Visual Interactive Creation, Customization, and Analysis of Data Quality Metrics

CHRISTIAN BORS, THERESIA GSCHWANDTNER, SIMONE KRIGLSTEIN, SILVIA MIKSCH, and MARGIT POHL, Institute of Visual Computing & Human-Centered Technology, TU Wien

3

During data preprocessing, analysts spend a significant part of their time and effort profiling the quality of the data along with cleansing and transforming the data for further analysis. While quality metrics—ranging from general to domain-specific measures—support assessment of the quality of a dataset, there are hardly any approaches to visually support the analyst in customizing and applying such metrics. Yet, visual approaches could facilitate users' involvement in data quality assessment. We present *MetricDoc*, an interactive environment for assessing data quality that provides customizable, reusable quality metrics in combination with immediate visual feedback. Moreover, we provide an overview visualization of these quality metrics along with error visualizations that facilitate interactive navigation of the data to determine the causes of quality issues present in the data. In this article, we describe the architecture, design, and evaluation of *MetricDoc*, which underwent several design cycles, including heuristic evaluation and expert reviews as well as a focus group with data quality, human-computer interaction, and visual analytics experts.

CCS Concepts: • **General and reference** → **Metrics**; *Evaluation*; *Design*; • **Human-centered computing** → **Visual analytics**; *User studies*; • **Information systems** → *Data cleaning*;

Additional Key Words and Phrases: Data profiling, data quality metrics, visual exploration

ACM Reference format:
Christian Bors, Theresia Gschwandtner, Simone Kriglstein, Silvia Miksch, and Margit Pohl. 2018. Visual Interactive Creation, Customization, and Analysis of Data Quality Metrics. *J. Data and Information Quality* 10, 1, Article 3 (May 2018), 26 pages.
https://doi.org/10.1145/3190578

1 INTRODUCTION

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. The 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 a government

This work was supported by the Centre for Visual Analytics Science and Technology (CVAST), funded by the Austrian Federal Ministry of Science, Research, and Economy in the exceptional Laura Bassi Centres of Excellence initiative (#822746).
Authors' addresses: C. Bors, T. Gschwandtner, and S. Miksch, Institute of Visual Computing & Human-Centered Technology, TU Wien, Favoritenstrasse 9-11/193-7, 1040 Vienna, Austria; emails: {christian.bors, theresia.gschwandtner, silvia.miksch}@tuwien.ac.at; S. Kriglstein and M. Pohl, Institute of Visual Computing & Human-Centered Technology, TU Wien, Argentinierstrasse 8/Floor 2, 1040 Vienna, Austria; emails: {simone.kriglstein, margit.pohl}@tuwien.ac.at.

https://doi.org/10.1145/3190578

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 either to quickly evaluate datasets that are 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 [24] to get a sense of the data distribution and anomalies. Summary visualizations lack flexibility to accentuate different aspects of data quality. We argue that automatically computed quality metrics can facilitate quality assessment and expedite validation. These data quality metrics are commonly used in data warehousing and data quality research. They present powerful means for determining the overall quality of a dataset, as well as for defining, measuring, and managing the quality of information and data [10]. In contrast to isolated quality checks, quality metrics can be used to validate a variety of data characteristics and different properties simultaneously. However, generalized measures are often not sufficient for determining quality issues specific to a certain data domain. Context- and domain-specific characteristics of a dataset, along with intrinsic structures in the data, require proper adaption and customization of these metrics. To support analysts in effectively adapting data quality metrics, they need to be able (1) to customize quality metrics interactively to specific datasets and domains and (2) to assess quality easily and quickly.

In this article, we propose a visual analytics (VA) approach that enables data analysts to utilize and customize quality metrics in order to facilitate quality assessment of their specific datasets. We developed an environment—*MetricDoc* (*Metric D*ata *O*verview and *C*ustomization)—in which users can interactively customize data quality metrics for tabular datasets and visually explore the results of these metrics. The computed metric information provides comprehensive quality information for quick overview validation while simultaneously providing detailed information about potential dirty data entries, error types, and the distribution of such across the dataset. Our main contributions are:

- We provide a VA solution for the interactive **creation and customization of data quality metrics** with immediate visual feedback.
- We introduce a set of predefined **generic quality checks,** which serves as a starting point for creating context- and domain-specific quality metrics.
- We **visualize the overall data quality** according to the created quality metrics (Section 4.3.1).
- We design interactive visualizations for the **navigation and exploration** of quality issues and their distribution within the dataset (Section 4.3.3 and 4.3.2).
- We employ **iterative design and qualitative evaluation** to develop our VA application (Section 5).
- Our environment is **publicly available as an open-source project**,[1] an extension of the *OpenRefine* wrangling and cleansing software.

The remaining article discusses related work and problem analysis, from which we derived our requirements in Section 3. Our approach is described in Section 4, followed by an elaboration of the pursued design process in Section 5, a *case study* in Section 6, and the discussion in Section 7. Finally, conclusions and further research directions are given in Section 8.

2 RELATED WORK

Interactive data quality analysis is an iterative process that deeply intertwines with the tasks of data profiling, wrangling, and cleansing. Data wrangling can be understood as the process of iterative data exploration and transformation, while data cleansing deals with the actual correction

[1] https://github.com/christianbors/OpenRefineQualityMetrics.

of erroneous data. Data profiling provides means for assessing the quality of a dataset. Moreover, there are numerous taxonomies of data quality issues and classifications of error sources [1, 3, 18, 27, 35, 38] that give systematic overviews. Most taxonomies serve as comprehensive bodies of knowledge without providing recommendations on how to actually detect or resolve quality issues within a dataset. Yet others also outline successful strategies to manage data quality, architectural requirements, efficient computational tools and techniques, or case studies of data quality initiatives [13, 43, 45]. Nevertheless, there are few to none ready-to-use tools, and most data quality analysts tackle the task of analyzing the quality of complex datasets with simple command line tools and scripting libraries, allowing them to batch-transform the dataset. Established environments (e.g., Microsoft Excel, R [42]) allow raw data exploration, but they are not supporting users with contextually appropriate visualizations. Thus, users are required to develop their own visual representations based on contextual and domain knowledge.

Current VA approaches for data quality (e.g., Wrangler [23] and Profiler [24]) present sophisticated probing and overview features paired with transformation and reformatting techniques. Some tools provide interactive visualizations for data profiling, such as histograms [39, 51] and scatterplots [39] to represent the structure of the data. Others provide visualizations for more specific data characteristics, such as the chronological sequence of time-series data (e.g., heatmaps [17], timeline charts [4]), or spatial data [7] to aid the identification of implausible values and, consequently, of possibly erroneous data entries. Kandel et al. [24] automatically calculate suitable visual representations to effectively represent different data characteristics. They provide interactive and linked summary visualizations to facilitate the detection of errors. Heer et al. [20] present data-driven algorithms that aim at providing a predictive interaction model for data transformations, while still retaining user guidance. We can derive from this publication that data mining and information retrieval approaches that compute automatic suggestions based on contextual meta-information facilitate quality assessment for the user. We have derived from this that contextual information and domain knowledge should be integrated into the data profiling process to make the visualizations and other quantitative representations more expressive and appropriate to their use.

