Usability of Visual Data Profiling in Data Cleaning and Transformation

Bjørn Marius von Zernichow

MSc at Department of Informatics

UNIVERSITY OF OSLO

2017

Usability of Visual Data Profiling in Data Cleaning and Transformation

Bjørn Marius von Zernichow

2017

2017

Usability of Visual Data Profiling in Data Cleaning and Transformation

http://www.duo.uio.no/

Print: Reprosentralen, Universitetet i Oslo

Abstract

Data collection has become a necessary function in most large organizations both for record keeping and in support of different data analysis activities that are strategically and operationally critical. In this context, proper data quality is a crucial aspect of extracting accurate information from data sources. Hence, incorrect, or inconsistent data may distort analysis and compromise the benefits of any data-driven approaches. To illustrate the impact of poor-quality data, IBM has estimated the yearly cost to be $3.1 trillion in US in 2016. Furthermore, recent surveys show that data scientists spend most of the time on cleaning and organizing data, and consider this work to be repetitive and tedious activities. Such estimates indicate that novel approaches and solutions for improving data quality are needed and can have significant impact in practice.

Among approaches to improve data quality, visual data profiling is the statistical assessment of datasets to identify and visualize potential quality issues such as data outliers or missing data values. Visual data profiling has the potential to help data scientists make an informed decision on how to deal with data quality issues. This thesis positions itself within the research area of exploratory data analysis and visual data profiling by providing data scientists an approach that simplifies data cleaning and transformation processes, thereby contributing to solutions that improve data quality.

The proposed approach is realized in a software prototype that, among others, identifies and visualizes data quality issues in tabular data. The approach, together with the associated prototype, have been empirically validated to determine to which extent visual data profiling approaches are useful and easy to use by data scientists. The validation process included a comparative usability test and survey to compare the prototype against an existing approach to data cleaning and transformation in terms of usefulness and ease of use. Finally, two expert reviews were conducted to identify usability issues introduced by the proposed visual data profiling approach in data cleaning and transformation processes. Based on this evaluation, future research opportunities are identified for improving and extending the proposed visual data profiling approach.

Acknowledgements

I would like to express my gratitude to everyone who contributed to the process of writing my thesis.

First, I will thank my main supervisor Dumitru Roman, and supervisor Nikolay Nikolov at SINTEF Digital – Smart Data – for their invaluable guidance, patience, motivation, and contributions to scientific and technical discussions, ideas, and academic writing. They have openly invited me to be part of the research environment at SINTEF that has been so important to steer the thesis process in the right direction.

Furthermore, I will also thank the remaining Smart Data team and involved employees at SINTEF Digital for their immense guidance and support.

Second, I would like to extend my gratitude to the LogID group at University of Oslo, and the HCI group at SINTEF, that made a valuable contribution to the evaluation of the approach that has been proposed as part of my thesis.

Third, I will show my appreciation to everyone at proDataMarket, EW-Shopp, and euBusinessGraph project meetings who participated in the comparative usability test and survey.

Finally, I would like to thank my family and friends for their support and understanding.

Contents

1 Introduction 13

1.1 Overall Context 13

1.2 Thesis Motivation 14

1.3 Research Questions 15

1.4 Thesis Contributions 17

1.5 Research Design 18

1.6 Thesis Outline 22

2 Related Work 25

2.1 Basic Data Profiling Cycle 25

2.2 Visual Data Profiling Tasks 28

2.3 Characteristics of Data in the Context of Data Profiling 29

2.4 Data Quality and Data Anomalies 30

2.5 Statistical Charts used in Data Profiling 34

3 Problem Analysis 41

3.1 Usability Testing 41

3.1.1 Identifying the Users of the Visual Data Profiling Approach 43

3.1.2 Defining the Usability Testing Strategy 45

3.2 Evaluation of State of the Art Approaches 51

3.3 Requirements 56

4 Architecture and Implementation 61

4.1 Architecture 61

4.2 Functions 67

4.3 Implementation 69

5 Evaluation 79

5.1 Validation of Functional Requirements 79

5.2 Validation of Usability Requirements 82

5.2.1 Analysis of Findings from the Comparative Usability Test 82

5.2.2 Analysis of Findings from the Cognitive Walkthrough 88

6 Conclusion 93

6.1 Evaluation of the Visual Data Profiling Approach in Terms of Requirements 93

6.2 Further Work 95

References 99

Appendices 103

List of Figures

Figure 1: User-centered design process......20
Figure 2: First two steps of visual data profiling cycle......26
Figure 3: Last step of visual data profiling cycle......27
Figure 4: Data distribution chart......36
Figure 5: Boxplot chart......38
Figure 6: Numeric histogram chart......39
Figure 7: User persona......44
Figure 8: Survey example question......48
Figure 9: Survey feedback cycle......48
Figure 10: Grafterizer user interface......54
Figure 11: Visual data profiling approach wireframe......57
Figure 12: 3-Tier architecture......62
Figure 13: Visual Data Profiling microservice architecture......63
Figure 14: Angular 2 prototype architecture......64
Figure 15: Angular 2 component class structure and metadata......65
Figure 16: Angular 2 template structure and data binding......65
Figure 17: Angular 2 data binding......66
Figure 18: Angular 2 interpolation......66
Figure 19: Angular 2 two-way data binding......67
Figure 20: Implementation of prototype, 1st iteration......72
Figure 21: Implementation of prototype, 2nd iteration......73
Figure 22: Implementation of prototype, 3rd iteration......74
Figure 23: Visual data profiling rules matrix......75
Figure 24: Visual data profiling statistical assessment profile......76
Figure 25: Implementation of prototype, final iteration......77
Figure 26: Comparative usability test results (usefulness)......83
Figure 27: Comparative usability test results (ease of use)......84

Figure 28: Data profiling visual recommender system pipeline 96
Figure 29: Scatter plots – positive (a) and negative (b) correlations between attributes...97
Figure 30: Example of user defined Clojure code in Grafterizer 103

List of Tables

Table 1: Mixed methods strategy..21

Table 2: Requirements..59

Table 3: Validation of visual data profiling approach 80

Table 4: Validation of functional requirements..81

Table 5: Survey rating scores, and difference, in terms of usefulness 86

Table 6: Survey rating scores, and difference, in terms of ease of use...................87

Table 7: Identified usability issues and suggestions for further research............90

List of Equations

Equation 1: Mean 34

Equation 2: Median 35

Equation 3: Interquartile range (IQR) 36

Equation 4: Suspected outliers 36

Equation 5: Variance 37

Equation 6: Standard deviation 37

Equation 7: Paired t-test 85

Equation 8: TDIST function 85

1 Introduction

1.1 Overall Context

Data collection has become a necessary function in most large organizations both for record keeping and in support of different data analysis activities that are strategically and operationally critical [1]. In this context, proper data quality is a crucial aspect of extracting accurate information from data sources. Hence, incorrect or inconsistent data may distort analysis and compromise the benefits of any data-driven approaches. Examples of data quality issues, also labeled anomalies, include occurrences of missing, extreme, erroneous and duplicate values [2].

To illustrate the impact of poor quality data, IBM has estimated the yearly cost of inadequate data quality to be $3.1 trillion in US in 2016 [3]. Further, data scientists spend 60% of their time on cleaning and organizing data, and 57% ranked this as a repetitive and tedious activity [4].

Considering the potential negative impact of poor data quality, there has been considerable research during the last decades, and different methods and tools have been proposed to cope with data cleaning [1]. **Data cleaning** is the process and techniques of identifying and resolving missing values, outliers, inconsistencies, and noisy data, to improve data quality [5]. Closely related to data cleaning processes, additional **data transformation** procedures, i.e. changing the data format while preserving the original meaning, are often required to improve data quality [5].

Despite considerable research recent years to suggest approaches that can improve data quality, there are still opportunities for research to propose solid solutions that will improve data quality and make cleaning and transformation processes more efficient [1]. The broad range of approaches to improving data quality includes suggesting data entry interface designs that prevent incorrect entries in databases, and data quality management solutions that focus on providing incentives to improve data quality. Furthermore, exploratory data analysis and cleaning approaches, together with automated data auditing and cleaning solutions, have been proposed to assist users in the process of improving data quality [1].

This thesis positions itself within the research area of exploratory data analysis by providing an approach that simplifies the data cleaning and transformation process, and reduces effort spent on preparing data for analysis.

1.2 Thesis Motivation

Data profiling is the statistical assessment of data sets to identify potential quality issues such as outliers or missing values. The proposed approach involves data profiling techniques that may be a key factor in achieving improved data quality [2]. Since determining what defines an error is context-dependent, human judgment is usually involved to determine whether the issues are actual errors and how the issues should be treated. The data quality assessment can be facilitated by a data profiling tool that performs statistical analysis [2], [5].

Visual data profiling is an extension of data profiling approaches, achieved by supplementing statistical assessment of data sets with adequate visualizations [2], [6]. The integration of statistical analysis and visual analysis can reduce the time users spend on exploring and assessing data quality issues by providing constant real-time feedback on content and structure of the data set. Considering that data scientists use more than half of their time cleaning and organizing data, and often find this activity tedious, visual data profiling approaches should be considered to potentially increase data quality, and reduce time and cost of work activities.

In terms of **user acceptance** of a system, it is essential that users believe that the system is useful and easy to use in order to adopt the technology [7], [8]. Hence, a visual data profiling extension should not only provide the capabilities that the user needs, but the extension should also be considered useful in data scientists' work activities, and be easy to use [7], [9]. We will for now refer to these qualities as the **usability** of the visual data profiling system.

This thesis explores usability of visual data profiling by proposing an approach that is evaluated with users in a data cleaning and transformation context. In search for an existing data cleaning and transformation solution that could benefit from visual data profiling, Grafterizer [10] was selected as a starting point to realize the visual data profiling approach by developing a prototype.

Grafterizer is a web-based framework on the DataGraft platform for data cleaning and transformation [10]. The framework represents state of the art within data cleaning and transformation research, but does not yet offer data profiling capabilities. Grafterizer provides in this sense good research opportunities for evaluating usability of visual data profiling since the current version serves as a benchmark in a comparison with the proposed prototype.

DataGraft is a cloud-based platform for hosted open data management, data transformations and publishing [10]. The concept of open data corresponds to the data that government and non-government institutions make available under open licenses. DataGraft is an integrated self-service solution that lets *data consumers* utilize open data for data-driven decision making instead of searching for data. On the other hand, *data publishers* can focus on providing high quality datasets instead of developing and managing their own platforms for publication and hosting. As a result, cost and time consuming processes might be reduced.

1.3 Research Questions

The scope of the thesis is to explore usability of visual data profiling in *tabular data cleaning and transformation processes*.

To understand users' experience with visual data profiling approaches, we will need to define who are the typical users. User profiles are discussed in detail later in this thesis, and we will for now *define our users as data consumers, more specifically data scientists, that use data for data-driven decision making.*

The data scientist[1] is an analytical expert that explores and analyzes large volumes of data to solve complex problems and reveal business insights. Dedicated solutions for cleaning and transforming tabular data, e.g. Grafterizer, are often part of a data scientist's toolbox.

Some assumptions have been made to guide the choice of purpose statement and research questions. We will assume that:

- Visual data profiling can improve data quality [2], [6], [11] by providing statistical analysis and assessment of data quality. The user, or a system, will utilize this information to make an informed decision on how to treat data quality issues.

- Data profiling systems should be perceived as useful and easy to use [7]. A user will consider a system to be *useful* if it enhances his or her work performance, and a system is *easy to use* if a user thinks that learning and using the system requires an acceptable amount of effort in terms of time and cost [7], [8].

A qualitative **purpose statement** [12] can be formulated as follows:

> The purpose of this thesis is to explore usability of visual data profiling in tabular data cleaning and transformation processes to improve data quality in the context of Grafterizer.

Based on the purpose statement, the guiding central **research questions** [12] in this study are:

1. What **visual data profiling approach**, realized through a prototype, can be proposed to **evaluate usability of visual data profiling** in tabular data cleaning and transformation?

2. **How *useful* are visual data profiling approaches** for users of tabular data cleaning and transformation tools?

[1] https://www.sas.com/en_us/insights/analytics/what-is-a-data-scientist.html

3. **How *easy to use* are visual data profiling approaches** for users of tabular data cleaning and transformation tools?

4. **Will visual data profiling approaches introduce usability issues** in tabular data cleaning and transformation applications, and if so; which types of usability issues occur and how can they be corrected?

1.4 Thesis Contributions

This thesis contributes to exploring usability of visual data profiling by providing an approach which is evaluated by means of a prototype that implements the suggested approach. The approach extends the research [10], [13] behind the current version of Grafterizer to include data profiling capabilities. The extended capabilities provided by the approach could ease the process of data cleaning and transformation, and improve data quality, for data scientists. This will be the basis for a powerful visual data profiling assisted data cleaning and transformation framework that will contribute to improving current state of the art, and provide important insights to research within the field of usability of visual data profiling.

Summary of Thesis Contributions

The thesis contributes to providing:

- A discussion of data quality and common data quality issues, and how this is related to visual data profiling.

- An evaluation of current state of the art solutions within visual data profiling, and data cleaning and transformation.

- An approach to using visual data profiling in tabular data cleaning and transformation processes to improve data quality.

- Realization of the visual data profiling approach by means of a prototype that includes features for identifying and visualizing data quality issues, i.e. missing values and outliers.

- An evaluation of the visual data profiling approach by empirical validation of the prototype. A comparative usability study and survey are used to compare the approach against the current version of Grafterizer in terms of usefulness and ease of use.

- Suggestions for future research within visual data profiling approaches based on the results of the evaluation that identify usability issues in the prototype.

1.5 Research Design

According to Venkatesh et. al [14] the choice of research methodology should be based on the research question, purpose and context. All research questions involve qualitative exploration where qualitative methods [12] would be suitable, and the implementation of a prototype to realize an approach to visual data profiling would fall into the category of technology research. Solheim and Stølen [15] define *technology* as 'the knowledge of artefacts emphasizing their manufacturing', and differentiate between two variants of research:

a. **Classical** research with the purpose of obtaining knowledge about what exists.

b. **Technology** research with the purpose of developing new and better artefacts.

The iterative technology research process [15] starts with a *problem analysis* to identify a potential need, and proceeds to the *innovation* stage where a technological artefact is developed. Finally, the artefact enters the evaluation stage to validate whether it satisfies the need.

The methodology will be extended to include both quantitative and qualitative methods in a *mixed methods research approach* [12], [14]. A mixed methods approach uses multiple methods, i.e. includes more than one method that can be quantitative or qualitative. One main reason for selecting this research design, is that *triangulation* of quantitative and qualitative data can be used to potentially provide stronger inferences than one single method would [14].

The results from qualitative methods are used to corroborate and assess the credibility of inferences obtained from the quantitative methods by providing complementary views and additional insight.

Epistemologically, the research in this thesis is mainly grounded in a pragmatic worldview [12], assuming that a combination of both quantitative and qualitative methods provides a more complete understanding of the research problem.

The activities that are involved to develop an artefact include requirements specification, design, implementation, and validation [16]. When discussing development models, we will consider an artefact to be a type of software to be consistent with Sommerville's terminology of software engineering [16]. First, the functionality and constraints of software must be defined. Second, the software is designed and implemented according to the requirements. Finally, the software is validated to ensure that it meets the expectations of the user.

Basically, there are two types of software process models. The traditional *waterfall* model treats each software development activity as a separate stage that follows sequentially, e.g. validation is not started until implementation has finished [16]. One of the disadvantages of using this model, is that it is difficult to get user feedback during implementation, and it could be risky and costly to wait with user feedback until the software is fully implemented [16].

Hence, an *incremental* software development model would be more suitable in terms of developing the prototype that supports the visual data profiling approach. An incremental process interleaves the development activities of requirements specification, design, implementation, and validation, and provides continuous feedback across activities [16]. The advantage of using this development model is reduced cost of implementing changes, and quicker access to user feedback.

