Know Your Enemy: Identifying Quality Problems of Time Series Data

Theresia Gschwandtner* Oliver Erhart†

Institute of Visual Computing and Human-Centered Technology
TU Wien

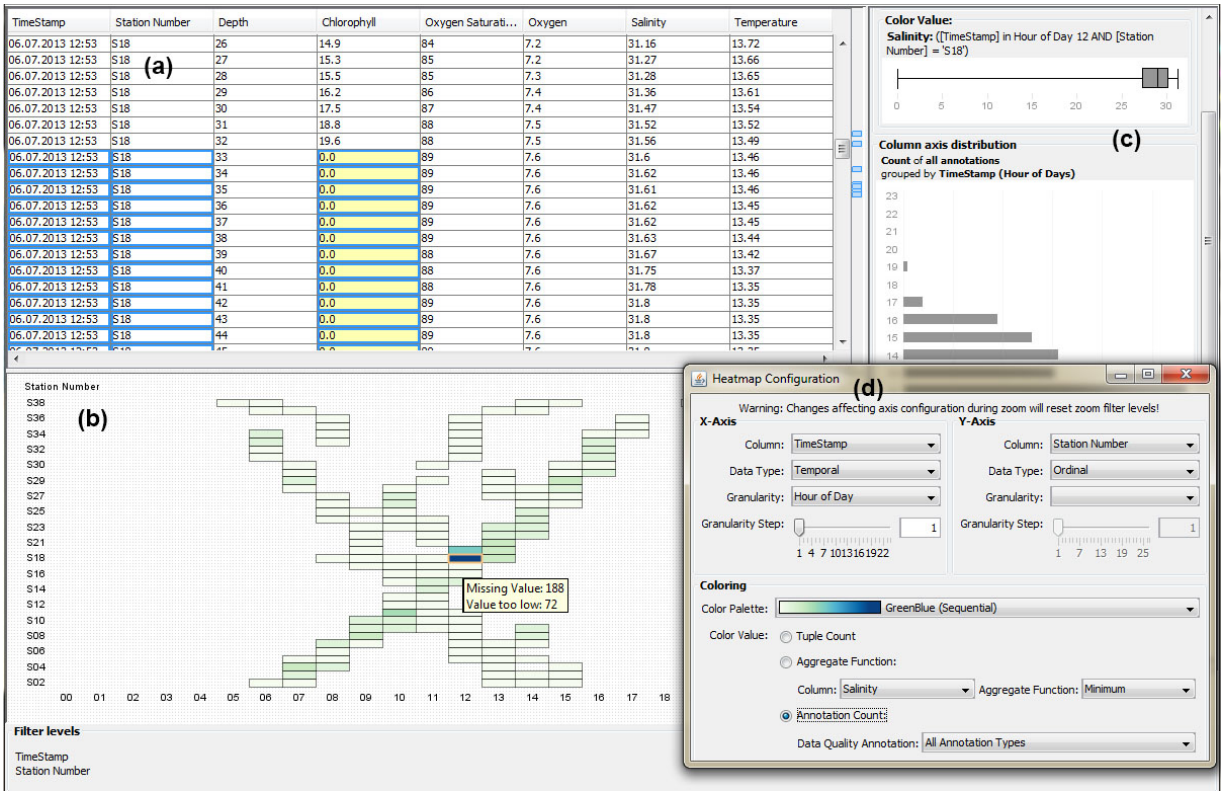

Figure 1: KYE is composed of three panes: (a) the interactive table view relates visualizations with original data values, (b) the heatmap view supports both, investigating automatically detected problems and identifying additional data quality problems, and (c) the statistics view provides additional information for an informed reasoning. Here we show how KYE helps to reason about possible quality problems detected by automatic means. The heatmap (b) is configured (d) to relate different water measurement stations (y-axis) with the hour the measurements were taken (x-axis). The color is mapped to the total amount of detected quality problems for each heatmap cell. This reveals possible quality problems for measurements at station S18 at 12.00 noon. The tooltip shows that most detected problems for this cell refer to missing values. By exploring these values in the table view (a) we reason that the sensor which measures chlorophyll broke at 33 meter water depth, and thus, caused a big amount of missing values.

Abstract

Sensible data analysis requires data quality control. An essential part of this is data profiling, which is the identification and assessment of data quality problems as a prerequisite for adequately handling these problems. Differentiating between actual quality problems and unusual, but valid data values requires the "human-in-the-loop" through the use of visual analytics. Unfortunately, existing approaches for data profiling do not adequately support the special characteristics of time, which is imperative to identify quality problems in time series data – a data type prevalent in a multitude of disciplines. In this design study paper, we outline the design, implementation, and evaluation of "Know Your Enemy" (KYE) – a visual analytics approach to assess the quality of time series data. KYE supports the task of data profiling with (1) predefined data quality checks, (2) user-definable, customized quality checks, (3) interactive visualization to explore and reason about automatically detected problems, and (4) the visual identification of hidden quality problems.

Index Terms: H.5.2 [Information Interfaces And Presentation]: User Interfaces—User-centered design; I.3.6 [Computer Graphics]: Methodology and Techniques—Graphics data structures and data types

1 Introduction

The outcome of any automatic data processing or analysis can be only as good as the quality of the data set that is processed. Real-life data often contains quality problems which can be for instance invalid, erroneous, or missing values, as well as outliers, or duplicate records [7]. Some tasks may require perfectly accurate data, while

*e-mail: gschwandtner@tuwien.ac.at
†00825648@student.tuwien.ac.at

Publication forthcoming in Proc. of IEEE Pacific Visualization Symposium (PacificVis '18)

others may be tolerant to some erroneous data entries. In any case, the quality of the data at hand must be assessed to understand if the data is fit for use. Data quality management includes *data profiling*, *data cleansing*, and *data transformation* [12]. While *data profiling* is concerned with identifying data quality problems in the data, *data cleansing* deals with the correction of these problems, and *data transformation* changes the data format to match given requirements. Especially in *data profiling*, human judgment is needed because it requires expert knowledge and reasoning about the data in its context to understand if an unusual value presents an actual data error [13]. Again, other problems can be easily detected by automatic means. Thus, a visual analytics (VA) approach lends itself to support the task of data profiling by combining automatic quality checks with visual exploration of the data.

A ubiquitous type of data are time series, which need to be analyzed and processed, for instance, in climate research or high-energy physics. Time is a special data type that induces specific data quality problems [9]. Moreover, time-dependent data values need to be analyzed in their temporal context to be able to identify certain quality problems (e.g., too high numbers of produced items within an hour, or huge velocity changes in very short time). In this paper we present the design and evaluation of a VA prototype called "Know Your Enemy" (KYE) which supports the task of data profiling – in particular of time series data. In this context, our contributions are:

- the design and evaluation of a VA data profiling solution, with special support for the characteristics of time series data which is crucial to identify a number of data quality problems,
- a discussion of design choices and iterative design refinements,
- a discussion of lessons learned, and the derivation of further research challenges.

2 Related Work

Whereas data quality is a broad term, this paper specifically focuses on data profiling for time series data. There are a number of VA approaches tackling the problem of data quality. Profiler [14] is a VA web application integrated with Wrangler's [13] data transformation engine. It handles five categories of anomalies: missing, erroneous and inconsistent data, extreme values, and key violations. One key aspect of Profiler is that it automatically provides suitable visualizations for different data types to give an overview of the data and the automatically identified quality problems. It supports date objects and the visualization of temporal bar charts. However, Profiler does not specifically support quality problems induced by time series data.

