis Reports. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):435–445, January 2018.
- [LWPL11] Jie Lu, Zhen Wen, Shimei Pan, and Jennifer Lai. Analytic Trails: Supporting Provenance, Collaboration, and Reuse for Visual Data Analysis by Business Users. In *Proc. of the 13th IFIP TC 13 Int. Conf. on HCI - Vol. IV*, INTERACT’11, pages 256–273, Berlin, Heidelberg, 2011.
- [MA14] Silvia Miksch and Wolfgang Aigner. A Matter of Time: Applying a Data-Users-Tasks Design Triangle to Visual Analytics of Time-Oriented Data. *Computers & Graphics, Special Section on Visual Analytics*, 38:286–290, 2014.
- [Mac86] Jock Mackinlay. Automating the Design of Graphical Presentations of Relational Information. *ACM Trans. Graph.*, 5(2):110–141, April 1986.
- [Mac92] Alan M. MacEachren. Visualizing Uncertain Information. *Cartographic Perspectives*, (13):10–19, June 1992.
- [Mac15] Alan M. MacEachren. Visual Analytics and Uncertainty: Its Not About the Data. In *EuroVis Workshop on Visual Analytics (EuroVA)*, pages 55–59, 2015.
- [MB02] Martin Maguire and Nigel Bevan. User Requirements Analysis: A Review of Supporting Methods. In *Proceedings of the IFIP 17th World Computer Congress - TC13 Stream on Usability: Gaining a Competitive Edge*, pages 133–148. Kluwer, B.V., 2002.
- [MCF⁺11] Luc Moreau, Ben Clifford, Juliana Freire, Joe Futrelle, Yolanda Gil, Paul Groth, Natalia Kwasnikowska, Simon Miles, Paolo Missier, Jim Myers, Beth Plale, Yogesh Simmhan, Eric Stephan, and Jan Van den Bussche. The Open Provenance Model core specification (v1.1). *Future Generation Computer Systems*, 27(6):743–756, 2011.
- [MF03] Heiko Müller and Johann-Christoph Freytag. Problems, Methods, and Challenges in Comprehensive Data Cleansing. Technical report, HUB-IB-164, Humboldt University Berlin, Berlin, 2003.
- [MFF⁺08] Luc Moreau, Juliana Freire, Joe Futrelle, Robert E. McGrath, Jim Myers, and Patrick Paulson. The Open Provenance Model: An Overview. In Juliana Freire, David Koop, and Luc Moreau, editors, *Provenance and Annotation of Data and Processes*, number 5272 in Lecture Notes in Computer Science, pages 323–326. Springer Berlin Heidelberg, January 2008.
- [MRC⁺07] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber. Documentation Mocap Database HDM05. Technical Report CG-2007-2, Universität Bonn, June 2007.

- [MRH⁺05] Alan M. MacEachren, Anthony Robinson, Susan Hopper, Steven Gardner, Robert Murray, Mark Gahegan, and Elisabeth Hetzler. Visualizing Geospatial Information Uncertainty: What We Know and What We Need to Know. *Cartography and Geographic Information Science*, 32(3):139–160, 2005.
- [MRO⁺12] A.M. MacEachren, R.E. Roth, J. O’Brien, B. Li, D. Swingley, and M. Gahegan. Visual Semiotics & Uncertainty Visualization: An Empirical Study. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2496–2505, December 2012.
- [MRSS⁺12] E. Maguire, P. Rocca-Serra, S. Sansone, J. Davies, and M. Chen. Taxonomy-Based Glyph Design—with a Case Study on Visualizing Workflows of Biological Experiments. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2603–2612, December 2012.
- [MS11] Peter Macko and Margo I. Seltzer. Provenance Map Orbiter: Interactive Exploration of Large Provenance Graphs. *Proceedings of the 3rd USENIX Workshop on the Theory and Practice of Provenance*, pages 1–6, 2011.
- [Mun09] T. Munzner. A Nested Model for Visualization Design and Validation. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):921–928, November 2009.
- [Mun14] Tamara Munzner. *Visualization Analysis and Design*. A.K. Peters Visualization Series. 2014.
- [NCE⁺11] Chris North, Remco Chang, Alex Endert, Wenwen Dou, Richard May, Bill Pike, and Glenn Fink. Analytic Provenance: Process+Interaction+Insight. In *CHI ’11 Extended Abstracts on Human Factors in Computing Systems*, CHI EA ’11, pages 33–36, New York, NY, USA, 2011. ACM. event-place: Vancouver, BC, Canada.