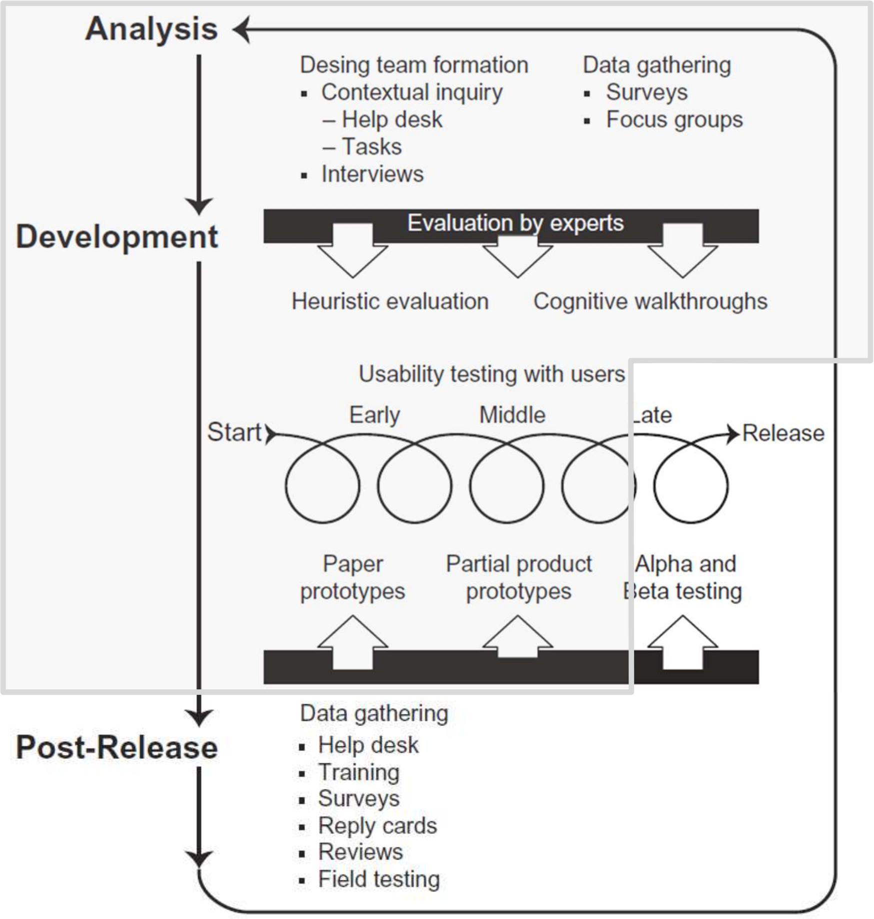


Figure 1: User-centered design process

Figure 1 shows an incremental development process, a user-centered design process [8], that is adopted in this thesis. The process is selected because of its user-centric, incremental organization of activities that are specifically suited for evaluating usability. The scope of this thesis is contained within the gray overlay box shape in Figure 1.

Applying this user-centered design process, we start with the problem analysis phase in Chapter 3 to define users and a usability testing strategy, and evaluate state of the art approaches. Finally, the identified needs of the users lead to a set of requirements.

The prototype is iteratively implemented in Chapter 4, and evaluated in Chapter 5. As can be seen from Figure 1 (the middle section indicated by a spiral), the iterative development process involves the use of prototypes, and expert evaluations in a usability testing method called cognitive walkthrough. Most of the activities discussed in Chapter 4 and 5 are part of this iterative cycle.

Mixed Methods Strategy

Considering the technology research process and user-centered design process, the following methods have been used in this thesis:

Table 1: Mixed methods strategy

STAGE	METHOD	*	**
PROBLEM ANALYSIS	Literature review incl. evaluation of related approaches	🟩	
IMPLEMENTATION + EVALUATION	Prototyping	🟩	
	Comparative usability test/ Survey	🟩	🟩
	Cognitive walkthrough	🟩	

Qualitative method *

Quantitative method **

The activities in Table 1 [17] are carried out in an *exploratory sequential mixed methods design* approach [12]. Findings from one stage inform the next stage and add overall richness to the study [14].

Next is a brief introduction to each of the methods and how they will be applied in this thesis. All methods are essential to the user-centered design process in Figure 1. The methods of prototyping, survey, and cognitive walkthrough are discussed in detail later in this thesis.

- **Literature review** is conducted to synthesize information from different academic sources, and ensure that existing solutions and approaches are taken into consideration [17]. The review also includes an evaluation of relevant approaches, such as software and applications described in the literature.

- **Prototyping** is applied as an iterative design and development process to realize concepts and requirements that are defined in the proposed visual data profiling approach [17]–[19]. By prototyping, we will always have something functional to test with users, collect feedback, implement changes, and then iterate.

- **Comparative usability test, survey based** is used to collect statistics and attitudinal data from users through an online questionnaire [20] which contain Likert-type rating scales. The test will compare the prototype against the current version of Grafterizer in terms of usefulness and ease of use [20]. The survey is anonymized and voluntary, and only non-sensitive information is collected.

- **Cognitive walkthrough** is a usability inspection method where evaluators inspect the user interface by completing a set of tasks to simulate users' problem solving approaches [21]–[24]. The aim of this process is to identify usability issues introduced by the visual data profiling approach in data cleaning and transformation processes.

1.6 Thesis Outline

The thesis is structured into six different sections that reflect the research process.

Chapter 1 – *Introduction* – introduces the reader to the context of the thesis and the topic to be investigated. A set of research questions are defined, and the appropriate research methodology is discussed and selected.

Chapter 2 – *Related Work* – introduces the concepts of visual data profiling and related theoretical and technological frameworks, such as data quality, and tabular data cleaning and transformation. This chapter provides the reader with the necessary background to understand the different processes that are involved in visual data profiling approaches.

Chapter 3 – *Problem Analysis* – defines the users of the visual data profiling approach, and a usability testing strategy. Next, state of the art frameworks and technologies are evaluated. Finally, user needs are identified in a process that leads to a set of requirements for the prototype that supports the visual data profiling approach.

Chapter 4 – *Implementation* – introduces the architecture of the prototype, and covers the iterative process of realizing the visual data profiling approach in a software prototype.

Chapter 5 – *Evaluation* – validates usability of the prototype to determine to which extent visual data profiling approaches are perceived useful and easy to use by data scientists. Furthermore, the evaluation uncovers usability issues in visual data profiling approaches that provide future research opportunities within the area.

Chapter 6 – *Conclusion* – summarizes the evaluation in accordance with the requirements, and proposes future research opportunities within the research field of visual data profiling

2 Related Work

Visual data profiling technologies are valuable in the context of data quality control because the process of reviewing and verifying data quality is a time and cost consuming activity [2], [11]. The basic principle behind visual data profiling approaches is to let a system perform the review of data quality and identification of data quality issues. The system collects statistics and information about the data, and then returns metadata that describes the quality of the data. Based on this information, the data scientist can make an informed decision about how the issues should be treated.

In terms of data scientists as users, a basic use case of visual data profiling would be to profile an unknown dataset before cleaning and transforming the data. We will consider this dataset to be in a CSV[2] format which is tabular data stored as plain text, separated by commas. When the dataset has been cleaned and transformed, the data scientist might want to apply machine learning techniques[3] to further examine and explore patterns in the dataset. Since the data scientist will communicate the findings to senior management that will make strategic decisions based on the information, it will be essential that the analysis is performed on high quality data. This is where visual data profiling approaches play a significant role to improve the overall data quality.

This chapter defines some of the key concepts and processes involved in visual data profiling.

2.1 Basic Data Profiling Cycle

Before we continue to the discussion of key concepts and processes, we will have a closer look at the use case for a data scientist above to better understand the mechanisms and processes behind visual data profiling. The profiling assisted

[2] https://en.wikipedia.org/wiki/Comma-separated_values

[3] https://www.sas.com/en_us/insights/analytics/what-is-a-data-scientist.html

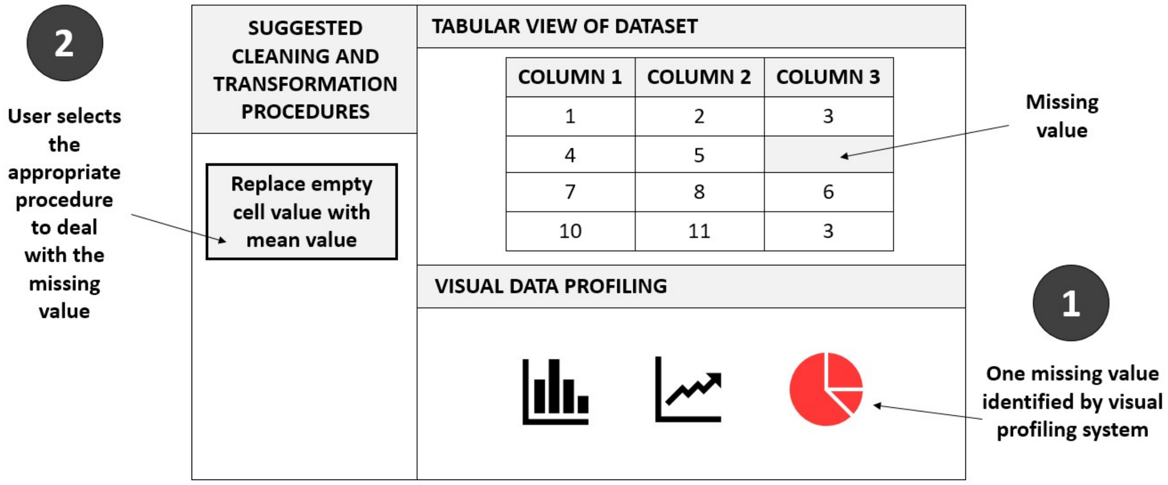
data cleaning and transformation process involves the following steps [2], [6], [25]:

1. **Discovery**: The user starts the data cleaning and transformation process by discovering the content, structure, and quality of the dataset. The visual data profiling system performs statistical assessment of data quality and returns the summarized feedback to the user.

2. **Cleaning and transformation**: Based on the statistical assessment of data quality, the user applies the appropriate procedures to clean the dataset, e.g. by correcting missing values. The dataset is further transformed to change shape into a desired format, e.g. by deleting a column.

3. **Validation**: Assisted by the data profiling system, the user validates the result of the applied cleaning and transformation procedures to ensure the output dataset has the intended content and structure.

The three-step approach above is an iterative process that can be summarized and illustrated in the following Figure 2 and Figure 3.


2
User selects the appropriate procedure to deal with the missing value
SUGGESTED CLEANING AND TRANSFORMATION PROCEDURES
Replace empty cell value with mean value
TABULAR VIEW OF DATASET
Missing value
VISUAL DATA PROFILING
1
One missing value identified by visual profiling system

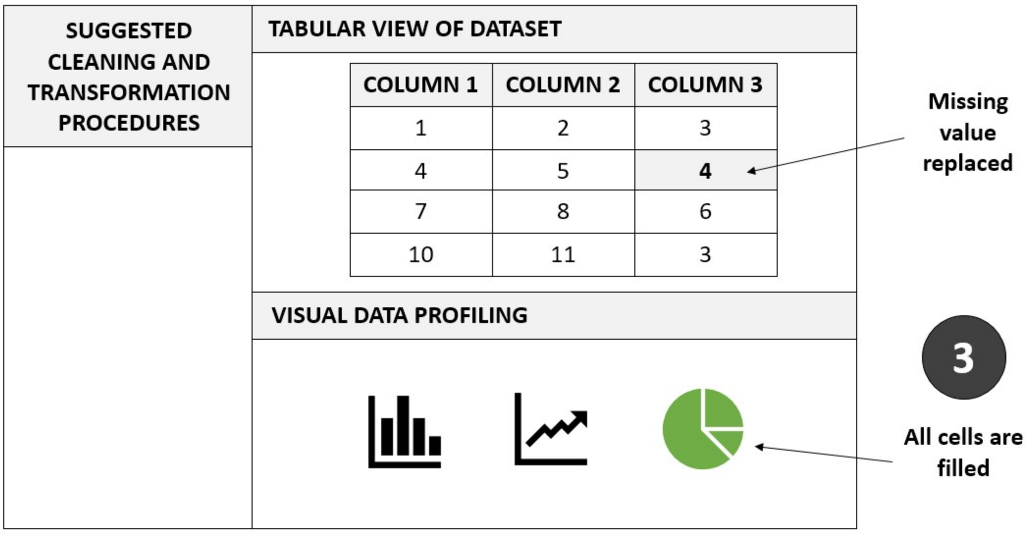
COLUMN 1	COLUMN 2	COLUMN 3
1	2	3
4	5	
7	8	6
10	11	3

Figure 2: First two steps of visual data profiling cycle

The user starts the data cleaning and transformation process by *discovering* (Figure 2, step 1) the quality of the data. A missing value is identified by the visual data profiling system. Next, the user selects an appropriate action (Figure 2, step 2) to *clean and transform* the dataset, i.e. by replacing the empty cell value with the mean value of all values in that column.

Figure 2 shows a basic user interface for visual profiling assisted data cleaning and transformation. The user interface consists of three main components:

- A *tabular view* that displays the status of content and structure of the dataset.
- A *visual data profiling view* that performs statistical assessment of the content and structure of the dataset, and identifies possible data quality issues. Visual charts are used to convey the information to the user.
- A sidebar (left) that *suggests relevant cleaning and transformation* procedures to correct data quality issues.



COLUMN 1	COLUMN 2	COLUMN 3
1	2	3
4	5	**4**
7	8	6
10	11	3

Figure 3: Last step of visual data profiling cycle

Finally, the user *validates* (Figure 3, step 3) that the data quality issue has been corrected by using the feedback from the visual charts as a confirmation.

As can be seen from Figure 2 and Figure 3, several technologies are involved in a data profiling system:

- A **logical system** that analyzes data content and structure to assess the degree of quality of the data, and identify data quality issues.
- **Statistical charts** that display the status of data quality, content and structure.
- A **data cleaning and transformation** system that has the capabilities to clean and transform the dataset.

The following sections of this chapter will discuss some of the underlying theoretical frameworks that are necessary to understand visual data profiling approaches.

2.2 Visual Data Profiling Tasks

According to Dai et al. [11], visual data profiling can be used in different scenarios such as data management, data integration, Extract-Transform-Load (ETL) processes, data migration, and data audit. Furthermore, visual data profiling tasks can be classified according to which type of feedback is expected. The scope of this thesis will include two of these categories of visual data profiling tasks, i.e. content profiling and set profiling [11]:

- **Content profiling** is a review of basic data information, including accuracy and timeliness as described in Chapter 2.4, and null values.
- **Set profiling** is a statistical analysis that typically provides data summary of distribution, frequency, value uniqueness, central tendency, row count, and maximum and minimum values. Statistical analysis and charts are discussed in Chapter 2.5.

Content profiling and set profiling have been selected for this thesis because of their suitability to be represented by statistical charts, and the relative ease of implementation in terms of a prototype. On the other hand, a profiling system that would implement pattern detection [11] would require considerable more effort and not necessarily help answer the questions set forth by this thesis.

2.3 Characteristics of Data in the Context of Data Profiling

Characteristics of data are inferred and used by the logical system behind a visual data profiling approach to perform a correct analysis and correctly identify data quality issues.

The following definitions of data characteristics are based on the work of Han et. al [5], and are used consequently in this thesis.

Data sets consist of **data objects** that represents entities. An entity can be described by its attributes. An **attribute** is a data field that represents a characteristic or variable of a data object. As an example, the 'person' data object could have typical attributes such as 'name', 'age', 'height' and 'eye_color'. In database terms, the rows equal data objects and the columns equal attributes.

Nominal, also called categorical, attributes represent a category or state. In terms of the 'person' data object, the attribute 'eye_color' could have the possible values brown, green and blue. Possible values of nominal attributes cannot be ordered in a meaningful way and are not quantitative.

Binary attributes are a subtype of nominal attributes that have only two categories or states – typically 1 or 0, alternatively true or false.