Talend Open Studio [28] is an open-source data profiling application, which provides mainly statistics (minimum, maximum, and missing values) about different data types of a data set (including dates). However, finer temporal granularities such as hours, minutes, and seconds, are not supported.

OpenRefine [30] is focused on the transformation of data sets supported by statistics and visualizations. Temporal bar charts can be used for detecting outliers, getting an overview about value distribution, and filtering temporal ranges. The integrated GoogleRefine Expression Language allows for user-defined transformation operations. While OpenRefine offers some implicit profiling means, it is rather focused on supporting data transformations.

DataMatch [11] is specifically focused on the detection of duplicate data records. It provides simple statistics for different data types, including dates. However, time of day is not supported and the duplicate detection of dates is limited as it handles dates as Strings.

Other data quality tools that provide some profiling functionality are DataManager [4], which provides profiling statistics, but does not support temporal data types at all, and Datamartist [5], which supports temporal data types as well as calculations such as day of week, and whether the date is a weekday. It also supports value distributions of years and months, but no finer granularities than this.

These approaches do not specifically focus on time series data, and thus special quality problems with respect, for instance, to interval lengths, evenly spaced time stamps, gaps and overlaps of intervals, plausible temporal ranges, or plausible time-varying data values are not supported. There are only selected approaches that focus on the quality of time series data and its special characteristics [1, 2, 8].

The visual-interactive preprocessing of time series data [2] presented by Bernard et al. is a system for preparing time series data for further processing by means of data reduction, data normalization, data segmentation, descriptors, and similarity measures. Thus, this solution supports data cleansing operations of time series but no data profiling.

TimeCleanser [8] is a VA prototype providing a number of automatic checks with a special focus on time-induced quality problems. Moreover, it provides visualizations such as line charts, bar charts, and heatmaps that help to detect anomalies. However, these charts provide predefined views on the data which cannot be configured flexibly enough to allow for a detailed exploration of all aspects of the data. In addition, it does not provide interactive visualizations to explore and reason about the quality problems detected by the provided checks.

Visplause [1], on the other hand, is a system for inspecting the quality of many time series at once. It provides automatic data quality checks, such as missing values, constraint violations, and anomalies, as well as visualizations to communicate the number of quality problems within each time series. Thus, it is rather specialized on communicating an overview to understand if the data is ready to use. Another difference of Visplause compared to our approach is that it mostly communicates the results of automatic checks but does not provide means for the identification of additional hidden quality problems.

Besides existing VA approaches, taxonomies of data quality problems provide valuable information to understand which tasks and problems should be supported by a data profiling solution. There are a number of taxonomies of general data quality problems [15, 23, 25], while Gschwandtner et al. [9] provide a taxonomy including important quality aspects for time series data. Thus, we use this taxonomy of dirty time-oriented data as a basis for our work.

3 KYE

To fill these gaps, i.e., designing a VA solution which (1) runs (predefined and user-defined) automatic checks to identify quality problems, (2) communicates these problems to the user, (3) provides interactive visualizations to investigate them, and (4) provides visualizations to identify additional quality problems, we have developed KYE. In this section we describe the requirements, the iterative design process, the provided automatic quality checks, the design choices, and the provided interaction functionality of our VA solution.

3.1 Requirements

In a first step we identified desirable features of our prototype by deriving open challenges from the works discussed in Section 2, by reviewing related literature (in particular the taxonomy of dirty time-oriented data [9]), and from our long-lasting experience with data quality work from previous projects. In particular, this resulted in the following list of requirements:

R1 Data quality checks: Our VA solution should provide predefined, ready-to-use, automatic quality checks, as well as the possibility to define customized quality checks, ensuring its adaptability to a multitude of application domains. Providing

automatic means is essential to facilitate common laborious profiling tasks.

R2 Exploration of quality problems: Our VA solution should communicate automatically detected data quality problems and make them explorable, so that the user can investigate context and possible causes of quality problems, and make informed decisions how to best handle them.

R3 Identification of hidden problems: Our VA solution should, in addition, provide means to explore the data at hand, and identify further data quality problems that were not detected by automatic means.

R4 Scalability: Our VA solution should provide a scalable overview of quality problems of possibly huge data sets.

3.2 Iterative design

For the design and implementation of our prototype, we followed an iterative design process. In a first step we analyzed the data, users, and task [20] we were going to tackle: (1) As we cannot support all **data** formats and types possible, we decided on some constraints which will still be met by the data of a multitude of domains: our prototype should support data tables containing ordinal, numerical, and temporal data. (2) In addition, we defined our target **users** as experts in their respective domains and data analysts, having some experience with computer-supported data analysis, and (3) we focused on the **task** of data profiling, i.e., identifying quality problems of a given data set. Moreover, our design was guided by the nested model for visualization design and validation [22].

Our first designs were sketches that illustrated how different visualization techniques could be used to foster the identification of specific data quality problems from Gschwandtner et al. [9]. Figure 2 (a) shows a time-line chart to identify abrupt changes of numerical values over time and Figure 2 (b) shows a chart displaying intervals as bars over time to identify irregularities in duration lengths. This resulted in a number of chart types, each aimed at supporting the identification of a specific quality problem. Thus, we shifted to more flexible visualization methods that could be used to communicate a multitude of problems. Figure 2 (c) shows a bar chart over time stacking different types of quality problems. However, in further design iterations we settled for a heatmap visualization (see Figure 2 (d)), as it best suited our needs (see Section 3.4). In a next step, we implemented a first proof-of-concept prototype to get a better feeling for our visualization design and for the interactions that are needed to support data profiling (see Figure 2 (e)). Eventually, Figure 2 (f) shows the final prototype including a statistics panel on the right and various interactive features.

3.3 Automatic Checks

For an efficient profiling process KYE provides a number of predefined quality checks (requirement R1). Each of these checks tackles one or more quality problems specific to time series data from Gschwandtner et al. [9]:

- Timing of values: checks if the time-stamps of data entries are evenly spaced. Users must specify the column(s) to check and the raster length. If the raster length is not provided, is is guessed automatically from the data. The check identifies entries that do not fit into this time raster as well as missing time-stamps.

- Interval lengths: checks the length of intervals in the data set defined by two points in time (start and end). An interval is valid if it has a positive length (the start time is before the end time). Optionally, it checks if the length of the interval is either within certain bounds (user-defined bounds or statistically derived from the data) or all intervals are of same lengths.

- Matching temporal ranges: checks if entries with different keys lie within the same temporal range (with some tolerance), to check, for instance, if the data set covers similar time ranges for different departments.

- Plausibility of times and values: checks if time-stamps and values are plausible. For instance, outlying time-stamps (far away from the majority of time-stamps or far away from their neighboring time-stamps) or values on weekends or during the night are identified as suspicious. The same applies to outlying quantitative and nominal values. The parameters for plausibility are either defined by the user or derived by statistical methods. We provide two outlier detection methods [24]: (1) for each value we compute the distance to the k^{th} nearest neighbour and identify as outliers the n values with the largest distances, and (2) we identify outliers with fewer than k neighbours (within a given distance) in the data set.

- Plausibility of duration-dependent values: checks if numerical values that are associated with a time interval and are expected to vary with duration (e.g., the number of items produced for different temporal aggregations: hour, day, etc.) are plausible, for instance, higher quantities for longer time intervals or similar quantities per minute. Again, the plausible ranges can be defined by the user or statistically computed.

- Valid sequences: checks if the temporal sequence of values is correct (times and values), for instance, quantitative values that are supposed to be rising or falling with time or nominal values that should follow a given sequence.

- User-definable checks: users can define specialized checks by entering regular expressions.