- [Nem63] P Nemenyi. *Distribution-free Multiple Comparisons*. Ph. D. thesis, Princeton University, 1963.
- [Nie94] Jakob Nielsen. Usability Inspection Methods. In Jakob Nielsen and Robert L. Mack, editors, *Usability Inspection Methods*, chapter Heuristic Evaluation, pages 25–62. Wiley & Sons, Inc., 1994.
- [Nor88] Donald A. Norman. *The Psychology of Everyday Things*. The Psychology of Everyday Things. Basic Books, New York, NY, US, 1988.
- [NXB⁺16] P. H. Nguyen, K. Xu, A. Bardill, B. Salman, K. Herd, and B. L. W. Wong. SenseMap: Supporting Browser-based Online Sensemaking Through Analytic Provenance. In *2016 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 91–100, October 2016.

- [NXW14] Phong H. Nguyen, Kai Xu, and B. L. William Wong. A Survey of Analytic Provenance. Technical report, Middlesex University, London, 2014.
- [Oĭ9] Devsoft Baltic OÜ. SurveyJS, 2019.
- [OJS⁺11] Daniela Oelke, Halldor Janetzko, Svenja Simon, Klaus Neuhaus, and Daniel A. Keim. Visual Boosting in Pixel-based Visualizations. *Computer Graphics Forum*, 30(3):871–880, June 2011.
- [OM02] C. Olston and J. D. Mackinlay. Visualizing Data with Bounded Uncertainty. In *IEEE Symposium on Information Visualization, 2002. INFOVIS 2002.*, pages 37–40, October 2002.
- [ORH05] Paulo Oliveira, Fátima Rodrigues, and Pedro Rangel Henriques. A Formal Definition of Data Quality Problems. In *IQ*, 2005.
- [PHKD06] Kristin Potter, Hans Hagen, Andreas Kerren, and Peter Dannenmann. Methods for Presenting Statistical Information: The Box Plot. *Visualization of large and unstructured data sets*, 4:97–106, 2006.
- [Pir05] Peter Pirolli. The Sensemaking Process and Leverage Points for Analyst Technology as Identified Through Cognitive Task Analysis. In *Proceedings of International Conference on Intelligence Analysis*, volume 5, pages 2–4, May 2005.
- [PKRJ10] K. Potter, J. Kniss, R. Riesenfeld, and C. R. Johnson. Visualizing Summary Statistics and Uncertainty. In *Proceedings of the 12th Eurographics / IEEE - VGTC Conference on Visualization*, EuroVis’10, pages 823–832, Chichester, UK, 2010. The Eurographs Association & John Wiley & Sons, Ltd. event-place: Bordeaux, France.
- [PLW02] Leo L. Pipino, Yang W. Lee, and Richard Y. Wang. Data Quality Assessment. *Commun. ACM*, 45(4):211–218, April 2002.
- [PRJ12] Kristin Potter, Paul Rosen, and Chris R. Johnson. From Quantification to Visualization: A Taxonomy of Uncertainty Visualization Approaches. *IFIP advances in information and communication technology*, 377:226–249, 2012.
- [PS96] Richard A. Powell and Helen M. Single. Focus groups. *International Journal for Quality in Health Care*, 8(5):499–504, 1996.
- [PWB⁺09] K. Potter, A. Wilson, P. T. Bremer, D. Williams, C. Doutriaux, V. Pascucci, and C. R. Johnson. Ensemble-Vis: A Framework for the Statistical Visualization of Ensemble Data. In *2009 IEEE International Conference on Data Mining Workshops*, pages 233–240, December 2009.
- [R C19] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2019.

- [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*, CHI '94, pages 318–322, New York, NY, USA, 1994. ACM.
- [RD00] Erhard Rahm and Hong Hai Do. Data Cleaning: Problems and Current Approaches. *IEEE Data Eng. Bull.*, 23(4):3–13, 2000.
- [Red98] Thomas C. Redman. The Impact of Poor Data Quality on the Typical Enterprise. *Commun. ACM*, 41(2):79–82, February 1998.
- [Red12] Thomas C. Redman. Data Quality Management Past, Present, and Future: Towards a Management System for Data. In Shazia Sadiq, editor, *Handbook of Data Quality*, pages 15–40. Springer Berlin Heidelberg, 2012.
- [RESC16] E. D. Ragan, A. Endert, J. Sanyal, and J. Chen. Characterizing Provenance in Visualization and Data Analysis: An Organizational Framework of Provenance Types and Purposes. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):31–40, January 2016.