Ordinal attributes have possible values that can be ordered or ranked meaningfully, but the magnitude between values cannot be inferred. As an example, a 't-shirt' data object could have the possible size attribute values small, medium and large. The values can be ordered, but it is not known only from looking at the values how much larger small is compared to medium.

Nominal and ordinal attributes are also referred to as **string** attributes in this thesis when a distinction is not required.

Numeric attributes are quantitative and measurable, represented as integers or real values. The values of **interval-scaled** attributes can be ordered on a scale since the difference between values are equal. As an example, a 'bank customer' data object will typically have an 'account balance' attribute where the size between each successive value is equal. While values of interval-scaled attributes can be negative, zero or positive, **ratio-scaled** attributes have a defined point of zero. In terms of the 'person' data object, the attribute 'age' has a defined zero point; a person cannot be less than 0 years old.

Univariate data analysis is the analysis of a set of values in a single column of a tabular data set that is useful for identifying missing values, and values that fall outside a given domain range, i.e. outliers [1]. Because of its simplicity and usefulness in data cleaning, the proposed visual data profiling approach in this thesis assesses data quality of single, univariate attributes.

To sum up, the visual data profiling approach needs to treat missing numeric values differently from nominal values. As an example, the system will suggest replacing a missing numeric value with the mean value of that column, but this suggestion would not be applicable to a nominal string value. Hence, the system will only be efficient if it correctly infers characteristics of data.

2.4 Data Quality and Data Anomalies

We assumed in Chapter 1 that visual data profiling can improve data quality. This section discusses what data quality is, and describes some data quality issues that are relevant in terms of the visual data profiling approach.

Data Quality

Data quality can be defined as data that fulfills the requirements of intended use, and is influenced by factors such as accuracy, completeness, consistency, timeliness, believability and interpretability [5], [26].

- *Accuracy* defines the degree of noise in the data. Inaccurate data contains errors or values that deviate from what is expected.

- *Completeness* is a measure of the presence of relevant attribute values or attributes in a dataset. Incomplete data may lack certain attributes or attribute values that would be of interest in terms of the intended use.

- *Consistency* in a dataset reflects to which degree the data is constant in time, and usable in different settings. As an example, different date formats in the same dataset would be considered inconsistent.

- *Timeliness* also affects the data quality. Consider a medium sized company in which some managers fail to submit on time a report of actual working hours for each respective department. The aggregate monthly report for the whole company would then have reduced data quality.

- *Believability* describes to which degree the users trust the data, while the concept of *interpretability* defines how easy the users understand the data.

The visual data profiling approach specifically addresses the *accuracy* and *completeness* of data quality. In terms of the scope of this thesis, accuracy and completeness are sufficient metrics of data quality that are easy to measure. Believability and timeliness of data would for example be more qualitative interpretations of data quality that would be more complex to investigate, and would not necessarily provide any added value to the investigation of the research questions of this thesis.

Accuracy of data can be illustrated by an example of the attribute ‘year of birth’. This attribute would only allow values of four-digit length to be present, we will call this the domain range, and any values above 2017 will fall outside this range. As an example, the value 2020 is inaccurate since it is an illegal value that falls outside the domain range.

In terms of *completeness* of data, consider again the example of the attribute 'year of birth'. If the dataset contains several missing dates of birth, the data is incomplete due to the presence of null values.

The next section of this chapter introduces some of the data quality issues, or anomalies, that are assessed by a visual data profiling system in terms of accuracy and completeness.

Data Anomalies

Data anomalies are data quality issues that may undermine or corrupt the process and result of data analysis [2]. In terms of the visual data profiling approach, we will focus on the types of anomalies that influence accuracy and completeness of the data quality.

Hence, the approach will identify and handle two types [2] of data quality issues:

- **Missing values**, i.e. missing values of an attribute.
- **Extreme values,** i.e. outliers that fall outside a given domain range of an attribute.

Extreme values can be identified by determining how far outside a given range the values are. Univariate outlier analysis is further discussed in Chapter 2.5 since there are specific types of statistical charts that are well suited to visualize extreme values in a dataset.

Below is a description of some basic methods to fill in *missing values* in data [5], and a discussion about which of the methods that are relevant for the visual data profiling approach. Based on the procedure proposed by Han et al. [5], missing values can be treated by:

1. *Ignoring the entity by deleting row(s) in tabular data.* The method is not very effective, and should only be used when several attributes are missing. By ignoring the entity, the remaining intact attribute values are not used in the analysis.

2. *Manually filling in the values.* With this method, the data scientist needs to examine the dataset and manually fill in any missing values. Hence, this is a time-consuming activity that is only effective when a few attribute values are missing.

3. *Using a global constant to replace the missing values.* All attribute values could be replaced by a common label such as 'Missing'. The disadvantage of using this method, is that an analysis system may mistakenly consider this to be an interesting pattern. The advantage is that remaining attribute values can be included in the analysis.

4. *Using a measure of central tendency to fill in missing values.* This is an effective method that replaces missing values with a value that represents the 'middle' value of a data distribution. Different measures of central tendency are discussed in Chapter 2.5.

5. *Using the most probable value to replace missing values.* This method is effective, and uses different machine learning approaches (e.g. regression or decision-tree induction) to infer the most probable value of an attribute.

All five methods introduce some degree of bias to a dataset, since the missing values are approximated, or completely ignored (as in method 1). Method 5 is probably the most effective method, but also the most complex to implement. Method 4, using a central tendency measure to replace missing values, is an effective approach that leads to reduced bias, and is easy and intuitive to implement and demonstrate in a prototype. Hence, method 4 will be examined in this thesis as a main means to replace missing numeric values.

2.5 Statistical Charts used in Data Profiling

This section of the chapter introduces the underlying statistical logic, i.e. central tendency and data distribution, behind a basic visual data profiling system.

Descriptive statistics serve as the backbone of the visual data profiling system. Set profiling relies heavily on statistical computation and statistical chart representations. Statistical descriptions are necessary to infer data types, assess data quality and provide a general overview of the characteristics of data. Statistical background is also necessary to understand which charts to use in different situations.

The descriptive statistics and charts described in Chapter 2.1.5 are based on the work by Han et al. [5].

<u>Central Tendency</u>

Mean

$$\bar{x} = \frac{\sum_{i=1}^{N} x_i}{N} = \frac{x_1 + x_2 + \cdots + x_N}{N}$$

Equation 1: Mean

The arithmetic mean is a numeric measure of the center of a dataset, and is identical to the aggregate function avg() in SQL. A mean measure is sensitive to extreme outlier values that could distort the accuracy of the mean value. When dealing with asymmetric or skewed data, the median measure will more precisely identify the center of a dataset.

Median

$$median = L_1 + \left(\frac{\frac{N}{2} - (\sum freq)_l}{freq_{median}} \right) width$$

Equation 2: Median

In a set of ordered data values, the median corresponds to the middle value that separates each half of the dataset. Since the median is expensive to compute in large datasets, the value can be approximated by interpolation.

Mode

The third measure of central tendency is the mode value. Mode is defined as the value that occurs most frequently in a dataset. If there are several values that occur most frequently, the mode measure will be either unimodal, bimodal or multimodal.

The mode measure will work with both quantitative and qualitative data, while the mean and median measure central tendency in numeric datasets only.

Distribution of Data

The distribution is a measure of the spread of numeric data. We will define the **range** of a numeric dataset as the difference between the maximum **max( )** and minimum **min( )** values. Dividing a dataset into nearly identically sized sections, **quantiles** are the data points that divide the sections. If the distribution of a dataset is divided in four identical parts, the quantiles are also referred to as **quartiles**.

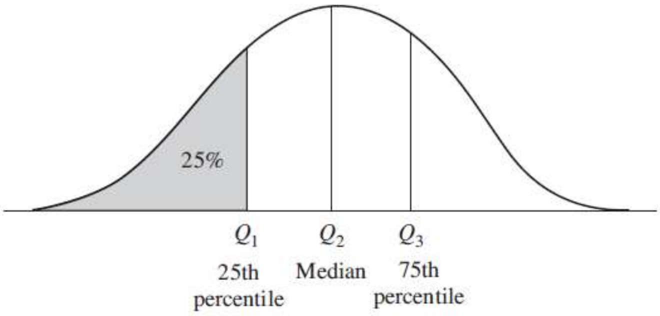


Figure 4: Data distribution chart

Considering Figure 4, Q1, Q2 and Q3 correspond to the first, second and third quartiles. Quantiles can also be expressed as percentiles which divide the dataset into 100 equally sized sections [7]. The spread between quartile Q3 and Q1 will be defined as the **interquartile range (IQR)**:

$$\text{IQR} = \text{Q3} - \text{Q1}$$

Equation 3: Interquartile range (IQR)

Outliers will be defined as suspected extreme values that are too far from the median to be considered in a dataset. A common way to identify suspected outliers is to measure which values fall in the range above or below *1.5 x IQR.*

Suspected outliers = 1.5 x IQR

Equation 4: Suspected outliers

Skewed data distributions often require more than only one measure (e.g. IQR) to identify spread. The **five-number summary** provides a more precise description of distribution, and consists of the values [Minimum, Q1, Median, Q3, Maximum].

Variance (σ^2) and **standard deviation** (σ) are two additional measures of the spread of data. A low standard deviation indicates that the data observations are distributed close to the mean, while a high standard deviation tells that the observations are distributed over a broader range of values.

In terms of variance, consider a numeric attribute X, and N observations, $x_{1,}$ x_2,…,x_N. The mean value of the observations is $\bar{x}$. Consequently, the **variance** will be defined as:

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2 = \left(\frac{1}{N}\sum_{i=1}^{N}x_i^2\right) - \bar{x}^2$$

Equation 5: Variance

Standard deviation is the square root of variance:

$$\sigma = \sqrt{\sigma^2}$$

Equation 6: Standard deviation

Boxplot Chart

When the five-number summary is computed, a boxplot visualization can be rendered.

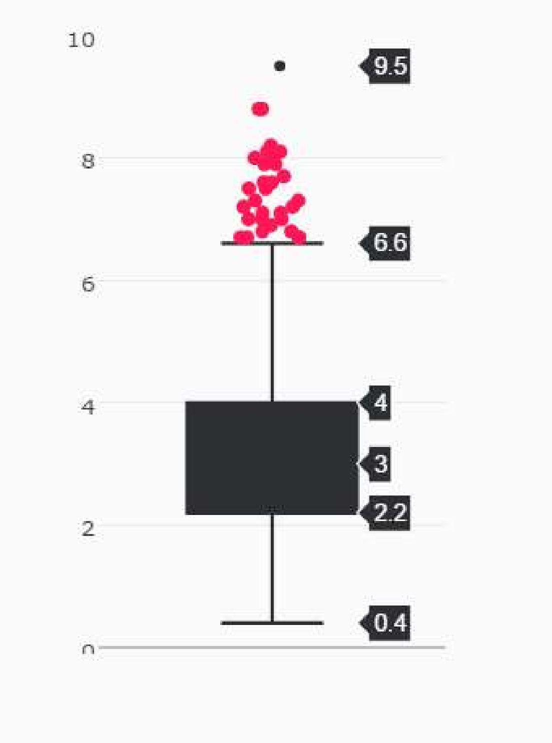


Figure 5: Boxplot chart

The boxplot in Figure 5 can be described in the following way:

- The boxplot visualizes the data distribution of some attribute x. The y scale measures the size of x.

- The black solid box represents the interquartile range between Q1 (value 2.2) and Q3 (value 4). The median is defined approximately in the middle of this box (value 3).

- The two lines that stretch from each side of the box are called whiskers, and ends at respectively the minimum (value 0.4) and maximum (value 6.6) value of the data distribution.

- The **suspected outliers** that belong in the range above 1.5 x IQR are visualized as red dots in the boxplot chart in Figure 24.

- The outlier at value 9.5 (black dot) lies outside the range of defined suspected outliers.

Histogram Chart

A histogram, also called frequency histogram or bar chart, summarizes the distribution of an attribute.

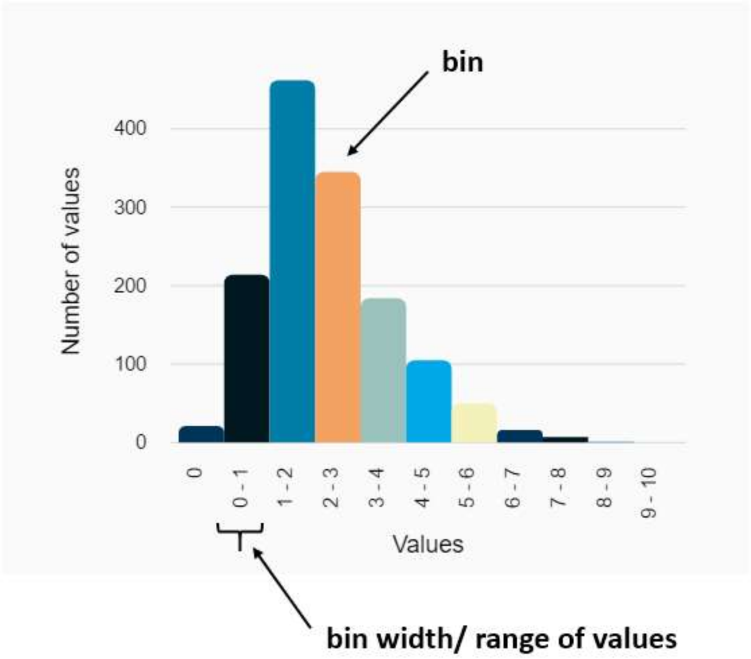


Figure 6: Numeric histogram chart

Nominal attributes will require one bar for each unique value of the attribute, and the height of the bar indicates the count of attribute frequency. *Numeric* histograms (e.g. Figure 6) partition the total range of values into equally sized *bins.* The term *width* corresponds to the range of values of a bin.

3 Problem Analysis

The problem analysis chapter of this thesis will discuss and analyze several aspects that are interrelated, and eventually leads to a set of requirements for the implementation of the prototype that realizes the visual data profiling approach.

Since we want to investigate to which degree visual data profiling approaches are perceived useful and easy to use, a framework is needed to explore and measure usefulness and ease of use. Hence, the concepts of usability and usability testing are introduced together with a usability testing strategy that provides the necessary research methodology to explore the research questions of this thesis. Furthermore, the typical user is described in more detail to build a user profile that is required by the testing strategy to provide valid results.

Having identified and described the typical user as a data scientist, a review of state of the art is conducted within existing research and solutions to visual data profiling. The review considers existing solutions and approaches that would be useful to data scientists, and that will influence a visual data profiling approach for this specific user group.

Finally, the user needs are identified in a process that leads to a set of requirements for how to realize a visual data profiling approach for data scientists by means of a software prototype.

3.1 Usability Testing

Usability testing is applied to evaluate how useful and easy to use the visual data profiling approach is, and reveals potential usability issues that are introduced in the prototype that realizes the approach.

To answer the research questions in terms of usability of the visual data profiling approach, we will need to define **usability** and **usability testing**. The widely used ISO (9241-11) standard [8], [27] defines usability as:

> *"The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use."*

This definition emphasizes three important aspects of usability. First, the approach should be targeted at specific users. Second, the users share a common goal to move towards. Third, the approach should work in the users' environment.

Furthermore, the definition provides essential **measures of usability** [8]:

- *Effectiveness*: The extent to which a user reaches a goal accurately. As an example, a user that wants to transform a dataset will consider an application effective if the resulting dataset meets the expectations of the user.

- *Efficiency*: How fast a user reaches a goal. As an example, a user may consider the work process of an application to be slow and time-consuming even if the dataset is accurately transformed. The application is then considered to be effective, but not efficient.

- *Satisfaction*: The general user experience of an application, and a measure of how users individually perceive satisfaction. Different users might not express the same level of satisfaction because they perceive the situation differently. E.g. if a user likes the design and colors of an application, the user will probably be more satisfied with the overall user experience.