These checks are applied by choosing a check from the menu and configuring its parameters. Invalid data entries are flagged to be further explored with the heatmap and table view provided by KYE. However, in the scope of this paper we focus on the interactive visual exploration of these automatically detected problems and the interactive visual detection of additional hidden problems.

3.4 Design Decisions

In addition to analyzing the works focusing on data quality (see Section 2), we took a step back and analyzed the suitability of basic visualization techniques to our needs. To this end, the TimeViz Browser [29], a collection of more than hundred scientific visualization techniques, was an important source of inspiration. We printed all visualization techniques for a first brainstorming session and identified five different approaches potentially suitable to communicate, explore, and identify data quality problems:

1. Stacked bar charts: This visualization could be used to stack annotated data quality problems in a bar chart, each bar representing either an instance of a user defined temporal granularity (e.g., each bar representing one day), or any parameter of the underlying data set (e.g., each bar representing temperature). The height of a bar would allow for the immediate identification of accumulations of quality problems. However, the aspect ratio of such a chart makes it a less ideal match for our needs (i.e., providing an overview of huge amounts of data entries). Moreover, specifics of time series data, such as calendrical phenomena (e.g., a problem occurs each day at 10 am) cannot be supported.

2. ThemeRiver [26]: Using a ThemeRiver-like visualization would be an option to show the trend of different types of quality problems over time. Again, this visualization has problems with scalability and the support of calendrical phenomena.

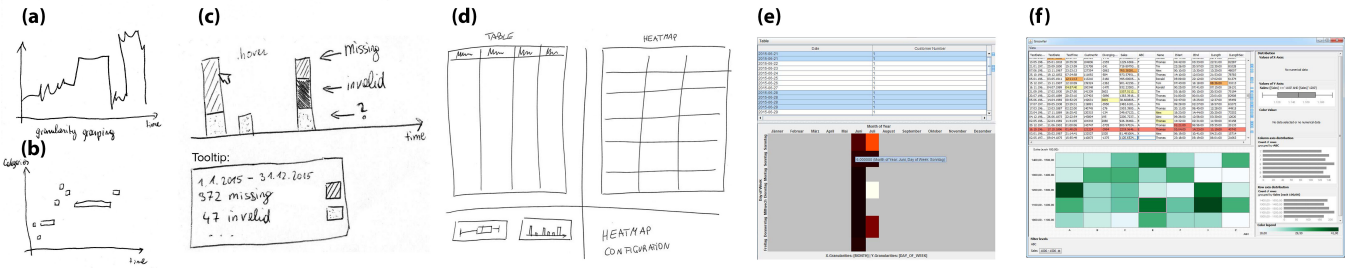

Figure 2: Iterative design process. The mockups and designs choices that finally led to the design of KYE range from simple sketches to high fidelity prototypes. The sketches in (a) and (b) were among our first ideas to detect specific quality problems. We considered (a) a time-line chart to identify abrupt changes of numerical values over time and (b) a chart displaying intervals as bars over time to identify irregularities in duration lengths. The stacked temporal bar chart in (c) was designed to communicate a multitude of quality problems in one graph. In (d) we already contemplated a heatmap visualization and sketched out the arrangement of different views. The first high fidelity prototype (e) was used to verify the design choices and help to understand what interactions would be needed. (f) shows the final design of the prototype.

Investigating the data at different time granularities can also not easily be solved with this type of visualization.

3. Spiral graphs: An interactive spiral graph [31] could be used to address calendrically occurring data quality problems. Drawbacks include the difference in visual saliency of phenomena of the same magnitude, depending on their position in the spiral (items close to the center are mapped to lesser space than items at the outer rings). Moreover, the spiral graph only displays time on the spiral axis, whereas we wanted to provide a solution flexible enough to contrast time with time (e.g., months on x-axis and days on y-axis), time with a non-temporal data type (i.e., a temporal data type on one axis and an ordinal or numerical data type on the other axis), or two non-temporal data types (non-temporal data on both axes).

4. Calendar-based visualizations: A Tile Map [21] or GROOVE [17] could be used to display the amount of detected data quality problems mapped to color or opacity and grouped by temporal granularity. Therefore users might discover calendrically occuring data quality problems.

5. Heatmaps: Similar to the calendar-based visualization techniques described above, conventional heatmaps could be used to present the data grouped into smaller bins and a quantitative value mapped to color or opacity. However, conventional heatmaps do not necessarily support the grouping of the data based on temporal granularities or calendrical structures. However, by mapping the two available axes of conventional heatmaps to non-temporal data types, it is possible to detect interesting patterns and irregularities.

Considering these results, we decided to combine the strength of both, calendar-based visualizations and heatmaps. Thus, we settled for a two-dimensional heatmap visualization in combination with temporal/calendrical aggregation functions (e.g., aggregating all Mondays). When employing suitable aggregation methods, heatmaps scale well to give an overview of large data sets (requirement R4), while on the other hand, when combined with zoom and filter techniques, they are also suited to scrutinize the data on a fine grained level. The two axes allow to relate two different temporal granularities, for instance, weekdays on the x-axis and hours on the y-axis, which is suited to visualize daily/weekly/monthly profiles. Moreover, heatmaps allow for the visual identification of outliers (see Figure 3a), trends, patterns (see Figure 3a and b), and seasonal behavior (see Figure 3c) in time series data. Heatmaps can also be configured to visualize cyclical patterns similar to spiral graphs, for instance, by displaying days at the x-axis and years at the y-axis. In addition, they are also suited to set into relation any other two non-temporal data dimensions (quantitative or nominal) to investigate dependencies and correlations, and thus, to find additional data quality problems.

3.4.1 The Heatmap

Aiming for a flexible, domain-independent approach, we allow users to configure the heatmap visualization to serve their specific needs (see Figure 1b).

Binning A user can map any columns of the data set to the axes of the heatmap, i.e. either quantitative data (e.g., price), nominal data (e.g., category) or temporal data (e.g., time of purchase). Thus, users have the ability to interactively change and combine axes regarding to their needs. Moreover, the step-width of axis ticks can be modified. This parametrization of axes defines the binning of data tuples. A cell of the heatmap contains a subset (list of tuples) of the data which falls into the ranges defined by the axes. Temporal axes (based on date entries, time entries, or both) display data in the provided temporal granularity. From, there the user can group the data into coarser granularities by zooming out. Nominal axes render categorical data sorted in alphabetical order. These categories represent the nominal entries included in the selected data column, and thus, the amount of categories depends on the amount of different nominal values within this column. A huge number of different nominal values would lead to a rather pixel-based representation. This can be used as a starting point to zoom into selected categories for further exploration. Quantitative axes represent a linear range of a column's minimum value to its maximum value. In contrast to temporal or nominal data, quantitative data, in particular floating point data, has no natural non-decomposable unit which can serve as the step-width of the axis. Therefore, the user needs to define the step-width for quantitative axes. Our visualization allows scrutinizing the data set at hand by interactively mapping

- both axes to different time granularities, which fosters the identification of nested calendrical phenomena (e.g., accumulation of problems on Mondays, 10 am),

- one axis to a time granularity and the other axis to a non-temporal data column, which fosters the identification of calendrical phenomena with other data parameters (e.g., accumulations of problems on Mondays in a certain department), or

- both axes to non-temporal parameters of the data set, which allows for the detection of data relations and patterns, as well as irregularities that might hint to data quality problems (e.g.,

accumulations of problems at low temperatures in combination with certain materials).