- [RGT15] Eric D. Ragan, John R. Goodall, and Albert Tung. Evaluating How Level of Detail of Visual History Affects Process Memory. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 2711–2720, New York, NY, USA, 2015. ACM.
- [RH01] Vijayshankar Raman and Joseph M. Hellerstein. Potter’s Wheel: An Interactive Data Cleaning System. In *Proceedings of the 27th International Conference on Very Large Data Bases*, VLDB '01, pages 381–390, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.
- [RLK⁺15] Martin Röhlig, Martin Luboschik, Frank Kruger, Thomas Kirste, Heidrun Schumann, Markus Bögl, Bilal Alsallakh, and Silvia Miksch. Supporting Activity Recognition by Visual Analytics. In *2015 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 41–48, Chicago, IL, USA, October 2015. IEEE.
- [RLKL⁺12] Norel Rimbu, Gerrit Lohmann, Gert König-Langlo, Cristian Necula, and Monica Ionita. *30 Years of Synoptic Observations from Neumayer Station with Links to Datasets*. PANGAEA, 2012.
- [Sad13] Shazia Sadiq. Research and Practice in Data Quality Management. In *Handbook of Data Quality*. Springer Verlag, Berlin, Heidelberg, 2013.
- [SBFK16] Dominik Sacha, Ina Boesecke, Johannes Fuchs, and Daniel A. Keim. *Analytic Behavior and Trust Building in Visual Analytics*. 2016.

- [Sch12] Margrit Schreier. *Qualitative Content Analysis in Practice*. SAGE Publications, 2012.
- [SEG05] Rajmonda Sulo, Stephen Eick, and Robert Grossman. DaVis: A Tool for Visualizing Data Quality. *Posters Compendium of InfoVis*, 2005:45–46, 2005.
- [SFTM⁺13] Awalın Sopen, Manuel Freire, Meirav Taieb-Maimon, Catherine Plaisant, Jennifer Golbeck, and Ben Shneiderman. Exploring Data Distributions: Visual Design and Evaluation. *Int. J. Hum. Comput. Interaction*, 29(2):77–95, 2013.
- [SGP⁺18] H. Stitz, S. Gratzl, H. Piringer, T. Zichner, and M. Streit. Knowledge-Pearls: Provenance-Based Visualization Retrieval. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–1, 2018.
- [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.
- [SLSG16] H. Stitz, S. Luger, M. Streit, and N. Gehlenborg. AVOCADO: Visualization of Workflow-Derived Data Provenance for Reproducible Biomedical Research. *Computer Graphics Forum*, 35(3):481–490, 2016.
- [SMM12] Michael Sedlmair, Miriah Meyer, and Tamara Munzner. Design Study Methodology: Reflections from the Trenches and the Stacks. *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis)*, 18(12):2431–2440, 2012.
- [SNHS17] Hans-Jörg Schulz, Thomas Nocke, Magnus Heitzler, and Heidrun Schumann. A Systematic View on Data Descriptors for the Visual Analysis of Tabular Data. *Information Visualization*, 16(3):232–256, July 2017.
- [SPG05] Yogesh L. Simmhan, Beth Plale, and Dennis Gannon. A Survey of Data Provenance Techniques. *Computer Science Department, Indiana University, Bloomington IN*, 47405, 2005.
- [SS17] Andreas Schreiber and Regina Struminski. Visualizing Provenance Using Comics. In *9th USENIX Workshop on the Theory and Practice of Provenance (TaPP 2017)*, Seattle, WA, jun 2017. USENIX Association.
- [SS18] H. Song and D. A. Szafrir. Where’s My Data? Evaluating Visualizations with Missing Data. *IEEE Transactions on Visualization and Computer Graphics*, pages 914–924, 2018.
- [SSK⁺16] D. Sacha, H. Senaratne, B. C. Kwon, G. Ellis, and D. A. Keim. The Role of Uncertainty, Awareness, and Trust in Visual Analytics. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):240–249, January 2016.

- [SSS⁺14] D. Sacha, A. Stoffel, F. Stoffel, B. C. Kwon, G. Ellis, and D. A. Keim. Knowledge Generation Model for Visual Analytics. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1604–1613, December 2014.
- [TC05] James J. Thomas and Kristin A. Cook. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE Computer Society Press, 2005.