Usability testing will here be defined as involving users that interact with a software application to uncover usability issues. Furthermore, usability testing is used to measure perceived usefulness and ease of use of the application. There are basically two main types of usability testing, namely formative testing and summative testing [8], [20]. *Formative testing* is iteratively performed throughout the development phase to identify and correct problems. *Summative testing* is performed by using metrics to describe the usability of an application, e.g. when comparing two designs of an application. Both formative and summative testing are used in this thesis to measure usability and user experience.

3.1.1 Identifying the Users of the Visual Data Profiling Approach

As discussed in chapter 1, this thesis adopts a user-centered design approach [8] in which users are involved in the development processes. The purpose of involving users is to make sure that we propose an application that *targets the specific needs of specific users*. As part of this process, it is also necessary to define and *understand the goals* that the users want to achieve by using a certain approach.

Two complementary methods of understanding users and their goals are applied in this thesis:

- *Personas*: descriptions of a typical user of the application.
- *Scenarios*: descriptions of the process or steps that a persona will execute in the application to reach a specific goal.

Users are *goal-oriented* and bring with them prior experience and expectations [8]. When presented with a new application, goal-oriented users compare the effort of learning with the potential positive effects of using the application. In general, adult users want to act immediately and with minimal effort towards their goal. They will also develop and apply *schemas*, or mental models, when learning to use a new application. A mental model tells a user how to use a certain application. Hence, based on previous experience and expectations, two users might apply completely different schemas. The concept of schemas is linked to users' motivation and commitment to learning. A user that is enthusiastic about using the interface and functionality of Microsoft Excel[4] for data cleaning, will probably feel less motivated and dedicated to learning a new application that is very different from Excel in terms of functionality and user interface.

Due to the impact of previous experience and schemas on user experience, the design of the application should adopt ideas from familiar user interfaces of industry-standard data cleaning and transformation approaches.

[4] https://products.office.com/en/excel

Considering the scope of this thesis, a typical user of the visual data profiling approach can be defined in the following persona in Figure 7.

PERSONA: THE DATA SCIENTIST

"I want to spend less time on data cleaning"

Age: 35
Work: Data Scientist
Location: Norway

Bio

Hanne is working for a small company in Norway that collects and analyzes real-time data streams from physical outdoor sensors measuring human traffic flow. The data-driven insights and analytics about people's traffic patterns are sold to businesses. Hanne is the only data scientist in the company that is involved in the process of cleaning, transforming, and analyzing large datasets.

Goals

- ✓ Clean and transform tabular data to prepare for analysis.
- ✓ Spend less time on cleaning and transforming data.
- ✓ Improve data quality before the data enters the analysis stage.

Frustrations

- ✓ I spend up to 80% of my time on cleaning and transforming datasets.
- ✓ Existing solutions are difficult to use.
- ✓ Poor-quality data undermines the analysis of data, and biases and corrupts data-driven business insights.

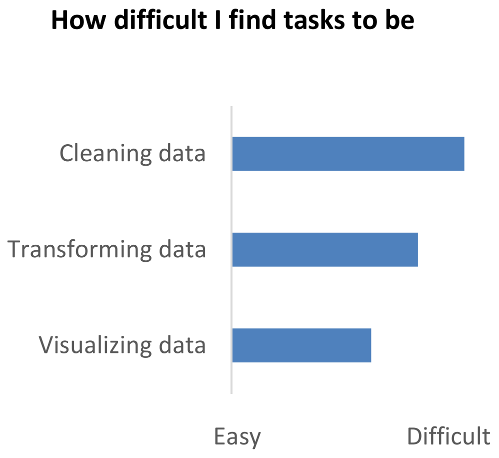


Figure 7: User persona

The concepts of usability and usability testing have now been defined, and a user profile of a data scientist has been proposed. The next step of the problem analysis process defines a plan for usability testing, which includes a detailed procedure for the methods that are involved in the evaluation. The test plan is based on the approach suggested by Barnum [8], in which we define test goals, how and where the approach is tested, and which user groups are included.

3.1.2 Defining the Usability Testing Strategy

We use two different methods of formative testing to validate the usability requirements and answer the research questions:

- *Comparative usability test (survey based)* – a usability analysis approach that is used to measure perceived usefulness and ease of use of two different approaches or applications [7], [20]. The current version of Grafterizer is compared with the visual data profiling approach.

- *Streamlined cognitive walkthrough* – a usability inspection method that is applied to evaluate how easy the visual data profiling approach is to use without prior instruction or training [17], [22]. This method identifies potential usability issues.

These two methods are appropriately selected to align with the user-centered design process. By combining a user-centered design process and agile approaches, it is essential that the adequate methods are selected [29]. Since the applied development process is highly agile and iterative, the comparative usability test provides useful feedback between prototype iterations. The user feedback gathered at each checkpoint enters the next implementation cycle to evolve into a framework that is iteratively more useful and easier to use.

The comparative usability test and streamlined cognitive walkthrough methods are cost-effective, and provide a reasonable balance between efforts spent and the potential gains of involving users on a regular basis.

The Survey-based Comparative Usability Test

The comparative usability test is a qualitative and quantitative usability analysis tool that will measure user experience of the prototype in comparison with the current version of Grafterizer. The purpose of conducting the survey can be formulated as a set of test goals:

1. Learn how users perceive the prototype in terms of **usefulness** [7]

2. Learn how users perceive the prototype in terms of **ease of use** [7]

The survey contributes to answering two of our research questions. Test goal 1 is related to the following research question:

- **How *useful* are visual data profiling approaches** for users of tabular data cleaning and transformation tools?

Next, the following research question is answered by test goal 2:

- **How *easy to use* are visual data profiling approaches** for users of tabular data cleaning and transformation tools?

A representative group of users is selected to participate in the survey. Voluntary participants from project meetings in current research initiatives are invited to participate in the comparative usability test, respond to the survey questionnaire, and provide qualitative feedback on the visual data profiling approach:

- **EW-Shopp**[5] (project meeting February 2017)

- **proDataMarket**[6] (project meeting March 2017)

- **euBusinessGraph**[7] (project meeting May 2017)

[5] http://ew-shopp.eu/

[6] https://prodatamarket.eu/

[7] http://eubusinessgraph.eu/

The participants represent typical and actual users in terms of background and expectations to user experience and functionality. In all three research initiatives, SINTEF[8] is committed to deliver capabilities in DataGraft and Grafterizer that are necessary to complete the research initiatives. In terms of this thesis, each of the three project meetings corresponds to a survey test session.

Facilitators of the online survey are Nikolay Nikolov[9] (SINTEF Digital, product expert and team lead DataGraft) and Bjørn Marius von Zernichow (the author of this thesis).

The comparative usability test consists of two parts. In each session, the following sequence of steps is conducted:

1. Survey participants observe a live demonstration of:

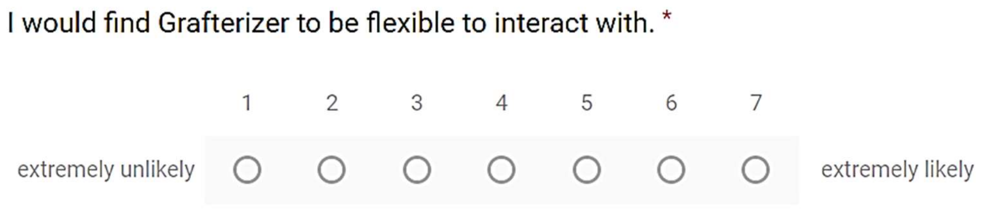
    - DataGraft.io
    - The current version of Grafterizer
    - The visual data profiling prototype

2. When the demonstrations finish, survey participants receive a link to an online survey that is intended to measure perceived usefulness and ease of use of each of the three demonstrated systems.

The survey measures user experience on the dimensions of usefulness and ease of use by asking participants 6 questions related to each of the dimensions. The questionnaire uses a Likert scale [7] that ranges from 1 to 7, as can be seen in Figure 8 below.

[8] http://sintef.no

[9] http://www.sintef.no/alle-ansatte/ansatt/?empId=5177



I would find Grafterizer to be flexible to interact with. *

	1	2	3	4	5	6	7	
extremely unlikely	○	○	○	○	○	○	○	extremely likely

Figure 8: Survey example question

The evaluation of DataGraft.io is omitted from this thesis since DataGraft is outside of the scope. Still, the evaluation of DataGraft is used by the DataGraft research team to collect information about usefulness and ease of use.

Finally, the survey participants provide qualitative feedback after each session and demonstration of the visual data profiling approach to suggest changes that should be implemented until next test session. The iterative develop – test – develop cycle can be illustrated in Figure 9 below.

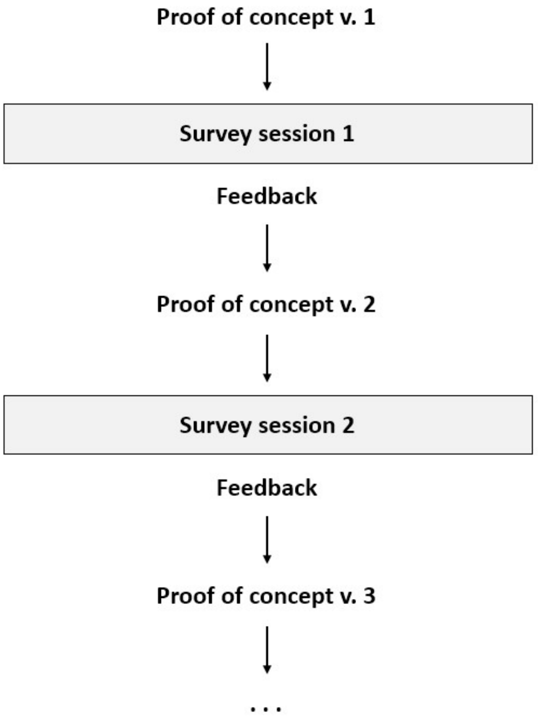


Figure 9: Survey feedback cycle

The Streamlined Cognitive Walkthrough

This method is a qualitative usability inspection method that involves expert reviewers [22]–[24].

The purpose of using the method to evaluate usability of the visual data profiling approach, can be formulated as a test goal:

- Understand the learnability of the prototype for new users, i.e. the ease of:
    - **Learning** how to use the system's **functionality**
    - Developing **skills** needed to perform basic and necessary **tasks**

The following research question is related to the test goal:

- **Will visual data profiling approaches introduce usability issues** in tabular data cleaning and transformation processes, and if so; which types of usability issues occur and how can they be corrected?

The advantage of the streamlined cognitive walkthrough method is its capabilities to identify possible usability issues. The main goal is to evaluate whether cues and feedback in the user interface reflect the way typical users cognitively process tasks and anticipate next steps of the system.

In total **four expert reviewers** are selected to participate in the sessions. Users are divided in two subgroups and two corresponding sessions:

a. **Session 1:** Two Human-Computer Interaction (HCI) experts from SINTEF Digital[10]

[10] http://www.sintef.no/en/information-and-communication-technology-ict/departments/networked-systems-and-services/human-computer-interaction-hci/

b. **Session 2:** Two linked data domain experts from the Logic and Intelligent Data (LogID) group at University of Oslo[11]

Each of the sessions will walk through one or more user scenarios to identify potential usability issues. The scenarios will be discussed in more detail in Chapter 5.

The sessions are facilitated by Bjørn Marius von Zernichow, the author of this thesis, and can be summarized in the following sequence of steps:

- The **facilitator** provides all information, context and material needed to conduct the walkthrough.
- A representation of the user interface is provided to the **expert reviewers.**
- **Facilitator** walks through scenarios and action sequences, and **expert reviewers** assume the role as usability experts answering two pre-defined questions for each step of the scenario.
- **Facilitator** records feedback from **expert reviewers.**
- After the review: **Facilitator** analyzes feedback and suggests changes in user experience and functionality

The duration of each session is 90 – 120 minutes.

We have now defined the users of the visual data profiling approach, and decided on a usability testing strategy. Chapter 5 follows up and implements this strategy to evaluate usability of the visual data profiling approach.

[11] http://www.mn.uio.no/ifi/english/research/groups/logid/

3.2 Evaluation of State of the Art Approaches

This section of the chapter evaluates existing research and solutions within the area of visual data profiling and related technologies that are necessary to build a visual data profiling approach. The development of the approach draws upon current research, and is inspired by existing solutions, within the areas of data profiling technologies, visual analysis systems, and tabular data preparation approaches.

Data Profiling Approaches

Profiler[2] is an example of a system for data quality analysis that includes data mining and anomaly detection techniques in addition to visualizations of relevant data summaries that can be used to evaluate data quality issues and possible causes. Profiler integrates statistical and visual analysis to reduce the time spent on data cleaning activities. The Profiler architecture and framework were developed by the former Stanford Visualization Group, now UW Interactive Data Lab[12]. This team also developed **Polaris** [30] that evolved into the commercialized business and analytics software **Tableau**[13], and **Data Wrangler** [31] that together with Profiler merged into the commercialized data preparation solution **Trifacta**[14].

The above-mentioned profiling solutions all originated in research environments, are well documented in research literature, and represents effective and user-friendly approaches to data profiling.

Moreover, **Talend**[15] uses similar visual profiling techniques as Trifacta to automatically explore data characteristics and data quality issues. Talend focuses on ease of use and an intuitive user-interface.

[12] http://idl.cs.washington.edu/about

[13] https://www.tableau.com

[14] https://www.trifacta.com/

[15] https://www.talend.com/products/data-preparation/

In terms of usability testing of a visual data profiling approach, it would be challenging to use Trifacta or Talend as the system under test. First, it is difficult to isolate the data profiling capabilities from the data cleaning and transformation functionality. Hence, it would be problematic to know what is really evaluated. Second, the solutions are not open-source, and cannot be further developed to extend the current version of Grafterizer.

In terms of logic behind an effective profiling system, Heer et al. [6] propose a framework for predictive interaction and data profiling in data transformation routines. Predictive interaction and profiling algorithms in interactive systems reduces the technical specification burden of the user, and guides the user to decide on applying the most relevant data transformation.

Data Profiling Visual Analysis Approaches

One of the most important components of the visual data profiling approach, is the chart visualizations that represent statistical properties of the data.

Fundamental visualization principles and techniques for quantitative data analysis are described by Mackinlay [32], Bertin [33], Cleveland [34], Ward et. al [35], and Few [36]. The work constitutes the basis of research based design guidelines for information visualization, and this thesis draws upon these fundamental principles when the visual data profiling approach is developed

Generating visualizations from large data sets requires an understanding of users' needs and preferences along with knowledge of visual encoding rules and perception guidelines [37]. There are two general approaches to building a visual analysis system. First, considering visual encoding only will generate all possible valid visualizations without acknowledging the specific needs and preferences of users [38]–[40]. Second, introducing a **visualization recommender system** in a visualization pipeline [38]–[40] will potentially reduce the information overload of presenting all available visualizations. Tracking and storing information provided by the recommender system enables **adaptation** of the visualization system due to an evolving knowledge about which visualizations are valid and preferred by users [38].

Hence, a visualization recommender system needs to:

1. Comply with *visual encoding rules* to ensure valid visualization configurations.

2. Present only visualizations that the user *needs and prefers.*

Considering the two approaches to building a visualization system, the first approach, easiest and most cost-effective to implement, is to build a visualization system that relies on visual encoding rules only. The second approach is more complex to implement as it requires a recommendation engine that can propose only visualizations that the user needs and prefers.

Because of time and cost concerns, the first approach should be used to implement visual analysis system capabilities in a visual data profiling approach. Nonetheless, this approach is sufficient in the context of testing usability of visual data profiling.

Finally, **Voyager** [40] is an exploratory data analysis tool that is open-source, originated in research, and provides state of the art within open source data exploration. Voyager specifies visualizations through Vega-Lite [41], a high-level declarative *JSON*[16] specification language based on Wilkinson's Grammar of Graphics [42], ggplot2 [43] and Tableau VizQL [30], [44]. Vega [45] is the underlying formal model for rendering Vega-Lite specifications.