Cell Coloring With flexible configuration of axes we offer different views on the data for data profiling. The color mapping of the heatmap cells holds additional possibilities for the detection of quality problems. We do not only strive to make detected data quality problems explorable within their context, but also to foster the identification of overlooked problems. Thus, our visualization supports different ways of mapping data to cell color (see Figure 3):

- Number of detected problems: To **communicate automatically detected quality problems** and make them explorable, KYE provides mapping cell color to the number of detected quality problems. Furthermore, users can choose to only show specific types of quality problems or all quality problems at once. Hence, users can explore and analyze the distribution of detected quality problems (requirement R2) and identify possible peculiarities.

- Tuple count: A straight-forward but essential way for identifying irregularities in the data is mapping cell color to the amount of tuples within a bin. This helps to get an overview of occurrences of tuples and their distribution, and thus, **identify potentially hidden quality problems** (requirement R3).

- Calculated key figures: Another option is to color the cells according to key figures calculated from parameter values. For instance, a data of items sold in a museum's shop may contain a data row for each opening hour, and the amount of items sold within this hour is given by a numeric value. A selected bin of our visualization may represent sales on Monday, December 1st, 2016. Thus, it is not of interest to map the tuple count (i.e. opening hours of this Monday) to cell color, but we need to calculate the sum of sales during this period. KYE provides the following key figures: *count* (of entries), *distinct count* (count of distinct entries), *median*, *sum*, *mean*, *standard deviation*, *minimum*, and *maximum values*. While *sum*, *mean*, *standard deviation*, *minimum*, and *maximum* can only be applied to numerical data, *count*, *distinct count*, and *median* can also be used for ordinal and temporal data. This mode is also aimed at **identifying potentially hidden quality problems** (requirement R3).

Moreover, we provide a variety of color scales (from ColorBrewer [10]), since there is no single color scale that would fit any need. For instance, depending on the user's culture and social background, it could be confusing to map hot temperatures to blue values. As most color palettes map low values as a very light color, it can be difficult to differentiate between heatmap cells that contain low values and empty heatmap cells. In many cases this might not be a problem, however, we are especially interested in identifying bins containing a very small number of values, as they may indicate quality problems. Thus, we introduced a dotted background pattern to the heatmap to make these cases visually distinguishable. Another way to emphasize outliers on each side of the spectrum is to choose a diverging color scale.

3.4.2 Table View

For being able to reason about suspicious data or data with automatically detected quality problems, it is essential that the user can connect the visual representations that point to these patterns to the original data values. Thus, we provide a table view showing the raw data table (see Figure 1a). Table cells that contain data for which a problem was detected have colored backgrounds, using a qualitative color scale to encode different types of data problems by different colors hues. A blue border is added to the table cell if it contains data that was selected in the heatmap. On the right side of the table's scroll bar, we place a summary view of table entries selected in the heatmap, and their position within the table. As these entries can be spread across the table, an automatic navigation to one table row after selecting a heatmap cell does not make sense. Thus, we provide these visual markers as interactive navigation aids. When the user selects a marker, the table is automatically scrolled to the respective position.

3.4.3 Statistics View

The statistics view (see Figure 1c) consists of three box-plots and two histograms. The three box-plots summarize different aspects of the data in a selected heatmap cell: distribution of values on the x-axis, distribution of values on the y-axis, and distribution of values mapped to color (given that these are numerical values). The two histograms show the values that are mapped to the color of heatmap cells (i.e., number of detected problems, tuple count, or calculated key figures) for each axis, to offer a different perspective on these numbers.

3.5 Interactions

For the design of the interactive functionality provided by KYE we implemented the Visual Information Seeking Mantra by Shneiderman [27]:
Overview: A good overview of the data and existing quality problems is realized by the heatmap visualization and the possibilities to configure the mapping of data to axes and cell color. Further interaction means to configure the overview to the user's needs are step-width and the computation of key figures about the data and subsequently mapping these to cell color (see Section 3.4).
Zoom & filter: We provide zooming into a heatmap cell. To this end, we need to consider three different ways of zooming: (1) If a heatmap axis represents numerical data, we zoom into the numerical interval associated with the cell. (2) If an axis represents temporal data, we zoom into the respective temporal interval and switch to the next finest temporal granularity to be represented by that axis. For instance, zooming into a heatmap cell representing the year 1999 would change the axis granularity to represent months of the year 1999. (3) Nominal data can be zoomed in until the axis represents only one nominal bin. We provide zooming into heatmap row, column, or both, and zoom levels also affect the data displayed in the statistics view. Moreover, it is possible to perform nested zooms (e.g., first zooming into a selected range on the x-axis, and from there, zooming into a selected range on the y-axis). For orientation, we provide breadcrumb-like filter levels below the heatmap, which work as interactive buttons to easily clear selected zoom levels.
Details-on-demand: We implemented different ways to explore details. On the one hand, tooltip texts give further information on hovered heatmap cells. For instance, if the heatmap color is mapped to the number of detected quality problems, the tooltip shows a textual description of the number of prevalent data quality types. On the other hand, the data table (original data entries) and the statistics view (box plots, histograms, and statistics figures) give further details about the selected data from a different perspective.
Relate: We link the representation of detected quality problems to the original data entries in the table. Selecting a heatmap cell not only brushes the corresponding table elements, but also causes the appearance of interactive markers next to the table's scroll bar (similar to TODO-markers implemented in Eclipse IDE [6]), that indicate the location of all corresponding data entries within the table. Clicking on such a marker causes the table to scroll automatically to the clicked table element.
History: We provide a breadcrumb-like history of zoom actions. With this, each zoom level (and filter operation) can be undone.
Extract: The extraction of user-defined quality checks would be a good addition to the functionality of the prototype and is planned in future work.

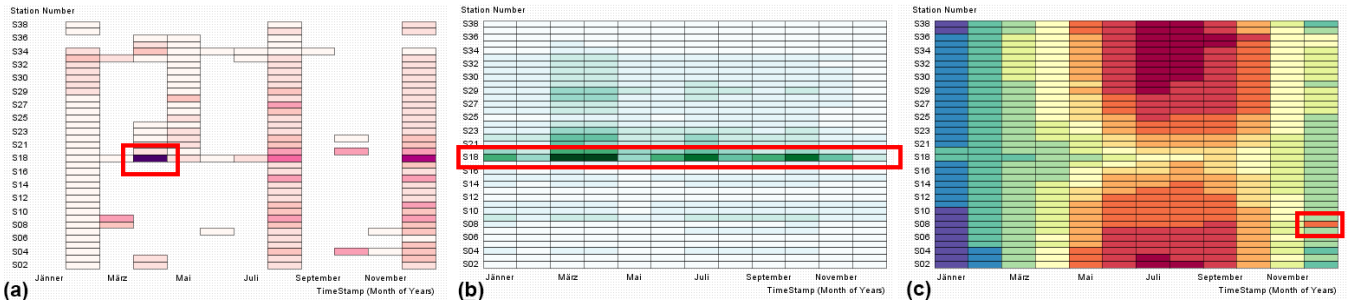

Figure 3: Different ways of color mapping provide different views on the data, relevant to understand possible data quality problems. The data and axes configuration of the heatmap are the same in each picture: months on the x-axis and stations on the y-axis. The color mapping in (a) encodes the amount of detected quality problems for the data in each heatmap cell. This reveals that February, August, and December are especially error-prone for almost all stations. Moreover, there was an unusual amount of errors detected for station S18 in April. In (b) we map the tuple count to color. There are more measurements taken at station S18 than at other stations. (c) shows mapping a calculated key figure, i.e. the mean temperature for each heatmap cell. This points to an outlying high mean temperature in December.