Interactive visualization techniques employed in profiling and wrangling applications primarily act as navigation aids for exploring the dataset or are used for contextual filtering of the data [24, 39], while distribution overview visualizations implicitly indicate possible data quality problems if outliers persist in the data. They aim at leveraging users' ability to quickly assess provided information and effectively explore datasets. Thus, we chose overview visualizations as a starting point and provide interactive means to dive deeper into the data.

Other approaches employ linking and brushing [25] to guide the user into regions of interest. This particularly poses a challenge in large datasets and demands for efficient implementation. Keim [25] emphasizes the importance of applying filters on large datasets through either interactive selection or querying in order to determine a desired subset of data. For raw data representations, simple tabular data views are commonly used, while alternative data overview visualizations, like TableLens [44] or SparkLines [52], are underrepresented due to their limitation to only encode numeric data effectively. Sopan et al. [48] present column overviews to encode high-dimensional multivariate distribution data. Depending on the specific task at hand, multiple view representations are employed to facilitate error detection, e.g., highlighting erroneous data entries in a data cleansing scenario [17, 23] or linking and brushing data entries for profiling and wrangling tasks [20, 23, 24]. For both large datasets and data with complex structures, it is beneficial to provide linking and brushing to encourage exploration. So far no task-oriented visualization approach has provided a visual overview of data quality invariant of data types and dataset size. We try to fill this blank spot by providing a visual overview of data quality for means of dataset validation and exploration of quality issues.

To provide such an overview, we plan to employ data quality metrics in order to give consistent and invariant measures of quality. Existing data quality approaches (e.g., [7, 17, 23, 24]) utilize selected metrics such as completeness, validity of data, correctness, and uniqueness. These metrics may consist of one or more quality checks and allow for a more systematic overview of which quality dimensions may be impacted within a given dataset. Batini et al. [2] and Pipino et al. [40] categorize data quality projects and data quality metrics and provide suggestions on metric implementations. We identified common drawbacks among these data quality approaches. On the one hand, most provide predefined quality checks for the most basic error types, with possibilities to parameterize these checks to meet more domain-specific requirements. However, users need to have a very clear idea of the required checks to accomplish this. These tools do not provide means for assessing the effects of the parameterization or their interrelations with other checks applied on the dataset. On the other hand, these tools lack means for effectively grouping these numerous checks into different dimensions for making them easily reusable.

Kandel et al. [22] present future research directions and challenges in data wrangling, discussing challenges faced by data quality analysts. They outline that quite often the domain knowledge of an expert is needed to define efficient quality checks, and they propose employing automated processes in combination with visual interfaces for manual wrangling tasks. Systems that provide contextual information from previous data profiling, wrangling, and cleansing projects could significantly benefit users during their data quality analysis tasks.

Based on the shortcomings of previous approaches and data quality use cases, we elaborated on the requirements of *MetricDoc* in the following section.

3 REQUIREMENTS

Before starting the development and design of *MetricDoc*, we recapitulated requirements that should be met by our approach. We derived these requirements from (1) literature research and identified shortcomings in other data quality projects (e.g., [22]), (2) our long-lasting experience with VA data quality projects [17, 18, 29, 30], and (3) our collaborations with various company partners, in multiple discussions with the target users of such a system, i.e., data analysts dealing with data quality. Human-computer interaction (HCI) experts were actively involved in the design process (see Section 6), giving feedback about requirements regarding visual elements and general user experience. Moreover, we derived requirement specifications from two user stories for data quality assessment: (1) a data quality expert being assigned to validate the quality and usability of an unknown dataset and (2) a data quality expert receiving an updated dataset that needs revalidation.

Moreover, Miksch and Aigner's [34] design principle of data, users, and tasks was pursued:
The *users* are data quality analysts with expertise in data profiling and comprehensive knowledge in their respective working domains.

The *data* consist of a tabular dataset subject to analysis, with quantitative, qualitative, and time-oriented data supported for analysis.

The *tasks* for assessing data quality are split into:

- T1. performing a first assessment of the quality of a dataset (using general quality metrics),
- T2. adding custom quality checks and customizing parameters of quality 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 particular subsequent analysis task.

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

R1: Customizable Quality Metrics. Data quality metrics should appropriately reflect the quality of the data at hand. To accomplish this, users should be able to adapt quality 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, we have determined design rationales (cf. Section 4.1) 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 METRICDOC: A VISUAL EXPLORATION ENVIRONMENT OF DATA QUALITY METRICS

In this section, we describe the design and architecture of *MetricDoc*, an environment to compose, customize, and visually explore data quality metrics (an overview of the environment can be seen in Figure 1). Our environment is based on the server backend of the open-source data wrangling tool *OpenRefine* [39]. We extended *OpenRefine*'s data structure to store metainformation about the tabular dataset and leveraged the integrated General Refine Expression Language (GREL) [39] for the development of our quality metrics. The implementation, however, is aimed to be standalone and could be integrable into other data profiling or wrangling environments quickly, which allows users to utilize this extension in their preferred environment with only minor adaptations to any given data structure. As the central element of our environment we employ data quality metrics, which communicate both the overall dirtiness and detailed error information of the data. Quality metrics are implemented in a modular way, which allows swift algorithm modification and, more importantly, potential expansion of functionality.

Architecture. A schematic overview of the environment's architecture can be found in Figure 2, illustrating the interconnection of different data models, representations as well as interactions. The data structures in *MetricDoc* extend OpenRefine's [39] column data representation, adding data quality information in the form of quality metrics. Similarly, the data structures of other data profiling or wrangling tools can be extended, regardless of the storage approach employed (e.g., column-wise, row-wise, tuple-wise). Additionally, server-side data quality operations (calculations, setup procedures, etc.) ensure proper project persistence and data management. Leveraging the underlying data structures and operations, the web-based exploration environment features interactive visualizations (built in *D3* [5]) and raw data views using DataTables [11].

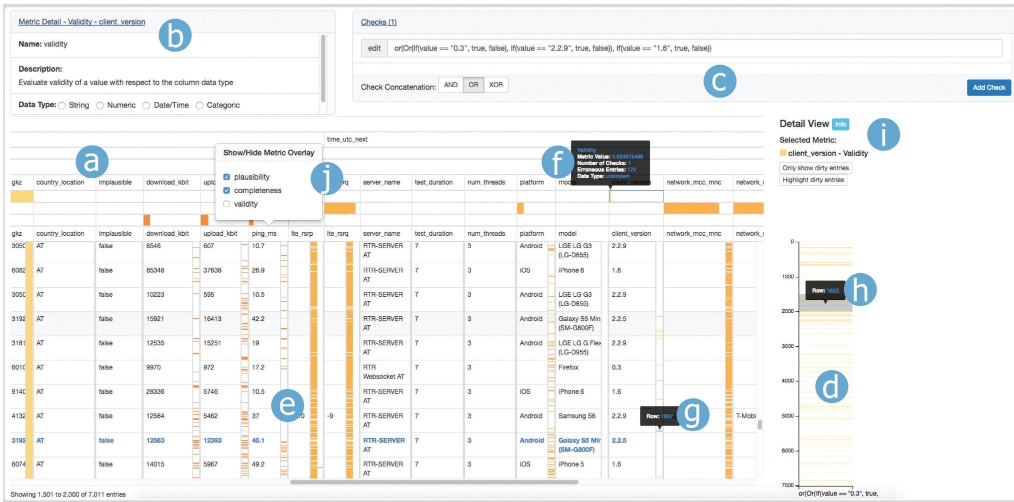

Fig. 1. MetricDoc: An interactive visual exploration environment for assessing data quality (this is a composed view, which shows multiple popups and tooltips at the same time). The environment consists of the *quality metrics overview* (a), the metric information view (b) and customization tabs (c), the *metric detail view* (d), and the tabular *raw data view* enhanced with *error distribution* heatmaps (e). Mouseover tooltips provide detailed information on metrics (f) and data errors (g,h); metric distribution heatmaps can be enabled and disabled individually (j). Case study (see Section 6) 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.