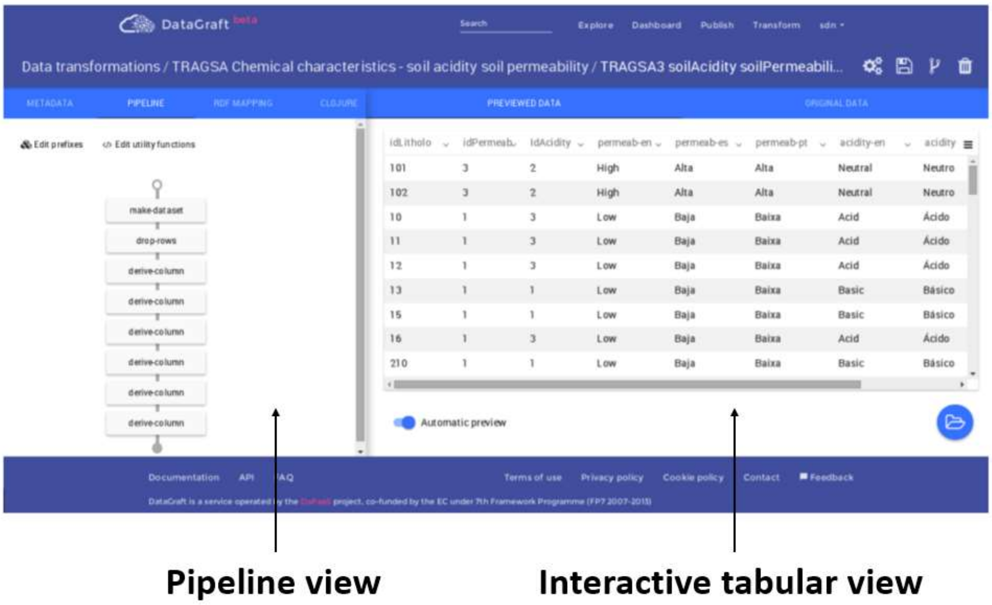
The advantage of using high-level declarative language to specify visualizations, is that the user's burden of specification is reduced, and less time is spent on defining charts. The disadvantage is that a high-level declarative language often restricts some of the functionality and flexibility you would find in the underlying formal model.

The visual data profiling approach is inspired by Voyager, Vega, and Tableau, and implements a high-level declarative language to specify visualizations.

[16] http://www.json.org/

Tabular Data Preparation Approaches

Grafterizer is a web-based framework for data cleaning and transformation on the DataGraft platform [10]. Grafterizer features a tabular view that displays the result of currently applied cleaning and transformation procedures, and a powerful pipeline view where the user can specify and edit data cleaning and transformation steps. The Grafterizer user interface is displayed in Figure 10. The resulting dataset will be in tabular format (i.e. as a CSV file), and the user can choose to transform the tabular data into RDF[17], a graph-based data model used to describe things and their relationships, to produce linked data. Linked data[18] is the concept of using the Web to interconnect related data.



idLitholo	idPermeab.	IdAcidity	permeab-en	permeab-es	permeab-pt	acidity-en	acidity
101	3	2	High	Alta	Alta	Neutral	Neutro
102	3	2	High	Alta	Alta	Neutral	Neutro
10	1	3	Low	Baja	Baixa	Acid	Ácido
11	1	3	Low	Baja	Baixa	Acid	Ácido
12	1	3	Low	Baja	Baixa	Acid	Ácido
13	1	1	Low	Baja	Baixa	Basic	Básico
15	1	1	Low	Baja	Baixa	Basic	Básico
16	1	3	Low	Baja	Baixa	Acid	Ácido
210	1	1	Low	Baja	Baixa	Basic	Básico

DataGraft is a service operated by the DaPaaS project, co-funded by the EC under 7th Framework Programme (FP7 2007-2013)

Figure 10: Grafterizer user interface

[17] https://www.w3.org/RDF/
[18] http://linkeddata.org/

Grafterizer is a powerful, open source framework for tabular data cleaning and transformation, but lacks visual data profiling capabilities. Grafterizer is a suitable fit for testing usability of visual data profiling, since the current version would be a benchmark for the visual data profiling approach.

Furthermore, Forrester [46] has evaluated data preparation tool providers and identified the seven most significant ones in 'The Forrester Wave Q1 2017 report', of which three providers are of specific relevance to explore this thesis' research questions:

1. **Trifacta**[19] is a self-service data preparation tool that leverages machine learning algorithms and predictive transformations to enable users in making more informed and effective data cleaning decisions. Trifacta optionally connects to Tableau for data analysis. Recently (March 9, 2017) Trifacta partnered with Google [47] to provide a cloud-based data preparation solution, **Cloud Dataprep**[20], as an addition to the existing Google Cloud Platform.

2. **Paxata**[21] is a self-service data preparation system for business analysts that focuses on decreasing time-to-insight. Paxata utilizes machine learning and semantic analytics in the data preparation process, and connects to Spark[22] for large-scale data preparation.

3. **Unifi**[23] is a self-service data preparation and data integration platform that utilizes advanced machine learning algorithms and artificial intelligence to recommend semi-automated ETL processes, and provide insight about datasets.

The visual data profiling and tabular data preparation capabilities of Trifacta, Paxata and Unifi represent state of the art within data cleaning and transformation, but the solutions are commercial and not publicly available. In terms of

[19] https://www.trifacta.com
[20] https://cloud.google.com/dataprep
[21] https://www.paxata.com
[22] http://spark.apache.org
[23] http://unifisoftware.com

all three data preparation tools, the underlying source code is not exposed. Still, the tools represent some of the most significant providers, and influence the design and implementation of the visual data profiling approach in terms of profiling capabilities and usability of such.

Microsoft Excel[24] is a widely-used tool to prepare data for analysis and gaining insight into data. A central feature of Excel is the direct manipulation interface [48] where users can interact with the table to manipulate the dataset (e.g. selecting columns and/ or rows, right-clicking for options). The advantage of a direct manipulation interface, is that many users are already familiar with this interface, and less time is required to learn to use the tool.

3.3 Requirements

The requirements for the system describe what the system should do and its purpose, reflecting the needs of users of the system [16].

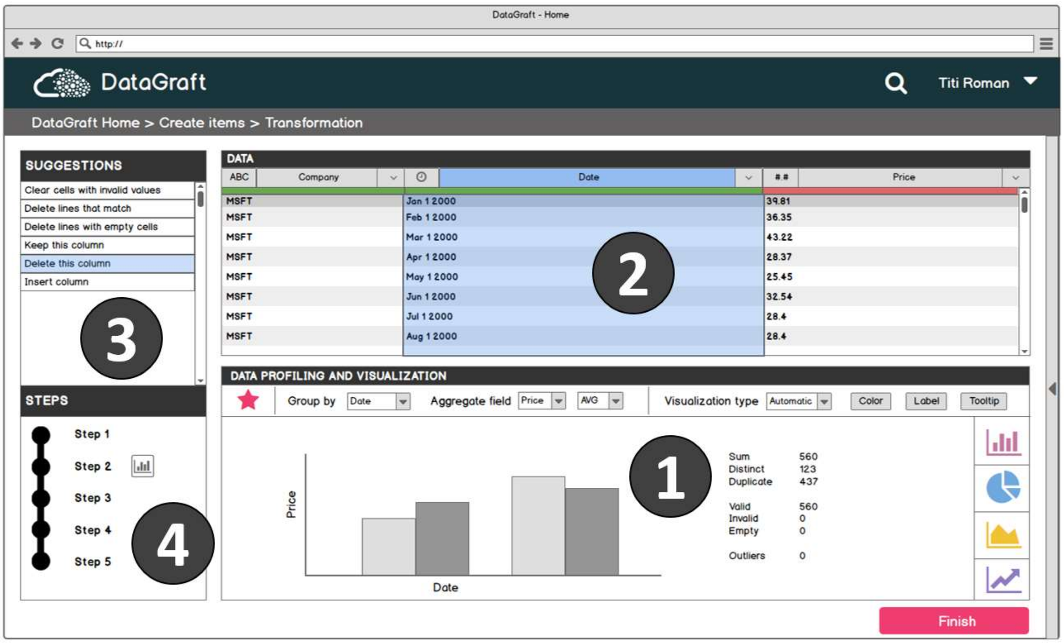
The requirements emerge from needs of the current users of Grafterizer, and as a research opportunity to propose an approach that will contribute to improving data quality in this context. Grafterizer provides state of the art functionality within data cleaning and transformation capabilities, but there is still a need for improving user experience by providing approaches that assist the users in achieving their goals of cleaning and transforming data. User feedback shows that Grafterizer has a steep learning curve, and is complex to use. Hence, novel approaches should be considered to provide useful functionality, and a user interface that is easy to learn and easy to use. The overall goal should be to provide an approach that will contribute to improving data quality by extending the current capabilities of Grafterizer.

Based on this, the visual data profiling approach should provide the necessary statistical profiling capabilities that are needed to assist the user in identifying data quality issues and ease the process of improving quality. The visual data profiling capabilities should be integrated with a table view interface that lets the

[24] https://products.office.com/en/excel

user manipulate columns and rows directly. Furthermore, the user interface should provide data cleaning and transformation functionality that is relevant to the user and appropriately addresses the goals that the user tries to achieve. The applied data cleaning and transformation sequences should finally be reflected in a steps pipeline.

To facilitate the requirements process, a wireframe has been created to describe the user interface and functionality, and the needs of users, that leads to a set of requirements. A wireframe is a sketch, or illustration, that outlines the basic graphical user interface components, and functionality to resemble the final version of the application [18]. The wireframe is the first step to realizing the visual data profiling approach. Wireframes can be directly used in the implementation of the user interface of a prototype that supports the visual data profiling approach. Balsamiq[25], an online wireframing tool that is used by SINTEF[26] to design user interfaces in prototypes, has been selected to create the wireframe.



ABC Company	Date	## Price
MSFT	Jan 1 2000	39.81
MSFT	Feb 1 2000	36.35
MSFT	Mar 1 2000	43.22
MSFT	Apr 1 2000	28.37
MSFT	May 1 2000	25.45
MSFT	Jun 1 2000	32.54
MSFT	Jul 1 2000	28.4
MSFT	Aug 1 2000	28.4

Sum	560
Distinct	123
Duplicate	437
Valid	560
Invalid	0
Empty	0
Outliers	0

Figure 11: Visual data profiling approach wireframe

[25] https://balsamiq.com
[26] http://sintef.no

The user interface of the visual data profiling approach illustrated in the wireframe in Figure 11, consists of the following main components and capabilities:

- A **visual data profiling** component (Figure 11, component 1).
- A tabular **table view** that provides data cleaning and transformation functionality (Figure 11, component 2).
- A sidebar that **suggests relevant data cleaning and transformation** actions to the user (Figure 11, component 3).
- A **steps pipeline** that reflects applied data cleaning and transformation steps (Figure 11, component 4).

The functionality of the visual data profiling approach in Figure 11 can be described in a sequence of steps that illustrates the visual data profiling cycle. As an example, consider a dataset with stock prices at various dates for five different companies (i.e. as shown in Figure 11):

1. The column 'Date' (middle column, highlighted) is selected in the table view.
2. Suggested transformations display in the 'Suggestions' sidebar.
3. The 'Data profiling and visualization' view will provide a statistical assessment and profile of the data for the selected column 'Date'.
4. Optional: User may select any sections of the visualizations that will further suggest transformations for that specific section only, e.g. if the user selects a specific range of dates, the suggested transformations will be valid for this range only.
5. Selecting a suggested transformation will add a transformation step to the pipeline and the table view will update to reflect the changes.
6. Repeat steps 1 – 5 to continue profiling, cleaning and transforming the dataset.

The functionality described in Figure 11 can be summarized as a set of requirements in Table 2 below.

Table 2: Requirements

Requirements of the visual data profiling approach		**Type**
R1	Provide visual data profiling capabilities	**F**
R2	Provide data cleaning and transformation functionality	**F**
R3	Provide data cleaning and transformation suggestions	**F**
R4	Provide a pipeline that reflects applied data cleaning and transformation steps	**F**
R5	Provide a solution that is useful to the user	**NF**
R6	Provide a solution that is easy to use	**NF**

Functional requirement	**F**
Non-functional requirement	**NF**

We distinguish between functional requirements and non-functional requirements [16]. *Functional requirements* are the statements that describe the services and functionality that the system should provide. Requirements 1 to 4 in Table 2 are examples of such specifications. *Non-functional requirements*, on the other hand, are specifications that not directly relate to the specific services or functionality of the system. These statements might affect and involve the overall system architecture, such as specifications related to performance, security, reliability, availability, and usability. Requirements 5 and 6 in Table 2 fall into the category of non-functional usability requirements.

4 Architecture and Implementation

In the previous chapter, a set of requirements for the visual data profiling approach was defined.

This chapter focuses on the implementation of the approach based on the requirements, and starts with defining an architecture that will be used to realize the visual data profiling approach.

Next, the functionality needed to demonstrate and validate the visual profiling approach, is defined as a set of eight data cleaning and transformation functions.

Finally, the chapter describes the process of iteratively implementing the capabilities and functions of the visual data profiling approach in a software prototype.

4.1 Architecture

One of the purposes of developing a prototype, is to reuse as much as possible of the architecture and code in the final development version that will extend the current capabilities of Grafterizer. Considering the importance of reuse, the high-level architecture will reflect the overall architecture as it would be defined in a production ready version.

The high-level system architecture will be based on a microservice architecture, and use the design principles of Separation of Concerns (SoC) [49]. SoC is traditionally achieved in layered architectures, e.g. in a 3-Tier architecture, by defining interfaces and encapsulating information. A 3-Tier architecture (i.e. Figure 12, adopted from Familiar [49]) would separate concerns into a presentation layer, an application tier, and a data layer.

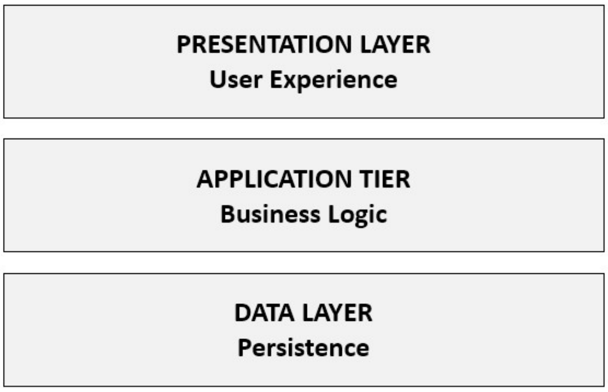


Figure 12: 3-Tier architecture

A microservice architecture would take the SoC one step further by dividing the application tier and data layer into separate, domain-driven services that would operate autonomously from other services. A network-protocol would provide secure end point access to the services. While the SoC in a layered architecture is horizontal, the SoC in a microservice architecture would be both horizontal and vertical.

There are two important characteristics that define what a microservice is [49]. First, a microservice is *autonomous* – it can exist and operate independently of the surroundings. Second, it is *isolated* and can be used in different space and time (e.g. one service can exist in Europe, another in US, and both services can be used independently of each other at separate times).

Applying a microservice architecture in the context of this thesis would be beneficial in several ways:

- *Evolutionary:* considering the iterative approach of developing the prototype, it would make sense to identify one business domain at a time (e.g. first domain: data cleaning and transformations, second domain: visual data profiling) that would evolve into a set of microservices. In this way, capabilities can be incrementally added to the prototype.

- *REST APIs:* using open industry standards (i.e. REST) for exposing functionality will increase interoperability. As an example, the front end could be developed in JavaScript[27] while the back end microservices could be developed in Python[28] programming language.

Based on the considerations above, the following architecture in Figure 13 represents a microservice approach that implements the design principles of SoC both horizontally and vertically.