4 Use Cases

In this section we give two examples to illustrate how KYE can be used for different profiling tasks. The first example shows how KYE can visualize already annotated data entries to identify possible causes and relations. The second example illustrates KYE's support for visually detecting additional quality problems that might have been overlooked by automatic means. To this end, we use an adapted data set from U.S. Geological Survey containing measurements about water quality of San Francisco Bay [3]. It contains measurements at specific times, depths and stations providing information about temperature, salinity and oxygen.

4.1 Communicating Automatically Detected Problems

We outline how to use KYE to explore and reason about the data quality problems detected by automatic checks. In Figure 1b we configure the heatmap axes to show the different *stations* (y-axis) in relation to the *time* of measurement (x-axis). This reveals an x-shape which indicates that measurements start at station S1 and end at station S38 (or the other way around) and are taken in sequence. As we are interested in identifying error-prone measurement stations, we map the color to the amount of quality problems detected for each heatmap cell. This reveals prominent dark colors for measurements at station S18 taken around 12.00 noon which means that the vast majority of automatically detected errors happen at this station. When hovering this cell, the tooltip shows that most of these problems are caused by missing values (see Figure 1b). By brushing this cell the corresponding raw data values are highlighted in the linked table view (Figure 1a) and interactive markers next to the scroll bar indicate their position. Clicking on these markers facilitates the navigation to the respective table rows. We investigate the raw data values in the table view and reason that the sensor that measures chlorophyll values must have broken at this day at 33 meter water depth, which caused missing values for the subsequent measurements at deeper water levels. Moreover, the heatmap shows that data quality problems accumulate also for station S19 at 12.00 noon. This might indicated that the failure of the sensor was not noticed immediately and also led to some missing values for measurements at station S19. This visual exploration of automatically detected problems and relating them with the raw data values allow the user to make informed decisions if these detected quality problems are actually wrong data entries or if they are extreme but still valid.

4.2 Identifying Additional Data Quality Problems

After exploring and verifying problems detected by automatic checks, we now freely explore the data to see if there are other problems that were overlooked by the automatic checks. A domain expert might have some guesses where to look for such problems. Another possibility is to explore the data set step by step by trying different axis configurations with different (temporal) granularities and mapping the color to different data values. Figure 4 shows such a setting, mapping *day of month* and water *temperature* to axes and *chlorophyll* values to color. This reveals a suspicious non-empty heatmap cell with salinity values for very low temperatures at the eights day of month (Figure 4a). When selecting this heatmap cell and highlighting the corresponding data entries in the table view, we see that *salinity* values, as well as *oxygen* and *oxygen saturation [%]* were flagged as missing values by automatic checks (yellow background in the table view). However the unusual low *temperature* values were not detected (Figure 4b). Adjacent table rows show temperatures about 10 Celsius, so this is likely to be a data quality problem. Moreover, we spot a non-empty heatmap cell of unusual high temperatures about 27 Celsius on the third day of month (Figure 4c). The histogram in the statistics view reveals another anomaly, which we can also find in the heatmap. There are no measurements at the second day of month (Figure 4d). A side effect of exploring data quality problems with KYE is that an inexperienced user automatically gains a deeper understanding of the data set. For instance, from the given heatmap configuration we learn that *temperature* and *chlorophyll* do not correlate – otherwise the coloring of the heatmap would result in a gradient pattern. The very dark colors of single cells (Figure 4e) give additional hints to outlying *chlorophyll* values which should be further investigated to understand if these values are valid or if they result from data quality problems.

5 Evaluation

To evaluate the usefulness of our prototype and to gain answers to our research questions, we performed a qualitative study with a special focus on revealing insights. To this end, we let six target users execute five tasks covering the different profiling tasks we wanted to support with KYE.

Sample For our evaluation we recruited six target users without any knowledge of our prototype. They qualify as target users as they are all working in IT, they all had to assess the fitness-for-use of data sets before, and they all have some experience with computer-supported data analysis. For data analysis they mainly used Microsoft Excel [19] to identify quality problems or to filter data for management needs.

Data We used the U.S. Geological Survey containing measurements about water quality of San Francisco Bay [3] also for evaluating our prototype. We provided our participants with a subset of the data set, containing the columns: timestamp, station number, depth, chlorophyll, oxygen saturation [%], oxygen, salinity, and

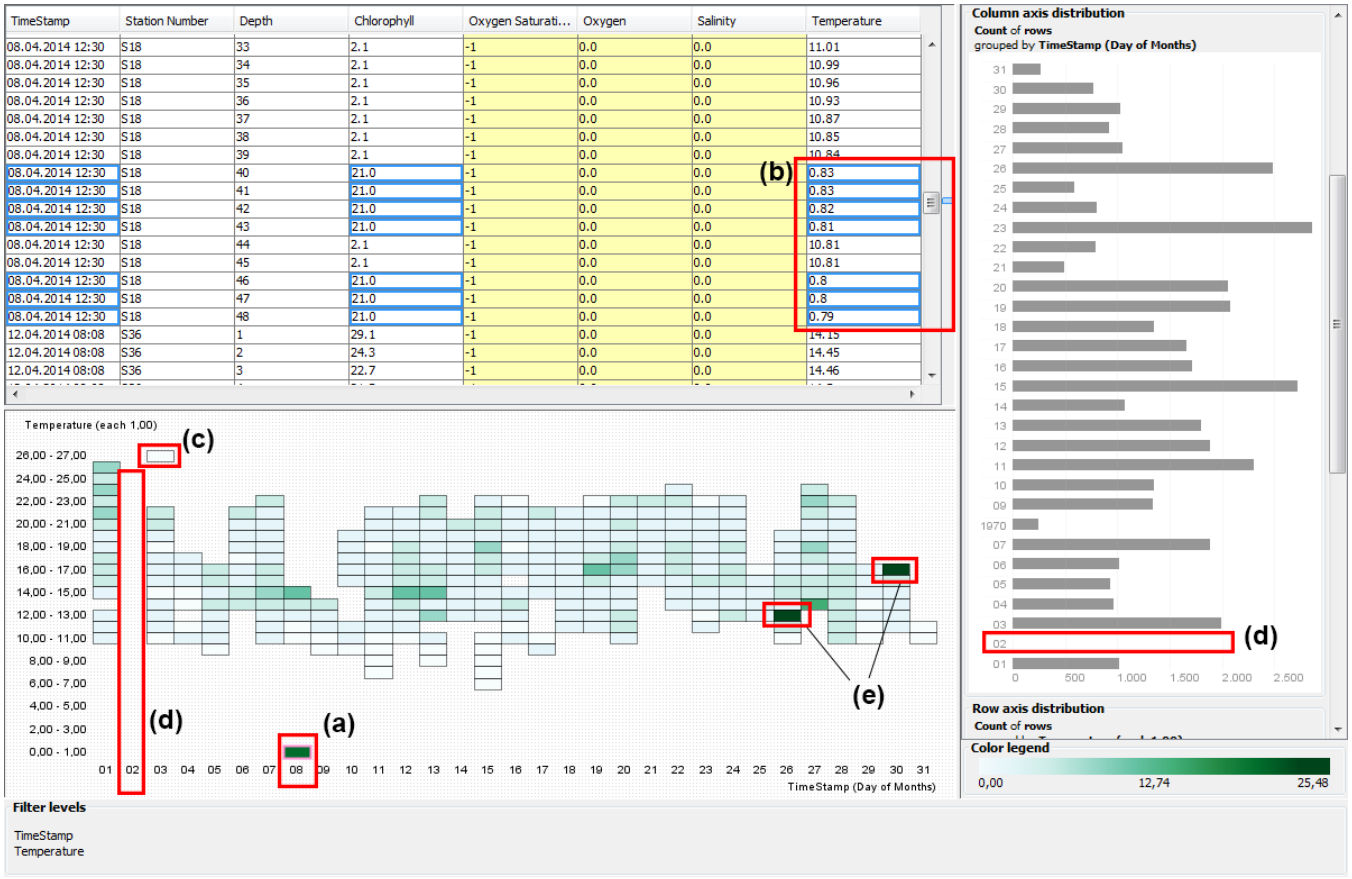

Figure 4: An example of how KYE can be used to detect quality problems that were not detected by automatic checks. The x-axis of the heatmap shows *day of month* and the y-axis shows water *temperature*, while *chlorophyll* values are represented by color. We can identify two suspicious heatmap cells with unusual high (c) and unusual low temperatures (a). When selecting the cell with low temperatures (a), the linked table view shows the raw temperature values (b). Considering the neighbouring temperature values we reason that these unusual low temperatures must be a data quality problem. Moreover, there are no measurements on the 2nd day of month (d). And dark colors in the heatmap point to outlying mean *chlorophyll* values that might also present quality problems (e).