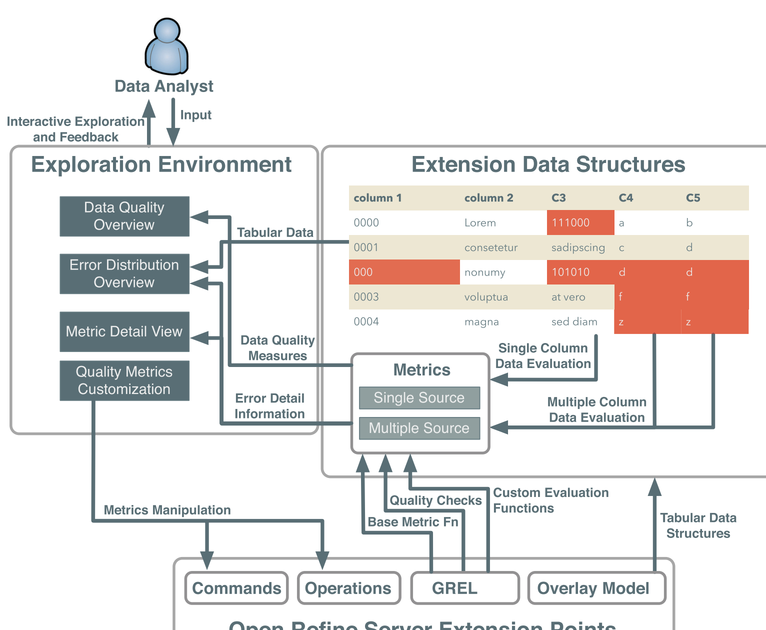

Fig. 2. The *MetricDoc* environment architecture. It builds upon OpenRefine [39] to apply quality metrics on tabular data and provides an interactive exploration environment on the front end. The red cells in the data table illustrate errors in the data.

4.1 Design Rationales

The design on *MetricDoc* should consist of a tabular data representation enhanced by visual elements for presentation and navigation of the dataset based on data quality 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 data quality and preprocessing—rely on scripting and textual interfaces for profiling data and developing quality metrics, and 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. [47], 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 [17, 23, 24], as well as tabular-like overview visualization techniques [44, 48] with orientation toward interactive exploration [25]. Accordingly, we distilled the following design rationales based on the requirements we defined in Section 3.

D1: Providing Consistent, Informative Visual Encodings [R2–R4]**:** Due to the potentially large scale of the data, 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. However, we want to avoid too many different visualization types to keep learning demands 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–R4]**:** Users' data analysis workflows and tasks may differ considerably, requiring access to different data aspects and visual representations, including data quality information. Data quality 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 detailed information about the dirtiness in the data. Through brushing and linking [36] across visualizations and data views, we give various error information to support the quality assessment tasks. Along with this design, we leverage effective exploration techniques on different granularity levels to support a quick identification of quality issues throughout the dataset, by inferring location information and contextual information on surrounding data.

D3: Interactively Supporting Quality Metric Customization [R1]**:** Quality metrics are potentially complex measures (cf. Section 4.2) and require domain-specific adaptations [10]. 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 quality 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]**:** Data quality 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 data quality, and developing quality 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, we intend to incorporate visual encodings that quickly communicate where investigation is required, i.e., 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 data quality problems, while the absence of such visual cues signifies high data quality and no need for intervention.

With these design rationales defined, we proceeded with prototyping the *MetricDoc* environment. As described in Section 5, visual encodings and interactions were subject to change during iteration cycles, with the core elements left widely unchanged.

4.2 Quality Metrics

Derived from existing definitions of quality metrics [2, 40, 45] and generic data quality models [12], we define a quality metric as the quantified measure of a data quality dimension that gives proportional information about the lack of quality regarding a certain information aspect.

The quality of a dataset with respect to a metric Q_m is quantified as the inverted ratio of the number of determined dirty entries $||D_m|| = \sum_{i=0}^{n} d_i$ for $d_i \in [0,1]$, the total data entry count $||N_{col}||$, and the number of validation functions k:

$$Q_m(col) = 1 - \frac{\sum_{i=0}^{k} \sum_{n=0}^{N_{col}} \mathsf{VF}_i(v_n)}{||N_{col}||} \in [0,1].$$

The dirtiness of each entry is determined by the measurement of data quality dimensions through validation function calls $\mathtt{vf}_m$ (in our practical case GREL functions), specifically implemented for the respective metric, as a `boolean` value:

$$\mathsf{VF}_{\text{column}}(v) = \bigcup_{m=0}^{M} \mathtt{vf}_m(v), \mathsf{VF}_{\text{spanning}}(v_0, v_1) = \bigcup_{m=0}^{M} \mathtt{vf}_m(v_0, v_1).$$

Each metric retains annotation information on which entries or tuples have been evaluated as dirty (see R3). As of this point, we distinguish between single- and multiple-column metrics (later referred to as spanning metrics) and evaluate dirtiness either row-wise or distribution dependent. That way we can determine both explicit and implicit errors—errors that require data transformation to be made explicit. For both metric types, different visual encodings need to be considered for the overview visualization (see Section 4.3.1). Metrics are extendable by custom quality checks to account for domain-specific aspects (format requirements, upper or lower measurement boundaries, etc.) in measuring quality. This modular approach supports the development of sophisticated metrics and formal validation techniques to increase contextual expressiveness.

Quality Checks. In analogy to our definition of a metric function, a quality check evaluates an entry or tuple toward dirtiness, returning a `boolean value`. Such checks can be validated either row by row or depending on distribution. Users can develop expressive validation schemata to accomplish versatile and useful checks for detecting dirty entries. Users can annotate quality checks to give contextual information for other users or for further reference. Multiple checks can be concatenated with logical expressions (`AND`, `OR`), extending the flexibility to represent dependencies adequately.

Available Metrics. Metrics measure specific quality dimensions. We have implemented the following quality metrics in *MetricDoc*, which constitute a set of widely applicable measures for profiling raw tabular data. The variety of metrics that have initially been made available for analysts to adapt and customize was selected based on different factors: (1) collaborator's familiarity and preferences of quality metrics based on prior work, (2) popularity of metrics in related literature, and (3) assessment of usefulness and generalizability of the metrics, along with potential to extend functionality.