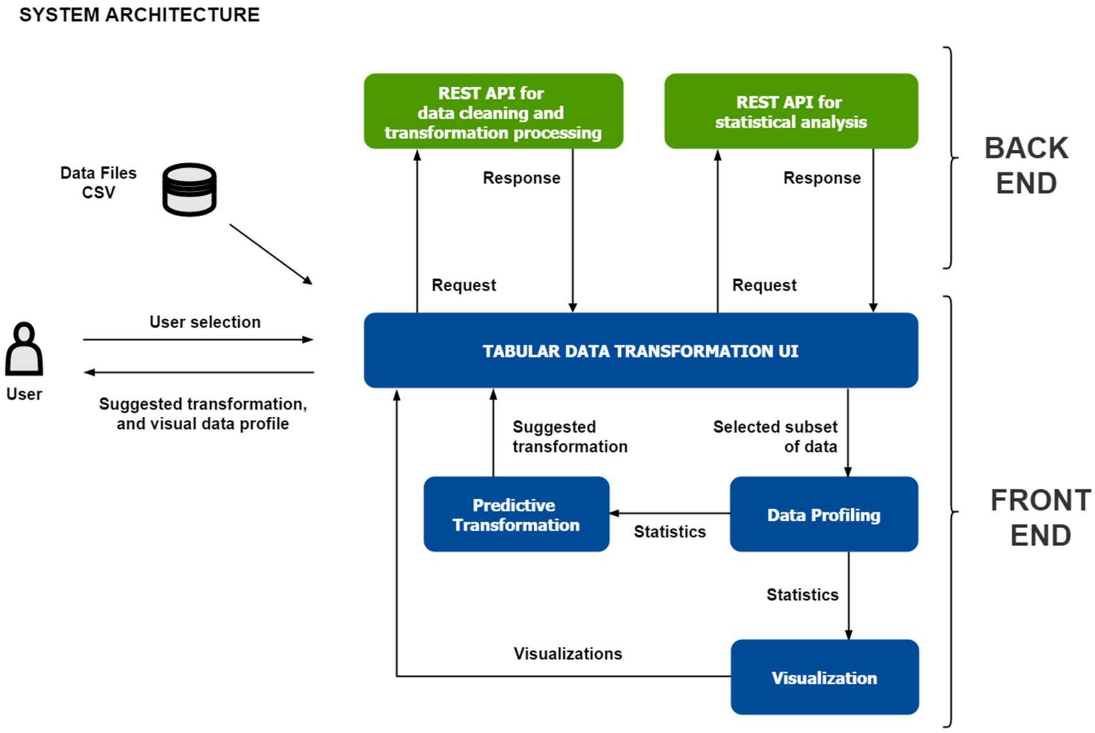


Figure 13: Visual Data Profiling microservice architecture

The visual data profiling approach implements a front-end version of the architecture in Figure 13, i.e. the back-end services will be implemented as part of the

[27] https://www.javascript.com
[28] https://www.python.org

front-end framework. Hence, all statistical analysis and data cleaning/ data transformation processing will be performed front end.

When the prototype eventually moves towards a production ready application, the microservices back end architecture could be implemented. All processing-intensive business logic and persistence related concerns that are client side in the proof of concept will then be moved to the respective back end services. Still, this implementation is outside of the scope of this thesis.

Figure 14 illustrates the front-end architecture that is used in the implementation of the prototype. Angular 2[29] is selected as development framework for building the visual data profiling prototype, and the architecture is based on Angular 2 best practices for architectural patterns [50].

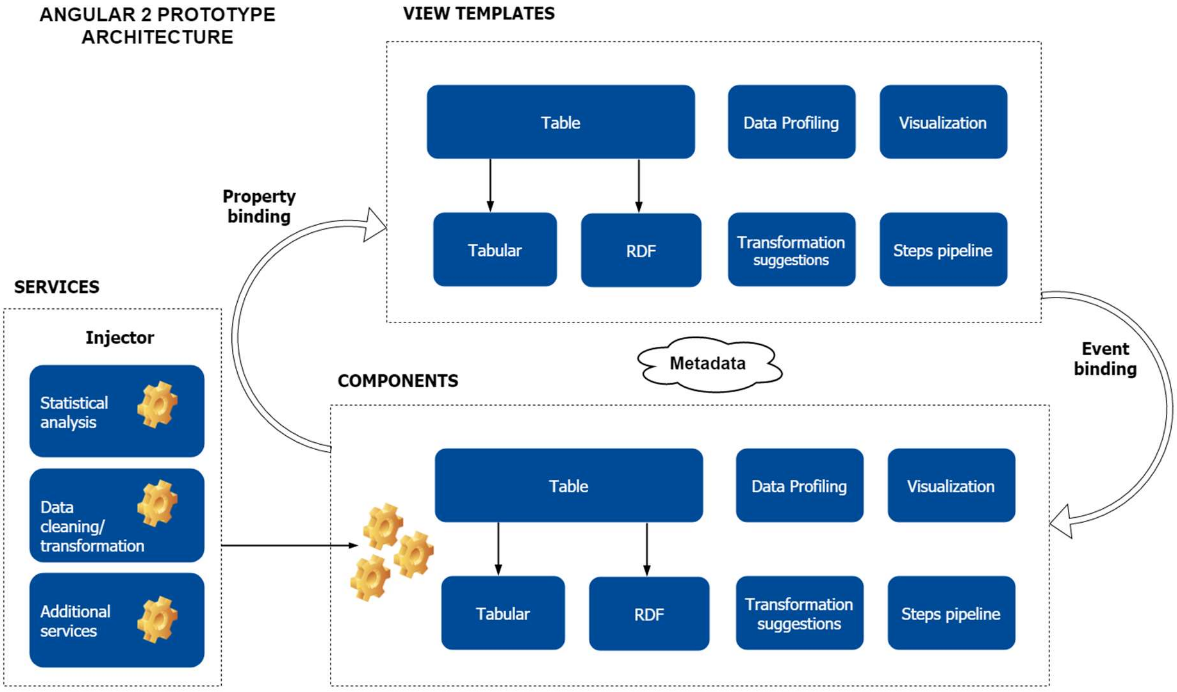


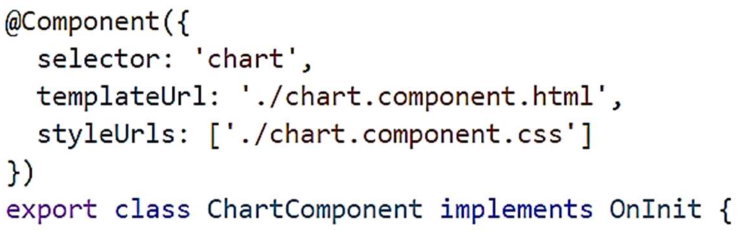
Figure 14: Angular 2 prototype architecture

[29] https://angular.io

The building blocks of the prototype architecture in Figure 14 can be defined as follows [50]:

Components: The Angular 2 components define the application logic inside classes that will manage the view templates. As an example, the chart component in Figure 14 manages the chart template to render a visualization using HTML[30] and CSS[31].

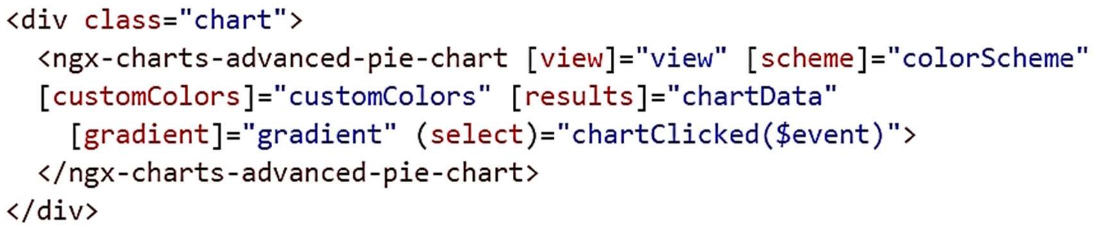
Metadata: The component will be no more than a regular class until Angular is told that it is a component. Metadata defines how Angular will process the class, e.g. by attaching the @Component decorator in Figure 15 to the class.



```
@Component({
  selector: 'chart',
  templateUrl: './chart.component.html',
  styleUrls: ['./chart.component.css']
})
export class ChartComponent implements OnInit {
```

Figure 15: Angular 2 component class structure and metadata

View templates: View templates include HTML that gives directions on how Angular will render the component, e.g. the HTML code in Figure 16 defines how the pie chart element <ngx-charts-advanced-pie-chart> will be rendered.



```
<div class="chart">
  <ngx-charts-advanced-pie-chart [view]="view" [scheme]="colorScheme"
  [customColors]="customColors" [results]="chartData"
    [gradient]="gradient" (select)="chartClicked($event)">
  </ngx-charts-advanced-pie-chart>
</div>
```

Figure 16: Angular 2 template structure and data binding

[30] https://no.wikipedia.org/wiki/HTML
[31] https://en.wikipedia.org/wiki/Cascading_Style_Sheets

Data binding. Data binding facilitates communication between components and templates. There are four types of data binding that are illustrated in Figure 17 [50] below.

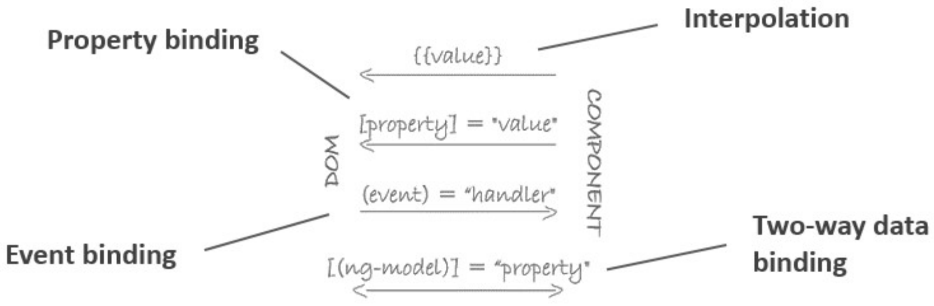
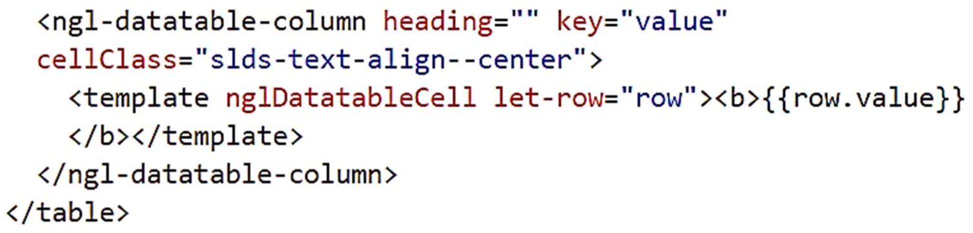


Figure 17: Angular 2 data binding

1. *Interpolation* displays the property value of a component within an HTML element start and end tags, e.g. the template element in Figure 18 will render the component properties {{row.value}} of the JavaScript object 'row'.



```
  <ngl-datatable-column heading="" key="value"
  cellClass="slds-text-align--center">
    <template nglDatatableCell let-row="row"><b>{{row.value}}
    </b></template>
  </ngl-datatable-column>
</table>
```

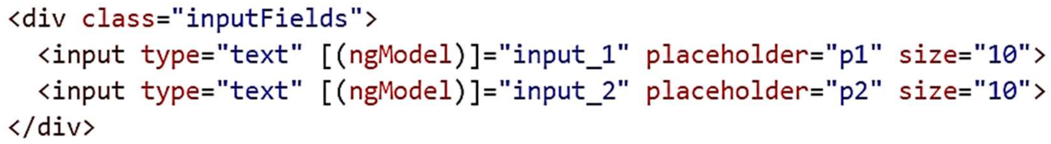
Figure 18: Angular 2 interpolation

2. *Property binding*, also called one-way data binding, passes a property value of a component into a target element property. Property binding can only set the value of a target element, and it is not possible to read it. In Figure 16, the property value 'chartData' sets the target value [results] of the element

<ngx-charts-advanced-pie-chart>. The template then renders to display the updated values from the component.

3. *Event binding* is facilitated by event listeners that will listen for user input and actions, and pass information back to the component.

4. *Two-way data binding* is achieved by using the ngModel directive in Angular 2, combining both property and event binding in a single operation. Figure 19 shows how two-way data binding is facilitated by using the [(ngModel)] directive. The 'input_1' property value is passed from component to template input field. When the user enters a new value, an event is triggered and the updated input value now flows from template to component target property value.



```
<div class="inputFields">
  <input type="text" [(ngModel)]="input_1" placeholder="p1" size="10">
  <input type="text" [(ngModel)]="input_2" placeholder="p2" size="10">
</div>
```

Figure 19: Angular 2 two-way data binding

4.2 Functions

The key functionality that is needed to evaluate usability of the visual data profiling approach is implemented in the prototype. The functionality is based on which data cleaning and transformation steps are needed to demonstrate and validate the visual data profiling approach in a user scenario developed by Statsbygg[32] and SINTEF.

The user scenario is named 'State of Estate', and is based on a dataset that is cleaned and transformed by utilizing Grafterizer [10]. Statsbygg is the Norwegian government's advisor in terms of construction and property affairs, and serves as

[32] http://www.statsbygg.no

a building commissioner, and property manager and developer. One of the purposes of cleaning and transforming the State of Estate dataset, is to integrate information about public buildings in Norway with for example accessibility in buildings [10]. Appendix A of this thesis contains a detailed description of the functions that are applied in the State of Estate user scenario, together with a detailed sequence describing how the steps are applied to the dataset.

In total 14 functions are defined, and implemented in the prototype:

1. *Set first row as header.*

2. *Replace (parameter 1) with (parameter 2)*, e.g. replace ',' with '.'

3. *Set text to uppercase letters.*

4. *Pad a string value with zeros until length 4*, i.e. insert zeros to the end of a value until the total length of the value is 4. As an example, the function will add '00' to the value '33', resulting in a new string value '3300' with length 4.

5. *Fill empty cells with a zero.*

6. *Reformat date values.* This function reformats date values to the format 'dd.MM.yyyy'. If a cell is empty, the cell value is set to 01.01.1753.

7. *Concatenate values and separate with '/'.* As an example, consider two columns with the attributes 'Column 1' and 'Column 2'. The function will combine the attribute values of each row into a new string value:

Column 1	Column 2	Result of function 7
Value1	Value2	'Value1/Value2'

8. *Concatenate values, no separation between values.* This function is identical to function 7, except that there is no '/' that separates the values.

9. *Insert column to the right.*

10. *Insert column to the left*.

11. *Insert row above*.

12. *Insert row below*.

13. *Remove column*.

14. *Remove row*.

4.3 Implementation

This part of the chapter describes the process of realizing the visual data profiling approach in a software prototype.

Prototyping is an iterative design and development process that is widely used in web development to build software features and evaluate requirements [17]–[19]. Hence, prototyping is an effective strategy to develop and evaluate a visual data profiling approach. The main principle behind the process is to always have something functional to test with users, implement changes based on the feedback, and then iterate.

The prototype adds interactivity to the wireframe that was developed in Chapter 3, and provides functionality needed to demonstrate and validate the visual data profiling approach during project meetings where proDataMarket, EW-Shopp, and euBusinessGraph participants are present.

The following technologies are involved in the development of the prototype:

- *Angular 2*[33] – JavaScript/ TypeScript development framework for building desktop and mobile web applications. Angular 2 builds on best practices from Angular JS[34].

[33] https://angular.io
[34] https://github.com/angular/angular.js

- *Angular CLI*[35] – command line interface tool that generates and serves Angular projects on a development server. The CLI provides scaffolding and a test suite for all Angular components.

- *TypeScript*[36] – superset of JavaScript that compiles to plain JavaScript. TypeScript adds class-based object-oriented programming to JavaScript and supports ECMAScript 2015[37]. JavaScript is an implementation of the ECMAScript object-oriented programming standard.

- *NodeJS*[38] – server side JavaScript runtime environment. The package ecosystem of Node, npm, provides a large ecosystem of open source libraries.

- *Handsontable Community Edition*[39] – JavaScript spreadsheet library for apps and websites. The library is fully customizable and provides Excel-like user experience.