temperature, as well as measurements from a period of five years (2012–2016), so that users were able to get familiar with the data set in the limited time of our study. As we were rather interested in evaluating the interactive visualization and exploration means of KYE and not how well participants could manage to choose automatic tests from the menu and to configure their parameters to their needs, we provided them with a pre-processed data set (i.e., automatic methods were already applied and detected problems were flagged, such as *missing value*, *value too high*, or *unexpected value*.

Tasks For the study we prepared five typical tasks to evaluate the strengths and weaknesses of KYE for profiling of time series data:

T1: Explore automatically detected quality problems. Identify and explore flagged quality problems. Can you identify patterns where or when specific problems occur?

T2. Identify additional quality problems. Identify measurement stations which are most error-prone. Which stations improved or decreased their measurement quality over the last years?

T3. Identifying possible causes. There is a significant amount of data quality problems within some of the stations. Identify the exact time when these problems occur (hint: zoom). Can you find correlations of data quality problems across different stations?

T4. Statistics view. Configure the x-axis to encode the column *timestamp*, as temporal data type with granularity *day of month* and a step-width of 1. Set the y-axis to encode the column *temperature*, with a step-width of 5. What can you tell about the measurement distribution when you look at the heatmap in the statistics view?

T5. Statistics.

T5a. Identify the coldest and hottest month.

T5b. Identify the following statistics of *temperature* and *salinity*: minimum, maximum, mean, median, and standard deviation.

Task 1 is designed to evaluate the visualization techniques of already annotated data entries. Task 2 and 3 are designed to evaluate KYE's means to identify further data quality problems and explore possible causes. Task 4 and 5 cover the statistical information provided by KYE. Moreover, we encouraged participants to take their time for freely exploring the data between or during tasks.

Sessions The sessions were conducted by one developer and one respective study participant in a quiet meeting room. The developer acted as observer for taking notes and supporting the participant if he/she had any logical or technical questions. The test sessions started with an introduction of the prototype and the test data set. Moreover, we conducted semi-structured interviews to learn about their previous experiences with visualization techniques and data analysis. Subsequently, the participants were asked to solve the given tasks using the *thinking aloud* protocol [18] to phrase whatever they encountered or thought while they were working with KYE. After they completed the tasks, we interviewed them about their impres-

sion of the prototype, and how they would rate the usefulness of its visualizations and interaction features. We also asked for possible improvements, missing features, and missing visualizations.

5.1 Results

The data collected from the evaluation study consists of notes of observations, audio recordings of the *thinking aloud*, and answers to interview questions. We analyzed it to understand strengths and shortcomings of KYE and what kind of findings are fostered by working with the interactive visualization. To analyze these findings, we adapted five categories from Klein [16] for gaining insights, and categorized our evaluation results accordingly:
Connection findings result from the identification of a connection between two or more events.
Coincidence findings result from the identification of unexpected relations between events, which results from repetition and not from detail information.
Curiosity findings are findings gained from the exploration of a single event that caused the user's curiosity.
Contradiction findings occur if there is discrepancy between events which causes doubts.
Creative desperation findings result from explorations that lead into dead-ends and requires the user to find new ways.

All tasks have been solved by all participants with an average duration of 60 minutes, while the pre- and post-interviews together took about 50 minutes. In total, we identified 91 findings that were gained by our participants (between 6 and 24 per user). Users who spent more time exploring the data set with KYE gained more findings about the data and its quality problems. We also observed differences in our study participants. One participant was mainly concerned with solving the tasks without much additional exploration, which resulted in the lowest amount of findings. Another participant understood each task as a starting point to dig into the data set, which resulted in the highest amount of findings.

5.1.1 Types of Findings

Most of the findings were *curiosity* findings (31.8%) and *contradiction* findings (29.7%). Less often we could attribute the findings to *connection* (18.7%) or *coincidence* (15.4%). *Creative desperation* was the cause of only 4.4% of findings gained. Moreover, we analyzed which of KYE's features caused how many findings. The heatmap visualization led to the majority of findings (65,9%), the interactive table resulted in 30.8% of all findings, and the statistics view only led to 7.7% of findings (percentages do not sum up to 100% because some findings were gained by using more than one visualization type).

Curiosity (31.8%). The majority of *curiosity* findings were derived from color differences in heatmap cells. For instance, a participant correctly identified that "Station S18 is most error-prone due to its dark coloring" ore "there seams to be a problem in April". Another participant reasoned: "due to the missing cells, we can see that there is no single measurement for the second day of month". Other curiosity findings were gained from exploring the table view discovering colored (flagged with a data quality problem) table entries (e.g., finding missing values while scrolling), or from inspecting the histogram visualization.

Contradiction (29.7%). Again most *contradiction* findings resulted from color differences in the heatmap. One participant reasoned: "temperatures more than 20C in December seem like a measurement error". Another participant noted: "there are measurements with more than 100% of oxygen saturation, which seems unrealistic". Other *contradiction* findings resulted from the table view. One participant noticed: "it seems odd that there are 7 measurements about 0 Celsius between regular 10 Celsius measurements".

Connection (18.7%). We could identify two types of frequent *connection* findings: The first one is finding a pattern by connecting different views, for instance, selecting a heatmap cell and cross-checking related data table entries ("the sensor must have been broken at 33 meter depth") or the histogram view. The second type resulted from connections between values. One participant noticed: "station S18 only measures exactly 27.0 Celsius for each measurement at that specific day".

Coincidence (15.4%). *Coincidence* findings included, for instance, this observation of one participant: "[...] it seems that the measurement sensor was stuck because there are so many following measurements with 17.1 Celsius temperature". Other *coincidence* finding resulted from reoccurring types of specific quality problems detected by automatic checks. For instance, users figured that a specific station was generally error-prone.

Creative Desperation (4.4%) *Creative desperation* findings occurred less often. One participant randomly scrolled through the table and found annotated values, after not being able to investigate these numbers with the current heatmap configuration. Other *creative desperation* findings happened while randomly configuring the heatmap, which lead to the identification of implausible negative salinity values.

5.1.2 Feedback from Post-Interviews

In post-interviews, participants mentioned a number of positive aspects about KYE. They especially liked the flexible configuration of the heatmap, which allows to set any data dimensions into relation and they found it well suited to identify possible data quality problems. Moreover, they thought the heatmap was well suited to give an overview of the data and it allows for the identification of temporal patterns, for instance, temperature changes over years. The combination of the heatmap with the table view was appraised as very useful. Brushing and linking between the heatmap and the table facilitated the verification of possible quality problems.