QM1: Completeness. The completeness of a dataset is a commonly used quality metric employed in several data profiling and wrangling applications [1, 17, 23], mainly referring to missing values. There are different types of measures defined in the literature to determine a missing value [3]. The employed default implementation computes *column completeness* (or *attribute completeness* [3]) and measures missing values within single columns: a column entry $v_{\text{col,row}}$ is identified dirty if it is either missing or marked as empty (through a particular identifier, e.g., `NaN` in R or Matlab). More sophisticated completeness validation can be modeled by constructing additional quality checks.

$$Q_{comp}(v_{\text{col,row}}) = \begin{cases} \textbf{false} & \text{if } v_{\text{col,row}} = \texttt{null} \text{ or } v_{\text{col,row}} \in \{\text{NaN, -, } \ldots\} \\ \textbf{true} & \text{else} \end{cases}$$

QM2: Validity. Validating data is a crucial part of analysis, since invalid entries might impede calculations or skew statistical evaluations. The reasons for data being invalid are highly diverse and context dependent [1]. Identifying values as invalid is a task that demands comprehensive domain knowledge from the user. The default validity metric includes a check to evaluate if a data entry complies with the automatically detected or manually specified data type of the column. Domain-specific validity characteristics can be incrementally added and refined, depending on the analyst's prior knowledge of the dataset.

$$Q_{valid}(v_{\text{col,row}}, type) = \begin{cases} \textbf{true} & \text{if typeof}(v_{\text{col,row}}) = type, \text{ for } type \in \{numeric, string, date, \ldots\} \\ \textbf{false} & \text{else} \end{cases}$$

QM3: Plausibility. Data analysts utilize statistics measures to gain distribution information about numeric column data in datasets and subsequently get insights of valid, implausible, and extreme entries. Such entries might manifest in datasets for instance due to erroneous data generation (e.g., human-created values) or inconsistent sources (e.g., different sensor calibration) [18]. The plausibility metric detects outlying entries by using either nonrobust (statistical mean $\overline{x}_{col}$ and standard deviation $std(X_{col})$) or robust statistics measures (median $\tilde{x}_{col}$ and a robust interquartile range estimator $s_{\texttt{IQR}} = \frac{\text{IQR}}{1.35}$) to help analysts with finding extreme entries.

$$Q_{plaus}(v_{\text{col,row}})_{\text{standard}} = \begin{cases} \textbf{true} & \text{if } (\overline{x}_{\text{col}} - 2 * std(X_{\text{col}})) < v_{\text{col,row}} < (\overline{x}_{\text{col}} + 2 * std(X_{\text{col}})) \\ \textbf{false} & \text{else} \end{cases}$$

$$Q_{plaus}(v_{\text{col,row}})_{\text{robust}} = \begin{cases} \textbf{true} & \text{if } (\tilde{x}_{\text{col}} - 2 * \mathtt{s}_{\mathtt{IQR}}) < v_{\text{col,row}} < (\tilde{x}_{\text{col}} + 2 * \mathtt{s}_{\mathtt{IQR}}) \\ \textbf{false} & \text{else} \end{cases}$$

QM4: 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 $v_{\text{col}_b,\text{row}} - v_{\text{col}_a,\text{row}}$ 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.

$$Q_{interval}(v_{\text{col}_a,\text{row}}, v_{\text{col}_b,\text{row}}, d, \triangleright) = \begin{cases} \textbf{true} & \text{if } (v_{\text{col}_b,\text{row}} - v_{\text{col}_a,\text{row}}) \triangleright d, \text{ for } \triangleright \in \{<, \leq, >, \geq, =\} \\ \textbf{false} & \text{else} \end{cases}$$

QM5: Uniqueness. The user can specify one or more columns that are expected to contain a unique combination of entries to check the dataset for duplicate entries.

$$Q_{unique}(\text{col}_m, \ldots, \text{col}_y) = \begin{cases} \textbf{true} & \text{if } \forall x \in M : M(x) = 1, \text{ for } M = \{\{x_i | x_i = (v_{\text{col}_{m,i}}, \ldots, v_{\text{col}_{y,i}}) \text{ for } i = 1 \ldots n\}\} \\ \textbf{false} & \text{else} \end{cases}$$

We have outlined the metrics employed to allow quality validation for various potential error sources. In the upcoming section, we describe *MetricDoc*, the environment that lets users build, customize, and leverage these quality metrics for data profiling.

4.3 The Visual Exploration Environment

MetricDoc's web user interface provides a visual exploration environment that features both a raw dataset representation and an overview of quality metrics along with a representation of the distribution of dirty data entries within the dataset (see Figure 1). Users can manage the deployment of quality metrics and corresponding quality checks on datasets. We put an emphasis on visual support for dirty data exploration as well as visual feedback during metric customization. In the following section, we will elaborate on the visual encodings we employed to provide an easily comprehensible exploration and metric customization for facilitated assessment.

4.3.1 Quality Metrics Overview (c.f. Figure 1(a)). The metrics overview (see Figure 3) is one of the main components in *MetricDoc* (see Figure 1). For single-source metrics, the representation resembles a tabular structure, column by column indicating a data quality summary, while rows in this table correspond to different quality metrics. The tabular representation aims at inducing a relation to columns in the original data table by aligning the *metrics overview* with the tabular representation of the original data that is positioned directly below. For each metric and each column, we indicate the amount of identified dirty entries by an error bar. Spanning metrics correspond to multiple source columns and implicit information cannot be deducted from one singular column. Hence, for these metrics we omitted positional relations with the *raw data view* and instead label the columns to indicate which are evaluated by means of such a spanning metric. 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, we satisfy R2 and keep consistent with D1, by informing the user about the general dirtiness of a dataset and providing an overview of any available quality metrics.

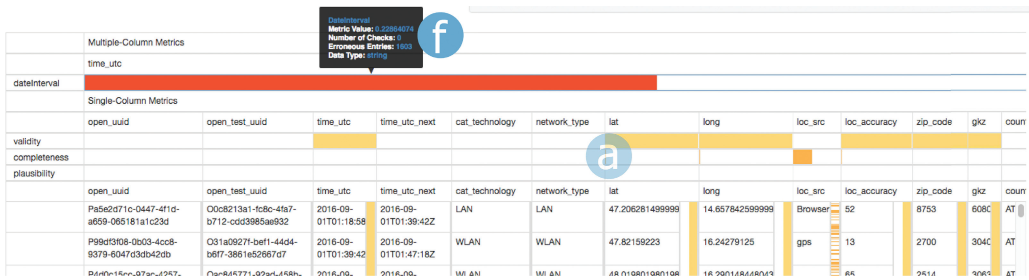

Fig. 3. *Quality Metrics Overview* (a), including metric detail within a mouseover tooltip (f). Single-Column and Multiple-Column Metrics are visually separated to emphasize information disparity (cf. Figure 1a,f).

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 toward 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., (f) in Figure 1). Upon selecting one or multiple quality metrics, the *metric detail view* shows information for further inspection.