- *Datalib*[40] – JavaScript library that provides functionality for data loading, type inference and statistics. Developed by the founders of Trifacta Data Wrangler[41].

- *Plotly.js*[42] and *ngx-charts*[43]– open source high-level declarative visualization libraries built on top of *d3.js*[44]. The libraries include various statistical charts such as box plots, histograms and pie charts.

- *Clarity Design System*[45] – HTML and CSS framework.

[35] https://cli.angular.io
[36] https://www.typescriptlang.org
[37] http://www.ecma-international.org/ecma-262/6.0
[38] https://nodejs.org/en
[39] https://handsontable.com/
[40] https://vega.github.io/datalib
[41] https://www.trifacta.com
[42] https://plot.ly/javascript
[43] https://swimlane.github.io/ngx-charts
[44] https://d3js.org
[45] https://vmware.github.io/clarity

- *Ng-Lightning*[46] – native Angular 2 components.

There are several reasons for selecting the specific approaches and technologies above.

First, Grafterizer is currently implemented in AngularJS, the JavaScript web development framework that precedes Angular 2. Angular 2 is more object-oriented and offers significant advantages over AngularJS in terms of its improved speed and performance, increased modularity of code, more effective data binding between the view layer and the business logic layer, in addition to improved unit testing. The Angular 2 project structure generated by the Angular 2 CLI is also well suited for collaboration between research development teams.

Second, Angular 2 works with both plain JavaScript and the newest features of JavaScript and TypeScript, i.e. features from the latest versions of JavaScript (EcmaScript), that improve code quality and structure.

Third, Handsontable is a robust spreadsheet library with several built-in features, and extensive customization capabilities. Using an existing library to build a tabular spreadsheet view is cost and time effective.

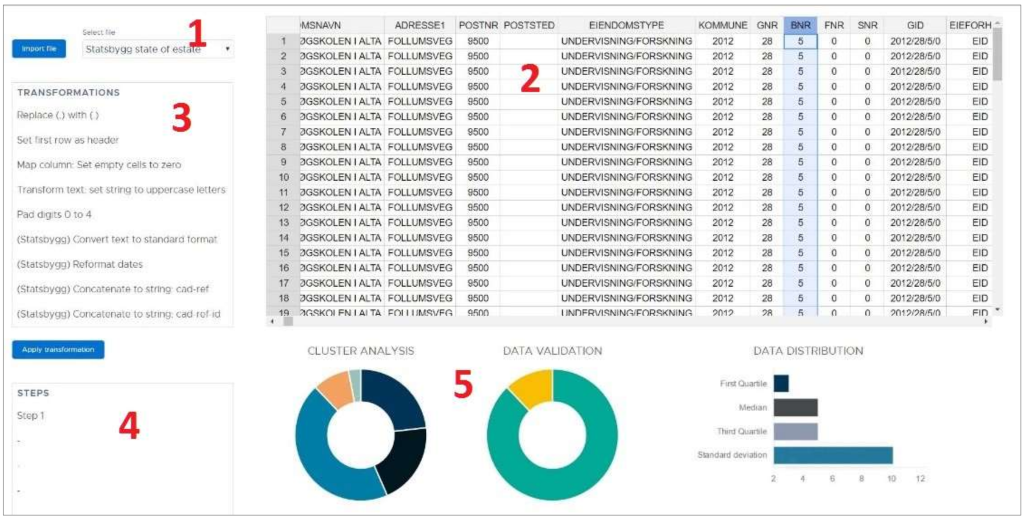
Finally, the statistical library Datalib builds on state of the art research within data profiling, and contains all the functionality needed to develop a prototype. Furthermore, the use of high-level declarative visualization libraries reduces the specification burden of declaring visualization functionality.

Implementation of Prototype, 1st Iteration

The first iteration of the prototype in Figure 20 below implements the very first features necessary to validate the prototype during the first survey session. The first iteration includes the table view, a basic visual data profiling service, and a sidebar menu with suggested transformations and a steps pipeline. This version focuses on implementing main components and internal logic, and validating

[46] http://ng-lightning.github.io/ng-lightning

basic functionality. Hence, there is less focus on implementing appropriate and consistent visual elements (e.g. colors, text, and layout).


1
2
3
4
5
Import file
Select file
Statsbygg state of estate
TRANSFORMATIONS
Replace (.) with ( )
Set first row as header
Map column: Set empty cells to zero
Transform text: set string to uppercase letters
Pad digits 0 to 4
(Statsbygg) Convert text to standard format
(Statsbygg) Reformat dates
(Statsbygg) Concatenate to string: cad-ref
(Statsbygg) Concatenate to string: cad-ref-id
Apply transformation
STEPS
Step 1
-
-
-
CLUSTER ANALYSIS
DATA VALIDATION
DATA DISTRIBUTION
First Quartile
Median
Third Quartile
Standard deviation
2
4
6
8
10
12

	MSNAVN	ADRESSE1	POSTNR	POSTSTED	EIENDOMSTYPE	KOMMUNE	GNR	BNR	FNR	SNR	GID	EIEFORH
1	ØGSKOLEN I ALTA	FOLLUMSVEG	9500		UNDERVISNING/FORSKNING	2012	28	5	0	0	2012/28/5/0	EID
2	ØGSKOLEN I ALTA	FOLLUMSVEG	9500		UNDERVISNING/FORSKNING	2012	28	5	0	0	2012/28/5/0	EID
3	ØGSKOLEN I ALTA	FOLLUMSVEG	9500		UNDERVISNING/FORSKNING	2012	28	5	0	0	2012/28/5/0	EID
4	ØGSKOLEN I ALTA	FOLLUMSVEG	9500		UNDERVISNING/FORSKNING	2012	28	5	0	0	2012/28/5/0	EID
5	ØGSKOLEN I ALTA	FOLLUMSVEG	9500		UNDERVISNING/FORSKNING	2012	28	5	0	0	2012/28/5/0	EID
6	ØGSKOLEN I ALTA	FOLLUMSVEG	9500		UNDERVISNING/FORSKNING	2012	28	5	0	0	2012/28/5/0	EID
7	ØGSKOLEN I ALTA	FOLLUMSVEG	9500		UNDERVISNING/FORSKNING	2012	28	5	0	0	2012/28/5/0	EID
8	ØGSKOLEN I ALTA	FOLLUMSVEG	9500		UNDERVISNING/FORSKNING	2012	28	5	0	0	2012/28/5/0	EID
9	ØGSKOLEN I ALTA	FOLLUMSVEG	9500		UNDERVISNING/FORSKNING	2012	28	5	0	0	2012/28/5/0	EID
10	ØGSKOLEN I ALTA	FOLLUMSVEG	9500		UNDERVISNING/FORSKNING	2012	28	5	0	0	2012/28/5/0	EID
11	ØGSKOLEN I ALTA	FOLLUMSVEG	9500		UNDERVISNING/FORSKNING	2012	28	5	0	0	2012/28/5/0	EID
12	ØGSKOLEN I ALTA	FOLLUMSVEG	9500		UNDERVISNING/FORSKNING	2012	28	5	0	0	2012/28/5/0	EID
13	ØGSKOLEN I ALTA	FOLLUMSVEG	9500		UNDERVISNING/FORSKNING	2012	28	5	0	0	2012/28/5/0	EID
14	ØGSKOLEN I ALTA	FOLLUMSVEG	9500		UNDERVISNING/FORSKNING	2012	28	5	0	0	2012/28/5/0	EID
15	ØGSKOLEN I ALTA	FOLLUMSVEG	9500		UNDERVISNING/FORSKNING	2012	28	5	0	0	2012/28/5/0	EID
16	ØGSKOLEN I ALTA	FOLLUMSVEG	9500		UNDERVISNING/FORSKNING	2012	28	5	0	0	2012/28/5/0	EID
17	ØGSKOLEN I ALTA	FOLLUMSVEG	9500		UNDERVISNING/FORSKNING	2012	28	5	0	0	2012/28/5/0	EID
18	ØGSKOLEN I ALTA	FOLLUMSVEG	9500		UNDERVISNING/FORSKNING	2012	28	5	0	0	2012/28/5/0	EID
19	ØGSKOLEN I ALTA	FOLLUMSVEG	9500		UNDERVISNING/FORSKNING	2012	28	5	0	0	2012/28/5/0	EID

Figure 20: Implementation of prototype, 1st iteration

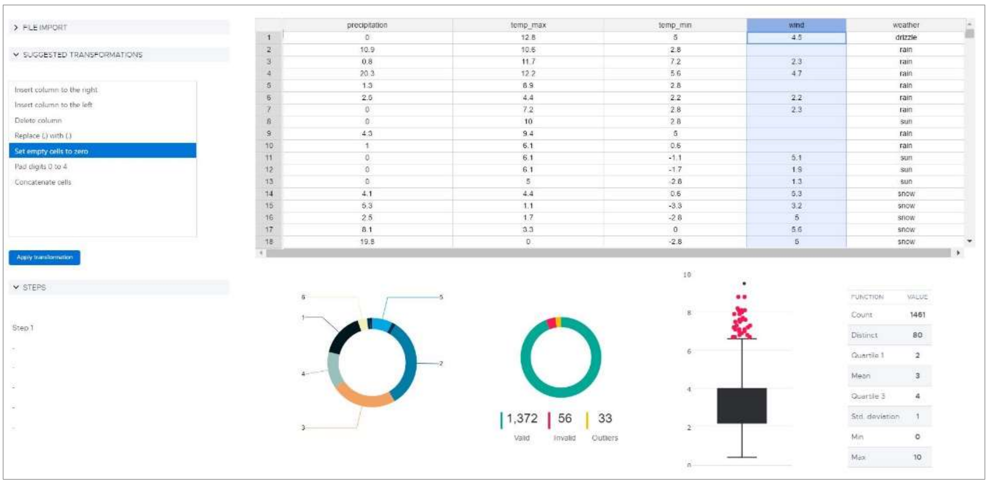
The prototype in Figure 20 implements basic functionality of the following components:

- Component **1**, the **file import**, is implemented for prototype development purposes only.
- Component **2**, the **table** view, is a direct-manipulation table with Excel-like features such as right-clicking functionality (e.g. copy/ paste, and insert column/ row).
- Component **3**, the **transformations** sidebar, lists all transformations without considering data type, or whether a column or row is selected. Later iterations implement a rule-based system that suggests relevant data cleaning and transformation procedures.

- Component **4**, the **steps** pipeline, displays the first step of data cleaning or transformation applied. Later iterations implement a functioning pipeline that reflects all steps applied.

- Component **5**, the **visual data profiling** service, features (from left to right) a data distribution chart, a chart that displays number of missing values, and basic measures of central tendency.

Implementation of Prototype, 2nd Iteration



	precipitation	temp_max	temp_min	wind	weather
1	0	12.8	5	4.5	drizzle
2	10.9	10.6	2.8		rain
3	0.8	11.7	7.2	2.3	rain
4	20.3	12.2	5.6	4.7	rain
5	1.3	8.9	2.8		rain
6	2.5	4.4	2.2	2.2	rain
7	0	7.2	2.8	2.3	rain
8	0	10	2.8		sun
9	4.3	9.4	5		rain
10	1	6.1	0.6		rain
11	0	6.1	-1.1	5.1	sun
12	0	6.1	-1.7	1.9	sun
13	0	5	-2.8	1.3	sun
14	4.1	4.4	0.6	5.3	snow
15	5.3	1.1	-3.3	3.2	snow
16	2.5	1.7	-2.8	5	snow
17	8.1	3.3	0	5.6	snow
18	19.8	0	-2.8	5	snow

FUNCTION	VALUE
Count	1461
Distinct	80
Quartile 1	2
Mean	3
Quartile 3	4
Std. deviation	1
Min	0
Max	10

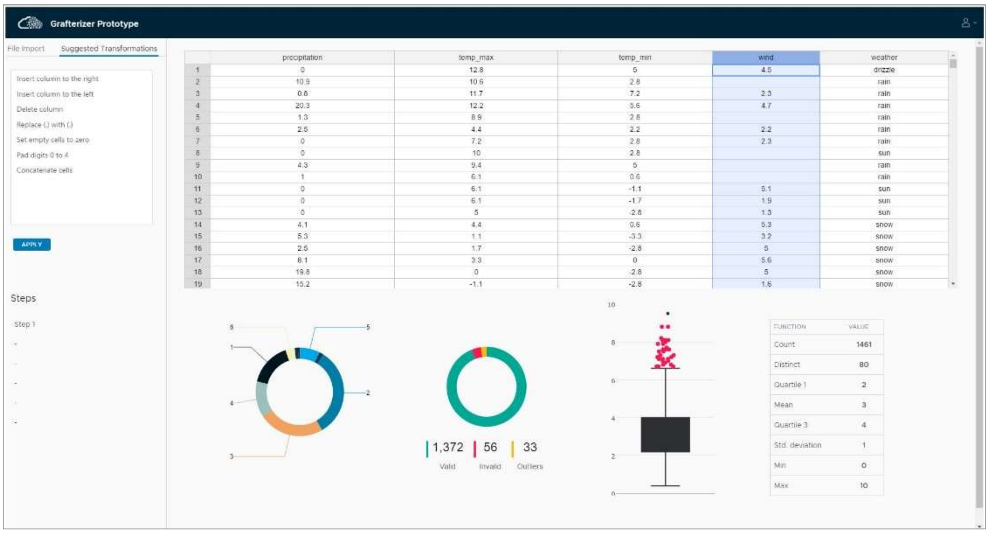
Figure 21: Implementation of prototype, 2nd iteration

Feedback from the comparative usability test after a demonstration and validation of the first iteration of the prototype, indicates that the following changes should be implemented in the second iteration (i.e. Figure 21):

- Include outlier detection in charts to identify extreme values.
- Include a boxplot chart to visualize data distribution and potential outliers.
- Include more common statistics for data distribution and central tendency (i.e. value count, distinct values, quartiles, mean, standard deviation, and minimum and maximum values).

The second iteration of the prototype in Figure 21 includes a visual data profiling service in which the logic behind the service is improved to provide more detailed statistical profiles. Furthermore, the iteration implements outlier detection, a boxplot chart and a statistics summary table.