When asking for possible improvements, they mentioned a functionality to zoom back out after zooming into a heatmap cell, a possibility to map the number of identified quality problems on heatmap axes (not only on color), making histograms more interactive, as well as means to apply filters before even starting the exploration (e.g., filter by a specific station). Problems included the necessity to frequently switch between the heatmap visualization and the configuration panel. Moreover, the flexible mapping of different data aspects to color might lead to confusion of what is currently displayed be the heatmap. A possible solution is to use different color scales for different data aspects.

6 Discussion and Lessons Learned

In this section we discuss how we met our design requirements defined in Section 3.1 and contemplate on lessons learned from designing and evaluating KYE. We carefully considered and analyzed the results from the evaluation as well as the feedback from post-interviews, which led to the identification of positive but also negative issues. While we appreciate the positive feedback from study participants and the number of findings gained with KYE, which suggests its suitability for data profiling tasks, we also identified possibilities for improvement. From this experience we derive further research challenges in the field of VA for data quality.

6.1 Meeting Design Requirements

In Section 3.1 we defined four requirements that should be met by the design of KYE.

R1 Data quality checks: KYE provides a number of predefined, ready-to-use, automatic quality checks as well as the possibility to define customized quality checks (see Section 3.3).

R2 Exploration of quality problems: Figure 1 shows how KYE supports the exploration of automatically detected data quality problems by highlighting these problems in the heatmap and investigating the problematic data entries in the linked table.

R3 Identification of hidden problems: Figure 4 shows an example of how KYE supports the identification of hidden problems. Problems that are usually better detected with visual means than with automatic checks include any data entries that are not out of bounds but seam implausible due to their position in the time series or their neighboring values.

R4 Scalability: KYE's heatmap provides visual and logical aggregation of the data as well as mapping calculated key figures to the color of heatmap cells (see Section 3.4.1). These possibilities for aggregation in combination with the possibility to zoom into heatmap cells and providing details on demand guarantee a salable overview of possible huge data sets.

6.2 Lessons Learned

One lesson we derived from our observations is that designing a flexible approach applicable in many data domains comes with the drawback of additional user burden. Users need to learn how to configure the visualization to serve their needs. This configuration could be built in if the approach needs to serve only one domain with well-defined tasks. However, for a general approach, this trade-off between flexibility and the user's learning costs needs to be carefully weighted. The approach presented by KYE has the advantage that it provides one main visualization that works for many data types and tasks. Thus, the user does not have to learn how to configure a multitude of different visualization types. Moreover, in the case of data profiling, trial-and-error configurations may also lead to interesting findings (see Section 5.1).

Another thing we learned from the evaluation of KYE is that we cannot expect our target users (i.e., anyone who needs to assess if the quality of a given data set is sufficient for his/her needs) to benefit from basic statistics information. The statistics view was not as helpful as the other visualizations provided. It only had an effect on 7.7% of all findings (see Section 5.1). One test participant considered the statistics view so irrelevant, he suggested a functionality to minimize it. It is evident from our evaluation that KYE's heatmap visualization in combination with the interactive raw data table is sufficient to solve the given tasks and gain a majority of findings.

Considering the identification of data quality problems it became apparent that the combination of visualizations with raw data values is imperative. Visualizations are essential to understand big data sets and find peculiarities in the data that require further investigation. However, at a certain level all study participants used the table view to look at the raw data values to understand if something represents an actual data quality problem or not. One does not work without the other for the task of data profiling: it is no surprise that huge data tables are not well suited to identify peculiarities, but on the other hand, visualizations alone are not sufficient to decide about data errors. However, suitable visualizations are essential to find hints and to formulate hypotheses.

For the identification of data quality problems in particular in time series data, the neighbouring values in the data table, i.e. the sequence of values at the finest possible granularity, was a major factor for deciding about value validity. While aggregation is essential to visualize big data sets, it might also obfuscate small differences at the level of individual data entries. Thus, it is important to enable a visual exploration at different (temporal) granularities and different viewpoints (i.e., let the user explore different aspects of the data, different key figures, at different aggregations).

Another thing we learned in the course of our evaluation, is that although the heatmap can be used to identify cyclic behavior, this is only true for cycles that follow a calendar based structure (e.g., monthly patterns, yearly patterns), since heatmap axes can only be configured with these calendar based instances. To be able to identify cyclic behaviour with other cycle lengths, for instance a 17 days cycle, we would need to provide the possibility to configure the number of instances displayed at one axis. This would enable users to identify possible quality problems of additional types of time series data.

6.3 Further Research Challenges

In the context of this paper we have designed and evaluated a VA approach for data profiling. While we provide predefined automatic quality checks and the possibility for users to formulate specific automatic checks, we did not focus on supporting the application of these checks with VA methods (by now users simply choose the respective quality checks from the menu and configure its parameters in a dialog window, or they enter regular expressions). Yet, for a comprehensive VA data profiling solution, this process should be better integrated with the actual visualization. Thus, one further research challenge in this field is the design and evaluation of **better ways to formulate, configure, and apply automatic quality checks, supported by VA methods**. While KYE is suited to cover the majority of data quality issues from a single data source (as outlined in Gschwandtner et al. [9]), there are some **quality problems that cannot be easily detected**, neither with automatic nor with visual methods. For instance, we do not support the task of understanding if a given data entry is conform to the real entity it refers to (e.g., the data contains an entry that Suzanne lives in Boston while she actually lives in New York). Other quality problems we cannot support yet are incorrectly derived values and ambiguous data. These require other means with cross-checks to other data sources and advanced AI techniques.

While we were tackling in particular the problem of data profiling, data quality management also includes **data cleansing** and **data transformation**. When dealing with time series data this holds specific difficulties. It is important to consider the different kinds of **temporal dependencies of data values**. The number of items sold within one hour has other plausible bounds than the number of items sold within one day. Other data values again behave differently: the number of staffs in a shop may stay the same whether considering one hour or the whole day. Cleansing erroneous values and replacing them with an estimated value requires the consideration of these dependencies. A related challenge is the transformation of time series data. Also when **transforming data** with unevenly spaced time-stamps or intervals with different lengths **into evenly spaced intervals**, these temporal dependencies need to be considered. If not supported correctly, this transformation of time series data may even introduce uncertainties and additional quality issues into the data. This is an open research challenge which asks for sophisticated VA support.

On a more general note, we identified research challenges that are not specific to time series data but can be generalized to profiling of all kinds of data. For instance, the support for user-defined quality metrics is still an open challenge. Grouping quality checks into quality metrics, such as completeness, validity, or plausibility, would allow users to create a re-usable expert tool set that fits their specific needs for more efficient data profiling of similar data sets and tasks. The **visual-interactive definition and fine-tuning of quality metrics** would be an interesting topic for further research.

Another general data profiling challenge is the **integration of expert knowledge**. By now, we include the expert and his/her domain knowledge (which is crucial for assessing the quality of the data) by interactively working with the prototype and accepting or discarding automatically detected problems. A formalization of this knowledge, however, would considerably reduce the expert's efforts. This could be accomplished by learning systems or methods that

support the input of expert knowledge.