4.3.2 Metric Detail View (c.f. Figure 1(d)). To represent detailed quality metric information, we developed a schematic error view (see Figure 4) 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. [48]. 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 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 display either 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 (c.f. Figure 1(e)). In addition to the heatmap-like overview of error distributions given in the *metric detail view* (see Figure 4(d)), we provide heatmap-like elements within the raw data table to meet D1 (see Figure 1(e)). We enhanced each column of the

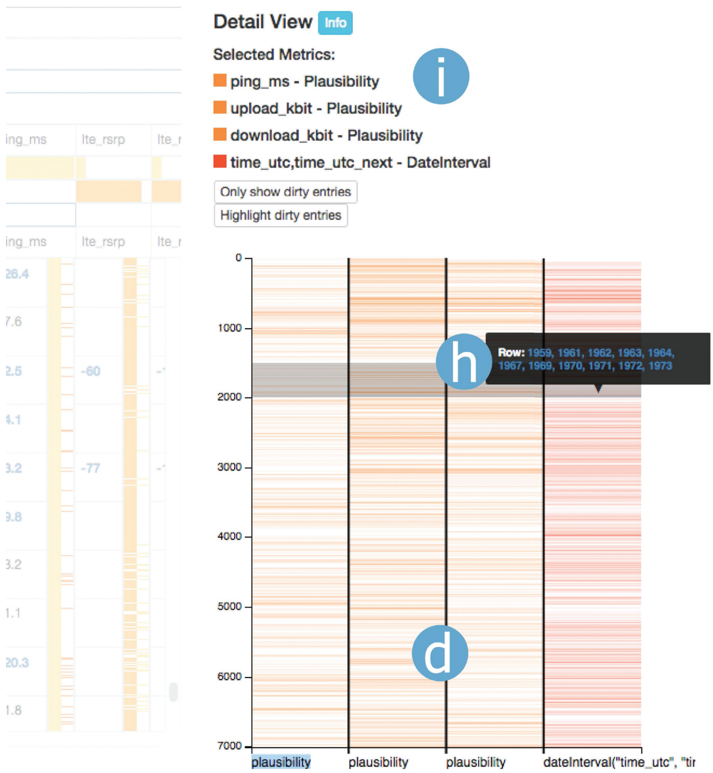

Fig. 4. *Metric Detail view* (d) showing the error distribution throughout the dataset of both the *completeness* metric of column *long* and the *date interval* metric for columns `time_utc`, `time_utc_next` as can bee seen in the legend (i). Users can toggle showing only dirty entries to facilitate 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 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 (cf. Figure 1(d) and (h)). Case study (see Section 6) task (3): The *metric detail view* shows test entries being called within a 10-second 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*).

raw data table representation with a scrollbar-like visualization, representing the relative position of dirty entries. 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 or 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 mouseover tooltips on error position and selection highlighting of raw data entries.

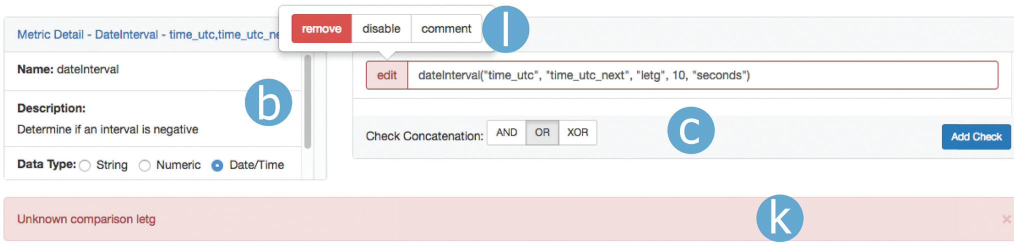

Fig. 5. Metric information (b) and customization tab (c), with edit feedback notification (j). In this case, the third metric parameter is misspelled, and the user is informed by an alert. After editing a metric, a notification informs the user of changes in the amount of detected errors. Quality checks 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 (cf. Figure 1(b) and (c)).

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). Data quality analysts could find the multitude of juxtaposed scroll elements distracting, and 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 a few quality metrics might be added to the dataset, this error distribution provides additional overview information and hence directs the analyst toward adding new quality metrics that fill the 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 (c.f. Figure 1(c)). Based on R1, quality metrics not only need to detect default errors specified by the data analyst but also should 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 Figure 5) that lets users edit quality metrics and gives additional information about the metric type. In the *checks tab* (Figure 5), 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 revalidation of the data quality, which is immediately reflected in the metric visualizations (*metric overview*, *metric detail view*, and *error distribution overview*). Moreover, the user is notified (see Figure 5(k)) about changes to error count and overall quality. Adding new metrics prompts a creation form, which 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.

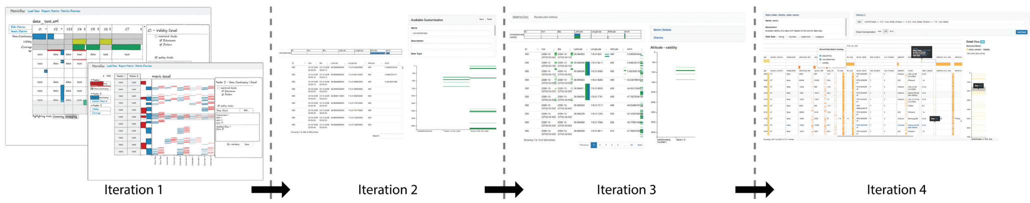

Fig. 6. The different stages of the design process of *MetricDoc*.

Disabling or enabling specific metrics or checks lets data quality analysts build up a backlog of quality checks that can be enabled for quick validation. This potentially boosts productivity, as 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.

In the following section, we will elaborate the design process as to how *MetricDoc*'s design iteration cycles were constituted.

5 ITERATIVE DESIGN PROCESS AND EVALUATION

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 [15, 26, 31, 32, 47, 50]. The development of *MetricDoc*—as of this writing—required four iteration cycles (see Figure 6). 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 3 and Section 4.1). In the following, we present the methods we applied throughout the iterative design process and evaluation and give a short retrospective analysis for each iteration.

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 [16, 19, 33]. 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 [14, 37, 49, 53]. 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, we applied the visualization-specific heuristics developed by Forsell and Johansson [14] and Tarrell et al. [49], which consider perception, cognition, usability, and interaction aspects. Furthermore, we 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 [9, 28, 33, 41]. Focus groups are means to get a quick understanding of users' perceptions, experiences, expectations, impressions, and opinions about a design from multiple points of view. Based on our previous work (e.g., [30]), we 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*, we conducted a focus group session with experts in the field of data quality, VA, and HCI in order to discuss and analyze the design from different points of view.

M4: MoSCoW Method [6, 8]. The heuristic evaluation, expert reviews, and focus group session were very constructive, and many interesting design ideas were collected. To prioritize the findings, we 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 one 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 development.

5.1 Iteration One: Conceptual Design

In the first iteration cycle, we concentrated on the creation of low-fidelity prototypes in consideration of the defined requirements (see Section 3). The goal of the prototypes was to explore different design ideas on how the data quality 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 6). 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 Section 3. 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 quality metrics and self-defined checks). Thus, in this iteration, we 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, we had **not yet prioritized metric customization** as an integral part of our functionality design.