- [TC06] James J. Thomas and Kristin A. Cook. A Visual Analytics Agenda. *Computer Graphics and Applications, IEEE*, 26(1):10–13, 2006.
- [TFB⁺14] Alvin Tarrell, Ann Fruhling, Rita Borgo, Camilla Forsell, Georges Grinstein, and Jean Scholtz. Toward Visualization-specific Heuristic Evaluation. In *Proceedings of the Fifth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization*, BELIV '14, pages 110–117. ACM, 2014.
- [TGK⁺17] C. Tominski, S. Gladisch, U. Kister, R. Dachselt, and H. Schumann. Interactive Lenses for Visualization: An Extended Survey. *Comput. Graph. Forum*, 36(6):173–200, September 2017.
- [THM⁺05] Judi Thomson, Elizabeth Hetzler, Alan MacEachren, Mark Gahegan, and Misha Pavel. A Typology for Visualizing Uncertainty. volume 5669, pages 146–157, March 2005.
- [TM04] Melanie Tory and Torsten Möller. Human Factors in Visualization Research. *IEEE Transactions on Visualization and Computer Graphics*, 10(1):72–84, 2004.
- [TZHH18] Jhon Alejandro Triana, Dirk Zeckzer, Hans Hagen, and Jose Tiberio Hernandez. VafusQ: A Methodology to Build Visual Analysis Applications with Data Quality Features. *Information Visualization*, 18(4):384–404, December 2018.
- [WBFL17] M. Wunderlich, K. Ballweg, G. Fuchs, and T. von Landesberger. Visualization of Delay Uncertainty and its Impact on Train Trip Planning: A Design Study. *Computer Graphics Forum*, 36(3):317–328, June 2017.
- [Wel10] Stefan Wellek. *Testing Statistical Hypotheses of Equivalence and Noninferiority*. Chapman and Hall/CRC, 2010.
- [Wij06] J. J. van Wijk. Views on Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):421–432, July 2006.
- [WN11] Esteban Walker and Amy S. Nowacki. Understanding Equivalence and Noninferiority Testing. *Journal of General Internal Medicine*, 26(2):192–196, February 2011.

- [WS96] Richard Y. Wang and Diane M. Strong. Beyond Accuracy: What Data Quality Means to Data Consumers. *J. Manage. Inf. Syst.*, 12(4):5–33, March 1996.
- [WSD⁺13] Rick Walker, Aiden Slingsby, Jason Dykes, Kai Xu, Jo Wood, Phong H. Nguyen, Derek Stephens, B. L. William Wong, and Yongjun Zheng. An Extensible Framework for Provenance in Human Terrain Visual Analytics. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2139–2148, 2013.
- [WXYR11] Matthew Ward, Zaixian Xie, Di Yang, and Elke Rundensteiner. Quality-aware visual Data Analysis. *Computational Statistics*, 26(4):567–584, December 2011.
- [WYM12] Yingcai Wu, Guo-Xun Yuan, and Kwan-Liu Ma. Visualizing Flow of Uncertainty Through Analytical Processes. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2526–2535, 2012.
- [XHWR06] Z. Xie, S. Huang, M. O. Ward, and E. A. Rundensteiner. Exploratory Visualization of Multivariate Data with Variable Quality. In *2006 IEEE Symposium On Visual Analytics Science And Technology*, pages 183–190, October 2006.
- [XWRH07] Z. Xie, M. O. Ward, E. A. Rundensteiner, and S. Huang. Integrating Data and Quality Space Interactions in Exploratory Visualizations. In *Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization (CMV 2007)*, pages 47–60, July 2007.
- [ZC07] Torre Zuk and Sheelagh Carpendale. Visualization of Uncertainty and Reasoning. In Andreas Butz, Brian Fisher, Antonio Krüger, Patrick Olivier, and Shigeru Owada, editors, *Smart Graphics*, Lecture Notes in Computer Science, pages 164–177. Springer Berlin Heidelberg, 2007.
- [ZCPB11] J. Zhao, F. Chevalier, E. Pietriga, and R. Balakrishnan. Exploratory Analysis of Time-Series with ChronoLenses. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2422–2431, December 2011.
- [ZSN⁺06] Torre Zuk, Lothar Schlesier, Petra Neumann, Mark S. Hancock, and Sheelagh Carpendale. Heuristics for Information Visualization Evaluation. In *Proceedings of the 2006 AVI Workshop on BEyond Time and Errors: Novel Evaluation Methods for Information Visualization*, BELIV '06, pages 1–6. ACM, 2006.