7 Conclusion

We presented the design and evaluation of KYE, a VA solution for data profiling with a special focus on supporting time series data. The purpose of KYE is three-fold: (1) communicating automatically detected quality problems (by means of predefined checks and the possibility for user-defined checks), (2) enabling users to investigate these problems and reason about them, and (3) making the data set visually explorable in a way that fosters the identification of further quality problems overlooked by automatic methods. We thoroughly considered possible design alternatives and decided for a two-dimensional heatmap visualization in combination with a table view and a statistics view. We employed an iterative design process and we evaluated the prototype with respect to its potential to identify data quality problems and to reveal new findings. This evaluation indicates that KYE is well-suited to support the task of data profiling of time series data. KYE was especially appreciated for its flexibility and the connection of visual items to original data entries. However, we also learned a number of lessons from this study. Most importantly, it became evident how crucial it is to carefully consider the trade-off between flexibility of the provided solution and the additional learning effort it takes for users to configure the solution to their specific needs. By providing only one visualization that needs to be understood and configured, we minimize the required learning effort, albeit it cannot be avoided completely. In conclusion, data quality management of time series data is a fascinating topic that still holds many unresolved problems. While we have successfully tackled the first step (i.e., data profiling), data cleansing and data transformation also require special consideration. Correctly handling time dependent values in cleansing and transformation operations, and VA support for normalizing unevenly spaced time series are examples of further extensive research challenges.

Acknowledgments

The Laura Bassi Centre of Expertise CVAST is funded by the Austrian Federal Ministry of Science, Research, and Economy (project number: 822746).

References

[1] C. Arbesser, F. Spechtenhauser, T. Mühlbacher, and H. Piringer. Visplause: Visual data quality assessment of many time series using plausibility checks. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):641–650, 2017. doi: 10.1109/TVCG.2016.2598592

[2] J. Bernard, T. Ruppert, O. Goroll, T. May, and J. Kohlhammer. Visual-interactive preprocessing of time series data. In *Proc. of SIGRAD 2012: Interactive Visual Analysis of Data*, pp. 39–48, 2012.

[3] J. E. Cloern and T. Schraga. USGS measurements of water quality in San Francisco Bay (CA), 1969-2015. `https://pubs.er.usgs.gov/publication/70179097`. Retrieved at Dec 18, 2017. doi: 10.5066/F7TQ5ZPR

[4] Data Manager. Data transformation, cleaning & cleansing. `http://datamanager.com.au/`. Retrieved at Sep 27, 2017.

[5] Datamartist. Data Profiling Tool. `http://www.datamartist.com/`. Retrieved at Sep 27, 2017.

[6] Eclipse Foundation. Eclipse. `https://www.eclipse.org/`. Retrieved at October 18, 2016.

[7] H. Galhardas and J. Barateiro. A survey of data quality tools. *Datenbank-Spektrum*, 1:14:15–21, 2005.

[8] T. Gschwandtner, W. Aigner, S. Miksch, J. Gärtner, S. Kriglstein, M. Pohl, and N. Suchy. TimeCleanser: A visual analytics approach for data cleansing of time-oriented data. In *Proceedings of the 14th International Conference on Knowledge Technologies and Data-Driven Business*, i-KNOW '14, pp. 18:1–18:8. ACM, 2014. doi: 10.1145/2637748.2638423

[9] T. Gschwandtner, J. Gärtner, W. Aigner, and S. Miksch. A taxonomy of dirty time-oriented data. In *Multidisciplinary Research and Practice for Information Systems*, Lecture Notes in Computer Science (LNCS) 7465, pp. 58–72. Springer, 2012.

[10] M. Harrower and C. A. Brewer. ColorBrewer.org: An online tool for selecting colour schemes for maps. *The Cartographic Journal*, 40(1):27–37, 2003. doi: 10.1179/000870403235002042

[11] D. Hoang. DataLadder - DataMatch 2017. `https://dataladder.com/data-matching-software/`. Retrieved at Sep 27, 2017.

[12] S. Kandel, J. Heer, C. Plaisant, J. Kennedy, F. van Ham, N. H. Riche, C. Weaver, B. Lee, D. Brodbeck, and P. Buono. Research directions in data wrangling: Visualizations and transformations for usable and credible data. *Information Visualization*, 10(4):271–288, 2011. doi: 10.1177/1473871611415994

[13] S. Kandel, A. Paepcke, J. Hellerstein, and J. Heer. Wrangler: Interactive visual specification of data transformation scripts. In *Proc. of the ACM Conference on Human Factors in Computing Systems (CHI 2011)*, pp. 3363–3372. ACM, May 2011.

[14] S. Kandel, R. Parikh, A. Paepcke, J. Hellerstein, and J. Heer. Profiler: Integrated statistical analysis and visualization for data quality assessment. In *Proc. of the International Working Conference on Advanced Visual Interfaces (AVI'12)*, pp. 547–554, May 2012.

[15] W. Kim, B. J. Choi, E. K. Hong, S. K. Kim, and D. Lee. A taxonomy of dirty data. *Data Mining and Knowledge Discovery*, 7:81–99, 2003. doi: 10.1023/A:1021564703268

[16] G. Klein. *Seeing What Others Don't: The Remarkable Ways We Gain Insights*. PublicAffairs, 2013.

[17] T. Lammarsch, W. Aigner, A. Bertone, J. Gärtner, E. Mayr, S. Miksch, and M. Smuc. Hierarchical temporal patterns and interactive aggregated views for pixel-based visualizations. In *Proc. of the 13th International Conference Information Visualisation ((IV09))*, pp. 44–50. IEEE, 2009. doi: 10.1109/IV.2009.52

[18] C. H. Lewis. Using the "thinking aloud" method in cognitive interface design. Technical report. IBM. RC-9265, 1982.

[19] Microsoft Excel. Spreadsheet Software 2016, Excel Free Trial. `https://products.office.com/en/excel`. Retrieved at July 14, 2017.

[20] S. Miksch and W. Aigner. A matter of time: Applying a data–users–tasks design triangle to visual analytics of time-oriented data. *Computers & Graphics*, 38:286–290, 2014. doi: 10.1016/j.cag.2013.11.002

[21] D. Mintz, T. Fitz-Simons, and M. Wayland. Tracking air quality trends with SAS/GRAPH. *Proc. of the 22nd Annual SAS User Group International Conference (SUGI)*, 1997.

[22] T. Munzner. A nested model for visualization design and validation. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):921–928, 2009. doi: 10.1109/TVCG.2009.111

[23] J. E. Olson. *Data Quality: The Accuracy Dimension*. Morgan Kaufmann, San Francisco, 1st ed., 2002.

[24] G. H. Orair, C. H. C. Teixeira, W. Meira, Jr., Y. Wang, and S. Parthasarathy. Distance-based outlier detection: Consolidation and renewed bearing. *Proc. VLDB Endow.*, 3(1-2):1469–1480, 2010. doi: 10.14778/1920841.1921021

[25] E. Rahm and H. Do. Data cleaning: Problems and current approaches. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 23:3–13, 2000. doi: 10.1145/1317331.1317341

[26] B. H. S. Havre and L. Nowell. ThemeRiver: Visualizing theme changes over time. In *Proc. of the IEEE Symposium On Information Visualization (InfoVis)*, pp. 115–123, 2000. doi: 10.1109/INFVIS.2000.885098

[27] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proc. of the IEEE Symposium Visual Languages*, pp. 336–343, 1996. doi: 10.1109/VL.1996.545307

[28] Talend. Open Studio Integration Software Platform. `https://www.talend.com/products/talend-open-studio`. Retrieved at Sep 27, 2017.

[29] C. Tominski and W. Aigner. The TimeViz Browser: A visual survey of visualization techniques for time-oriented data. `http://browser.timeviz.net/`. Retrieved at Dec 18, 2017.

[30] R. Verborgh and M. D. Wilde. *Using OpenRefine*. Packt Publishing, 1st ed., 2013.

[31] M. Weber, M. Alexa, and W. Muller. Visualizing time-series on spirals. In *Proc. of the IEEE Symposium on Information Visualization (InfoVis)*, pp. 7–13, 2001. doi: 10.1109/INFVIS.2001.963273