5.2 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 quality metrics, the distribution of the detected dirty entries, the corresponding tabular representation, detailed information about the quality 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 3 and the design rationales in Section 4.1, 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 [14, 49]. 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 the design rationale D1: different colors were used for the same quality 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), we assigned a unique color to each quality metric to avoid confusion. This iteration led to the following outcomes:

- We 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.

5.3 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 quality 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 and 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 data quality experts, one HCI expert, 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 2 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 3):

(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 quality metrics and to identify dirty entries with respect to one or both quality 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 data quality experts highlighted that the prototype was powerful for checking the different columns with respect to different quality 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 6, 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 data quality experts wished for more means to make changes to metrics, e.g., merging metrics, previewing customized metrics, and saving and exporting metrics. Hence, the focus group session led to the conclusion that providing a comprehensive metric customization interface could enable data quality 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 data quality issues** with predefined metrics (with the ability to change parameters) to also **developing metrics**.
- Instead of implementing the suggested preview window for customizing metrics, we chose to provide **direct feedback to customizing metrics** by validating them syntactically during editing. Metric recalculation is performed upon saving the metric.
- Data quality 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).

5.4 Iteration Four: Final Development and Inspection

The goal of this iteration was to gather feedback from **data quality 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 [14, 49]. Furthermore, open questions that occurred during development were settled. For example, they discussed design ideas on 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 the *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:

Table 1. Distribution of Development (Orange) and Design (Cyan) Efforts over the Course of the Four Iteration Cycles and Q4 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 [46].

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				

- 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** (gray background in the *metric detail view*) to **indicate that nondirty 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 we discovered that data quality experts, though they appreciated 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 5(k)).

5.5 Lessons Learned

An iterative design process with short cycles of development and testing had the benefit that we were 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 (cf. [30]). For evaluating *MetricDoc*, we intertwined iterative prototyping and development with heuristic evaluation, a focus group, and 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 that were discovered during evaluation and reviewing. Table 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* [46]. 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 revisited, due to usability issues and suggestions by expert users.

In addition to the changes highlighted after each iteration cycle, we 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 quality 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.
- Data quality 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 toward 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.
- We discovered scalability issues with the initial design of the *metric detail view* that resulted in overplotting 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, we 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 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 in terms of the environment's potential usage in different application fields: one expert noted that the prototype could also be valuable for developing a powerful visual search environment in order to find specific data entries in tabular datasets. This led 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*.

Since we considered various perspectives from different domains of expertise during the different iterations of *MetricDoc*, we not only had the possibility to assess progress and get feedback from different points of view but we also could 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, we also encountered difficulties regarding further evaluation. The variety of approaches of assessing data quality implies that there are multiple valid practices toward 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 data quality. Constructing a comprehensive usage scenario that covers different kinds of insights, usage, and customization of quality metrics, as well as utilizing multiple views for exploration and evaluating them toward other data profiling and quality metric tools, is out of scope of this article and will be the subject of future work.

To show the benefits of using the *MetricDoc* environment for developing quality metrics and assessing data quality, we illustrate a case study in the following section.

6 CASE STUDY

Isenberg et al. [21] describe a *case study* as a report on how a new visualization approach can be used to solve/improve upon a certain problem, which in our case would be assessing data quality and profiling datasets. We want to show the *MetricDoc* environment 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 data quality metrics and immediately incorporate them in their analysis. With overview and detailed visualizations, the analyst can evaluate both new and updated datasets. As an example, we analyze 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 1 and 3; for more information please refer to RTR's NetTest Documentation[2]). The primary use of this dataset is to compare different Internet Service Providers' (ISPs') service quality, logging information like download and upload speed (in kbit/s), latency (in ms), and signal strength (in dBm). Also, anonymous metainformation (device name, network information, unique identifiers, etc.) is collected in order to compare different ISPs. We define the tasks for analyzing the dataset to be (1) checking 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 timeframe 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 (cf. Figure 1(c)). Browsing the *metric detail view* (cf. Figure 1(d)), 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 (cf. Figure 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 to create an expressive metric across multiple columns.

[2]https://www.netztest.at/en/OpenDataSpecification.html.

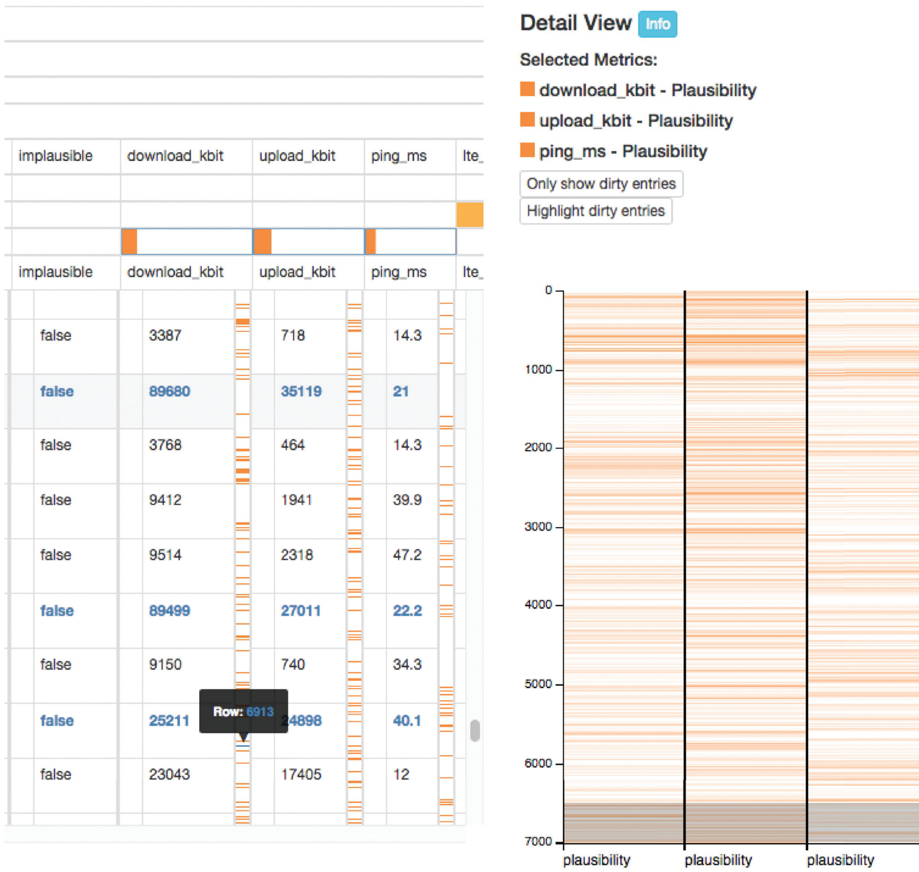

Fig. 7. Task (2): Extreme values can be observed; these might be subject to erroneous generation, skewing the ISP performance results.

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 data quality 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, entries that were identified implausible by all three metrics are highlighted. These three *plausibility* metrics can then be merged into one custom metric. Logically concatenating them detects entries that 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 entries detected by the *plausibility* metric directly correspond to outliers. The

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

Fig. 8. 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 that had a restricted number of threads (see column `num_threads`) predominantly had low download and upload rates as well as high latency.

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 the *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* 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). 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 recomputed 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 DISCUSSION

We consider user preference to be important to improve acceptance of *MetricDoc* among data quality experts. Design and development was focused on providing diverse interaction and exploration techniques to support users during data exploration based on data quality. Under this premise, our environment supports different workflows for metrics customization: both the creation of multiple simple metrics and the development of a 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. We also need to note limitations in terms of the complexity of metrics that can be developed in *MetricDoc*. Time-oriented quality 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, we prioritized the visual support for exploration and customization tasks. Immediate visual feedback is provided when changes occur, e.g., due to metric recalculations. 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 reapply existing metrics that 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 users to adapt metrics flexibly and assess the impact of the metrics on the dataset, supported by the employed visual feedback and visualizations.

8 CONCLUSION AND FUTURE WORK

In this article, we presented *MetricDoc*, an interactive visual exploration environment for the development and customization of data quality metrics for tabular datasets. *MetricDoc* allows for assessing the quality of a dataset by providing means for creating quality metrics and interactively exploring quality issues within the dataset that were detected by these metrics. We provide means for analysts to easily customize these metrics to suit their needs and domain-specific requirements. *MetricDoc* was designed, developed, and evaluated in a multistaged process that included prototyping, heuristic evaluation, and expert reviews, as well as a focus group with data quality, HCI, and VA experts. Although we already received valuable feedback from data quality experts during the focus group session, in future work we plan to conduct a long-term study with these participants so that missing features can be determined that occur when frequently and proficiently using the tool.

Foremost, we want to address the issue of scalability for the metric detail view by implementing a more sophisticated navigation technique to maintain effective error exploration while retaining positional awareness in the dataset. The set of quality metrics provided in the current environment could also be extended to employ statistical evaluation schemes, like contextual outlier detection and clustering, or leveraging linked open data information for validating entries. Due to the modularity of the metrics architecture, developers can easily implement metrics for particular data domains, like time-oriented and spatial data, which was not explicitly addressed. We also see the potential of our prototype to significantly reduce the time required for data profiling when dealing with multiple sources, since quality issues are especially hard to determine in this context. To this end, we want to develop a specific set of metrics and tools for handling multisource data.

Another important topic for future work is the integration of provenance information. Collecting provenance information during data wrangling and cleansing is crucial for retaining information about how the dataset changed over time. Integrating provenance into our environment could significantly increase its attractiveness to users: showing comparative visualizations of quality issues between different dataset revisions would provide a detailed view on the impact of specific transformations on data quality. This would also allow for a better-informed handling of data quality issues. Different versions of the data and different metrics can describe multiple aspects of the dataset and allow the user to explore alternative applications of the data.

REFERENCES

[1] José Barateiro and Helena Galhardas. 2005. A survey of data quality tools.*Datenbank-Spektrum* 14, 15–21 (2005), 48.

[2] Carlo Batini, Cinzia Cappiello, Chiara Francalanci, and Andrea Maurino. 2009. Methodologies for data quality assessment and improvement. *ACM Computing Surveys* 41, 3 (July 2009), 16:1–16:52. DOI: http://dx.doi.org/10.1145/1541880.1541883

[3] Carlo Batini and Monica Scannapieco. 2006. *Data Quality: Concepts, Methodologies and Techniques (Data-Centric Systems and Applications)*. Springer Verlag New York,Secaucus, NJ.

[4] Jürgen Bernard, Tobias Ruppert, Oliver Goroll, Thorsten May, and Jörn Kohlhammer. 2012. Visual-interactive preprocessing of time series data. In *Proceedings of SIGRAD 2012: Interactive Visual Analysis of Data*. 39–48.

[5] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. 2011. D3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (Dec. 2011), 2301–2309. DOI: http://dx.doi.org/10.1109/TVCG.2011.185

[6] K. Brennan. 2009. *A Guide to the Business Analysis Body of Knowledger*. International Institute of Business Analysis.

[7] Daniel Castellani Ribeiro, Huy Vo, Juliana Freire, and Cláudio Silva. 2015. An urban data profiler. In *Proceedings of the 24th International Conference on World Wide Web (WWW'15 Companion)*. ACM, New York, 1389–1394. DOI: http://dx.doi.org/10.1145/2740908.2742135

[8] Dai Clegg and Richard Barker. 1994. *Case Method Fast-Track: A Rad Approach*. Addison-Wesley Longman Publishing Co.

[9] Catherine Courage and Kathy Baxter. 2004. *Understanding Your Users: A Practical Guide to User Requirements Methods, Tools, and Techniques*. Morgan Kaufmann Publishers.

[10] Tamraparni Dasu. 2013. Data glitches: Monsters in your data. In *Handbook of Data Quality*, Shazia Sadiq (Ed.). Springer, Berlin, 163–178. http://dx.doi.org/10.1007/978-3-642-36257-6_8

[11] DataTables. 2017. DataTables | Table plug-in for jQuery. Retrieved from https://datatables.net/ (accessed May 2017).

[12] Jeremy Debattista, Makx Dekkers, Christophe Guret, Deirdre Lee, Nandana Mihindukulasooriya, and Amrapali Zaveri. 2016. Data on the Web Best Practices: Data Quality Vocabulary. Retrieved from https://www.w3.org/TR/vocab-dqv/.

[13] Wayne W. Eckerson. 2002. *Data Quality and the Bottom Line: Achieving Business Success through a Commitment to High Quality Data*. Technical Report. The Data Warehousing Institute. Retrieved from tdwi.org/research/2002/02/tdwis-data-quality-report.aspx (accessed April 17, 2014).

[14] Camilla Forsell and Jimmy Johansson. 2010. An heuristic set for evaluation in information visualization. In *Proceedings of the International Conference on Advanced Visual Interfaces (AVI'10)*. ACM, 199–206.

[15] Carla M. D. S. Freitas, Marcelo S. Pimenta, and Dominique L. Scapin. 2014. User-centered evaluation of information visualization techniques: Making the HCI-InfoVis connection explicit. In *Handbook of Human Centric Visualization*, Weidong Huang (Ed.). Springer, 315–336.

[16] Wilbert O. Galitz. 2007. *The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques*. Wiley & Sons.

[17] Theresia Gschwandtner, Wolfgang Aigner, Silvia Miksch, Johannes Gärtner, Simone Kriglstein, Margit Pohl, and Nik Suchy. 2014. 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)*. ACM, New York, 18:1–18:8. DOI: http://dx.doi.org/10.1145/2637748.2638423

[18] Theresia Gschwandtner, Johannes Gärtner, Wolfgang Aigner, and Silvia Miksch. 2012. A taxonomy of dirty timeoriented data. In *Lecture Notes in Computer Science (LNCS 7465): Multidisciplinary Research and Practice for Information Systems (Proceedings of the CD-ARES'12)*, Gerald Quirchmayr, Josef Basl, Ilsun You, Lida Xu, and Edgar Weippl (Eds.). Springer, Berlin, 58–72. DOI: http://dx.doi.org/10.1007/978-3-642-32498-7_5

[19] Rex Hartson and Pardha A. Pyla. 2012. *The UX Book: Process and Guidelines for Ensuring a Quality User Experience*. Morgan Kaufmann.

[20] Jeffrey Heer, Joseph Hellerstein, and Sean Kandel. 2015. Predictive interaction for data transformation. In *Conference on Innovative Data Systems Research (CIDR'15)*.

[21] Tobias Isenberg, Petra Isenberg, Jian Chen, Michael Sedlmair, and Torsten Möller. 2013. A systematic review on the practice of evaluating visualization. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (Dec. 2013), 2818–2827. DOI: http://dx.doi.org/10.1109/TVCG.2013.126

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

[23] Sean Kandel, Andreas Paepcke, Joseph Hellerstein, and Jeffrey Heer. 2011. Wrangler: Interactive visual specification of data transformation scripts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'11)*. ACM, New York, 3363–3372. DOI: http://dx.doi.org/10.1145/1978942.1979444

[24] Sean Kandel, Ravi Parikh, Andreas Paepcke, Joseph M. Hellerstein, and Jeffrey Heer. 2012. Profiler: Integrated statistical analysis and visualization for data quality assessment. In *Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI'12)*. ACM, New York, 547–554. DOI: http://dx.doi.org/10.1145/2254556.2254659

[25] Daniel Keim. 2002. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics* 8, 1 (2002), 1–8. DOI: http://dx.doi.org/10.1109/2945.981847

[26] Andreas Kerren, Achim Ebert, and Jörg Meyer. 2006. Introduction to human-centered visualization environments. In *Human-Centered Visualization Environments (Lecture Notes in Computer Science)*, Andreas Kerren, Achim Ebert, and Jörg Meyer (Eds.). Springer, 1–9.

[27] Won Kim, Byoung-Ju Choi, Eui-Kyeong Hong, Soo-Kyung Kim, and Doheon Lee. 2003. A taxonomy of dirty data. *Data Mining and Knowledge Discovery* 7, 1 (Jan. 2003), 81–99.

[28] Jenny Kitzinger. 1995. Qualitative research: Introducing focus groups. *BMJ* 311, 7000 (1995), 299–302.

[29] Simone Kriglstein, Margit Pohl, and Michael Smuc. 2014. Pep up your time machine: Recommendations for the design of information visualizations of time-dependent data. In *Handbook of Human Centric Visualization*, Weidong Huang (Ed.). Springer New York, 203–225.

[30] Simone Kriglstein, Margit Pohl, Nikolaus Suchy, Johannes Gärtner, Theresia Gschwandtner, and Silvia Miksch. 2014. Experiences and challenges with evaluation methods in practice: A case study. In *Proceedings of the 5th Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization (BELIV'14)*. ACM, 118–125.

[31] Simone Kriglstein and Günter Wallner. 2013. Human centered design in practice: A case study with the ontology visualization tool knoocks. In *Computer Vision, Imaging and Computer Graphics. Theory and Applications*, Gabriela Csurka, Martin Kraus, Leonid Mestetskiy, Paul Richard, and Jos Braz (Eds.). Springer, 123–141.

[32] Olga A. Kulyk, Robert Kosara, Jaime Urquiza-Fuentes, and Ingo H. C. Wassink. 2006. Human-centered aspects. In *Human-Centered Visualization Environments (Lecture Notes in Computer Science)*, Andreas Kerren, Achim Ebert, and Jörg Meyer (Eds.). Springer, 13–75.

[33] Martin Maguire and Nigel Bevan. 2002. 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*. Kluwer, B.V., 133–148.

[34] Silvia Miksch and Wolfgang Aigner. 2014. 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 (2014), 286–290. DOI: http://dx.doi.org/10.1016/j.cag.2013.11.002

[35] Heiko Müller and Johann-Christoph Freytag. 2003. *Problems, Methods, and Challenges in Comprehensive Data Cleansing*. Technical Report. HUB-IB-164, Humboldt University Berlin, Berlin. 50–99 pages.

[36] Tamara Munzner. 2014. *Visualization Analysis and Design*. A K Peters Visualization Series, CRC Press.

[37] Jakob Nielsen. 1994. Usability Inspection Methods. Wiley & Sons, Inc., Chapter Heuristic Evaluation, 25–62.

[38] Paulo Oliveira, Fátima Rodrigues, and Pedro Rangel Henriques. 2005. A formal definition of data quality problems. In *IQ*.

[39] Open Refine. 2017. OpenRefine. Retrieved from https://github.com/OpenRefine/OpenRefine https://github.com/OpenRefine/OpenRefine (accessed May 2017).

[40] Leo L. Pipino, Yang W. Lee, and Richard Y. Wang. 2002. Data quality assessment. *Communications of the ACM* 45, 4 (April 2002), 211–218. DOI: http://dx.doi.org/10.1145/505248.506010
[41] Richard A. Powell and Helen M. Single. 1996. Focus groups. *International Journal for Quality in Health Care* 8, 5 (1996), 499–504.
[42] R Core Team. 2017. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. Retrieved from https://www.R-project.org.
[43] Erhard Rahm and Hong-Hai Do. 2000. Data cleaning: Problems and current approaches. *IEEE Bulletin of the Technical Committee on Data Engineering* 23, 4 (March 2000), 3–13.
[44] Ramana Rao and Stuart Card. 1994. 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*. ACM, New York, 318–322. DOI: http://dx.doi.org/10.1145/191666.191776
[45] Shazia Sadiq (Ed.). 2013. *Handbook of Data Quality*. Springer Verlag, Berlin.
[46] Margrit Schreier. 2012. *Qualitative Content Analysis in Practice*. SAGE Publications.
[47] Michael Sedlmair, Miriah Meyer, and Tamara Munzner. 2012. Design study methodology: Reflections from the trenches and the stacks. *IEEE Transactions on Visualization and Computer Graphics (Proc. InfoVis)* 18, 12 (2012), 2431–2440.
[48] Awalin Sopan, Manuel Freire, Meirav Taieb-Maimon, Catherine Plaisant, Jennifer Golbeck, and Ben Shneiderman. 2013. Exploring data distributions: Visual design and evaluation. *International Journal on Human Computer Interaction* 29, 2 (2013), 77–95. DOI: http://dx.doi.org/10.1080/10447318.2012.687676
[49] Alvin Tarrell, Ann Fruhling, Rita Borgo, Camilla Forsell, Georges Grinstein, and Jean Scholtz. 2014. Toward visualization-specific heuristic evaluation. In *Proceedings of the 5th Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization (BELIV'14)*. ACM, 110–117.
[50] Melanie Tory and Torsten Möller. 2004. Human factors in visualization research. *IEEE Transactions on Visualization and Computer Graphics* 10, 1 (2004), 72–84.
[51] Trifacta. 2016. Trifacta Wrangler. Retrieved from https://www.trifacta.com/trifacta-wrangler/.
[52] Edward R. Tufte. 2006. *Beautiful Evidence*. Graphics Press.
[53] Torre Zuk, Lothar Schlesier, Petra Neumann, Mark S. Hancock, and Sheelagh Carpendale. 2006. 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)*. ACM, 1–6.

Received July 2018; revised January 2018; accepted February 2018