Implementation of Prototype, 3rd Iteration



	precipitation	temp_max	temp_min	wind	weather
1	0	12.8	5	4.5	drizzle
2	10.9	10.6	2.8		rain
3	0.8	11.7	7.2	2.3	rain
4	20.3	12.2	5.6	4.7	rain
5	1.3	8.9	2.8		rain
6	2.5	4.4	2.2	2.2	rain
7	0	7.2	2.8	2.3	rain
8	0	10	2.8		sun
9	4.3	9.4	5		rain
10	1	6.1	0.6		rain
11	0	6.1	-1.1	5.1	sun
12	0	6.1	-1.7	1.9	sun
13	0	5	-2.8	1.3	sun
14	4.1	4.4	0.6	5.3	snow
15	5.3	1.1	-3.3	3.2	snow
16	2.5	1.7	-2.8	5	snow
17	8.1	3.3	0	5.6	snow
18	19.8	0	-2.8	5	snow
19	15.2	-1.1	-2.8	1.6	snow

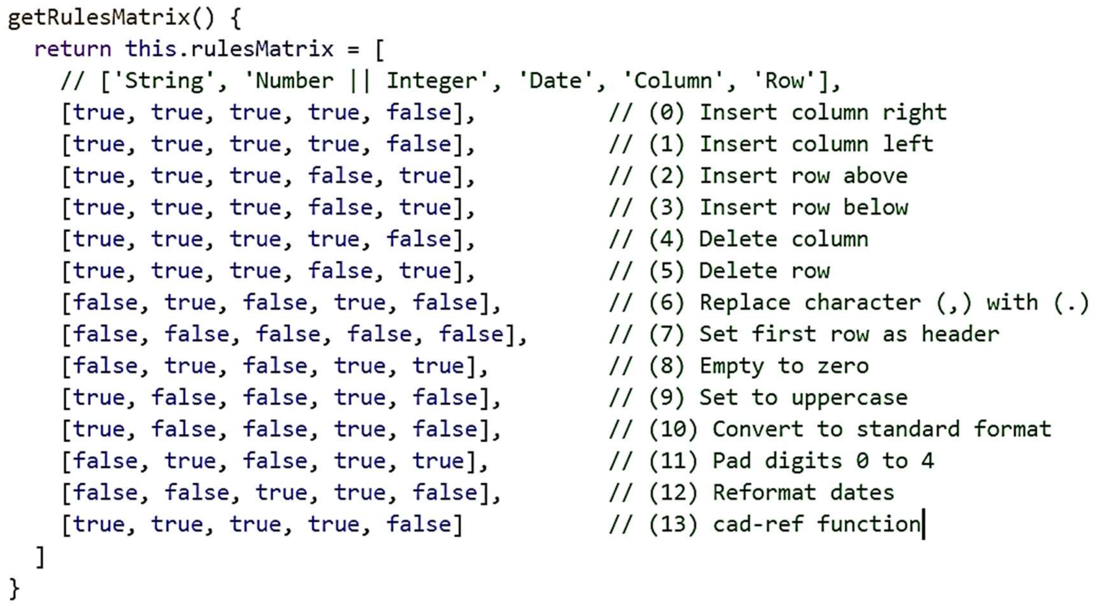
FUNCTION	VALUE
Count	1461
Distinct	80
Quartile 1	2
Mean	3
Quartile 3	4
Std. deviation	1
Min	0
Max	10

Figure 22: Implementation of prototype, 3rd iteration

Feedback from survey participants during the demonstration of the second iteration of the coded prototype, propose that the following changes are implemented in the next iteration to improve the quality of the visual data profiling approach:

- Suggest only data cleaning and transformation procedures that are applicable and relevant to the current data selection, i.e. implement a data profiling rules matrix.
- Show actual data cleaning and transformation steps in the steps pipeline.

The third iteration of the prototype illustrated in Figure 22 includes an implementation of logic for suggesting transformations, based on a rules matrix (i.e. Figure 23).

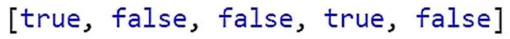


```
getRulesMatrix() {
  return this.rulesMatrix = [
    // ['String', 'Number || Integer', 'Date', 'Column', 'Row'],
    [true, true, true, true, false],          // (0) Insert column right
    [true, true, true, true, false],          // (1) Insert column left
    [true, true, true, false, true],          // (2) Insert row above
    [true, true, true, false, true],          // (3) Insert row below
    [true, true, true, true, false],          // (4) Delete column
    [true, true, true, false, true],          // (5) Delete row
    [false, true, false, true, false],        // (6) Replace character (,) with (.)
    [false, false, false, false, false],      // (7) Set first row as header
    [false, true, false, true, true],         // (8) Empty to zero
    [true, false, false, true, false],        // (9) Set to uppercase
    [true, false, false, true, false],        // (10) Convert to standard format
    [false, true, false, true, true],         // (11) Pad digits 0 to 4
    [false, false, true, true, false],        // (12) Reformat dates
    [true, true, true, true, false]           // (13) cad-ref function
  ]
}
```

Figure 23: Visual data profiling rules matrix

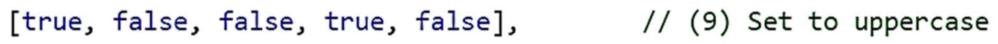
The application checks the statistical data profile that is returned by the visual data profiling service against the rules matrix illustrated in Figure 23. Consider the following example:

The data profiling service returns a profile of the current data selected in a column of string values. The String value of the profile array is *true*, while the Number, Date and Row values are *false*:



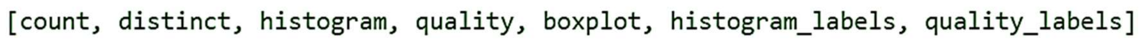
```
[true, false, false, true, false]
```

The application checks this profile against the enumerated list of transformations in Figure 23, and matches the profile array with the rules array function number 9, 'Set to uppercase', as a possible transformation that will be suggested:



```
[true, false, false, true, false],        // (9) Set to uppercase
```

The visual data profiling service analyzes and assesses the quality of the dataset, and returns a statistical profile. This profile is an essential part of the underlying core application logic that suggests transformations and renders profiling charts.



```
[count, distinct, histogram, quality, boxplot, histogram_labels, quality_labels]
```

Figure 24: Visual data profiling statistical assessment profile

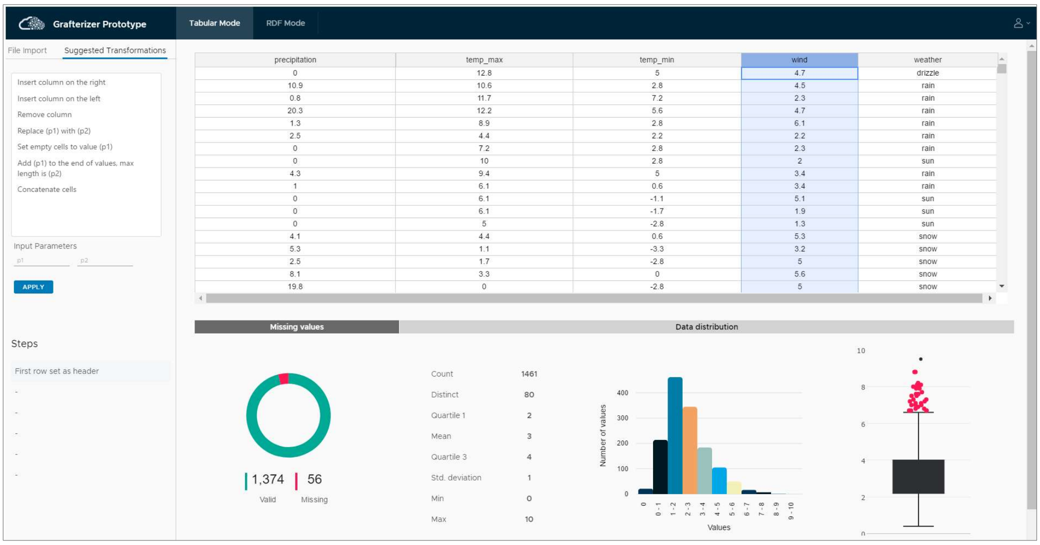
The visual data profiling statistical assessment profile includes the following information illustrated in Figure 24:

- **Count** – the total number of values in the selected column.
- **Distinct** – the number of unique values. As an example, a column attribute 'week' might count in total 1000 rows and 7 unique values, one for each day.
- **Histogram** – an array containing one value for each histogram bin.
- **Quality** – an array that contains three different values, one value representing valid entries, one for invalid entries and another one for outliers.
- **Boxplot** – array that contains all values necessary to render a boxplot chart, i.e. the first, second and third quartiles, and the median.
- **Histogram_labels** – labels for the histogram chart visualization.
- **Quality_labels** – labels for the quality chart visualization.

Furthermore, in terms of changes implemented in the third iteration of the prototype, the logic behind the steps pipeline is improved to save a snapshot of the current dataset at each step of the data cleaning and transformation pipeline.

The steps pipeline automatically updates and reflects the applied transformations at any given time. The user may click a specific step to view and edit applied transformations until that step of the process. The steps pipeline also serves as a function to undo operations.

Implementation of Prototype, Final Iteration



precipitation	temp_max	temp_min	wind	weather
0	12.8	5	4.7	drizzle
10.9	10.6	2.8	4.5	rain
0.8	11.7	7.2	2.3	rain
20.3	12.2	5.6	4.7	rain
1.3	8.9	2.8	6.1	rain
2.5	4.4	2.2	2.2	rain
0	7.2	2.8	2.3	rain
0	10	2.8	2	sun
4.3	9.4	5	3.4	rain
1	6.1	0.6	3.4	rain
0	6.1	-1.1	5.1	sun
0	6.1	-1.7	1.9	sun
0	5	-2.8	1.3	sun
4.1	4.4	0.6	5.3	snow
5.3	1.1	-3.3	3.2	snow
2.5	1.7	-2.8	5	snow
8.1	3.3	0	5.6	snow
19.8	0	-2.8	5	snow

Count	1461
Distinct	80
Quartile 1	2
Mean	3
Quartile 3	4
Std. deviation	1
Min	0
Max	10

Figure 25: Implementation of prototype, final iteration

The final iteration of the prototype in Figure 25 implements the following changes based on feedback from the survey session in which prototype iteration number three was presented:

- The leftmost data profiling chart represents missing values and valid (non-null) values for the currently selected column. Outliers are removed from this chart since it causes confusion to mix a representation of missing values and outliers.

- The three remaining charts (from left to right) represent the distribution of the currently selected column. The pie chart has been replaced with a histogram chart that is a more accurate chart type to visualize distribution.

5 Evaluation

5.1 Validation of Functional Requirements

The Statsbygg State of Estate user scenario that was described in Chapter 4 (and also referenced in Appendix A), has been used extensively throughout the iterative implementation phases to validate functional requirements of the prototype. Hence, the user scenario has been used to evaluate whether the prototype has succeeded in implementing the intended requirements that were defined in Chapter 3.

An additional user scenario has been included to validate functionality. This scenario is a generic dataset that measures Seattle minimum and maximum temperatures, precipitation, and wind on specific days with respectively sun, snow, drizzle or rain. The 1463 rows CSV dataset[47] is selected because of its suitability to run and test statistical functionality and features that are implemented as part of the visual data profiling approach. The dataset is originally used for testing purposes for the Datalib statistical library and Vega visualizations[48]. In terms of demonstrating visual data profiling capabilities, some attribute values have been removed to introduce missing value anomalies in the dataset.

Table 3 below shows the status of capabilities implemented in each prototype iteration (1 – 4). Additionally, the table sums up how the implementation and validation of each capability satisfies a requirement for the visual data profiling approach. A green cell in Table 3 indicates that the specific capability, or feature, has been demonstrated to serve its intended purpose in the validation sessions.

[47] https://github.com/vega/vega-datasets/blob/gh-pages/data/seattle-weather.csv

[48] https://github.com/vega/vega-datasets

Table 3: Validation of visual data profiling approach

Capabilities implemented in the prototype to realize the visual data profiling approach	**R**	**Prototype iteration**			
		1	**2**	**3**	**4**
Table	**R2**	🟩	🟩	🟩	🟩
Transformation sidebar	**R2**	🟩	🟩	🟩	🟩
Steps pipeline	**R4**	🟨	🟨	🟩	🟩
Data profiling service	**R1**	🟨	🟩	🟩	🟩
Transformation service	**R2**	🟨	🟨	🟩	🟩
Rule-based transformation service	**R3**	🟥	🟥	🟩	🟩
Statistical profiling: distribution table	**R1**	🟥	🟩	🟩	🟩
Statistical profiling: quality pie chart	**R1**	🟨	🟨	🟨	🟩
Statistical profiling: box plot chart	**R1**	🟥	🟩	🟩	🟩
Statistical profiling: histogram chart	**R1**	🟥	🟥	🟥	🟩

R Requirement
🟥 Not yet implemented
🟨 Partially implemented – needs refinement
🟩 Implemented

Based on this evaluation, the following requirements are successfully met through five iterations of prototyping and user scenario validation (i.e. Table 4 below).

Table 4: Validation of functional requirements

Requirements of the visual data profiling approach		Type	Validated
R1	Provide visual data profiling capabilities	**F**	✓
R2	Provide data cleaning and transformation functionality	**F**	✓
R3	Provide data cleaning and transformation suggestions	**F**	✓
R4	Provide a pipeline that reflects applied data cleaning and transformation steps	**F**	✓
R5	Provide a solution that is useful to the user	**NF**	Usability testing required
R6	Provide a solution that is easy to use	**NF**	Usability testing required

Functional requirement	**F**
Non-functional requirement	**NF**

The non-functional usability requirements R5 and R6 are evaluated and validated in Chapter 5.2. While requirements R1 to R4 can be validated by evaluating the extent of successfully implemented features through a user scenario, validation of usability requirements demands specific methods that would involve users.

Chapter 6 will summarize and conclude the validation of the functional requirements and non-functional usability requirements.

5.2 Validation of Usability Requirements

The three sessions of comparative usability testing and surveys have resulted in statistics about how users perceive the current version of Grafterizer compared to the visual data profiling approach in terms of usefulness and ease of use.

This section of the chapter will analyze the findings from the comparative usability test and the streamlined cognitive walkthrough.

5.2.1 Analysis of Findings from the Comparative Usability Test

In total 24 participants, which corresponds to the sample size from the three project meetings, responded to the survey questionnaire. The same users have evaluated both the current version of Grafterizer and the visual data profiling prototype, which defines the test setup as a within-subjects design [20]. The advantage of using this type of test design, is that it removes some sources of variation in the datasets, as compared to between-subjects design where different users test each version of the application.

The survey questionnaire asked respondents to rate each application on the dimensions of usefulness and ease of use, respectively:

a. Usefulness

Q1 Using Grafterizer in my job would enable me to accomplish tasks more quickly.
Q2 Using Grafterizer would improve my job performance.
Q3 Using Grafterizer in my job would increase my productivity.
Q4 Using Grafterizer would enhance my effectiveness on the job.
Q5 Using Grafterizer would make it easier to do my job.
Q6 I would find Grafterizer useful in my job.

b. Ease of use

Q1 Learning to operate Grafterizer would be easy for me.

Q2 I would find it easy to get Grafterizer to do what I want it to do.
Q3 My interaction with Grafterizer would be clear and understandable.
Q4 I would find Grafterizer to be flexible to interact with.
Q5 It would be easy for me to become skillful at using Grafterizer.
Q6 I would find Grafterizer easy to use.

The summarized results from all respondents are illustrated in Figure 26 and 27 below. The figures indicate the mean value of each question asked, e.g. the rating score of question Q1 in Figure 26 shows the average of all 24 respondents' rating score on that specific question. High rating scores equal high agreement with questions asked, while low scores correspond to disagreement with questions.

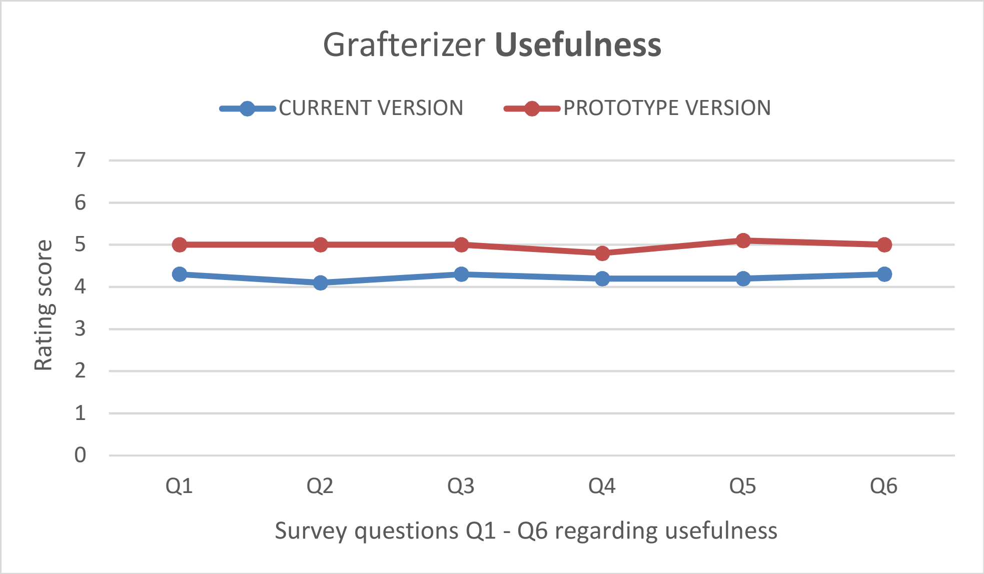


Figure 26: Comparative usability test results (usefulness)

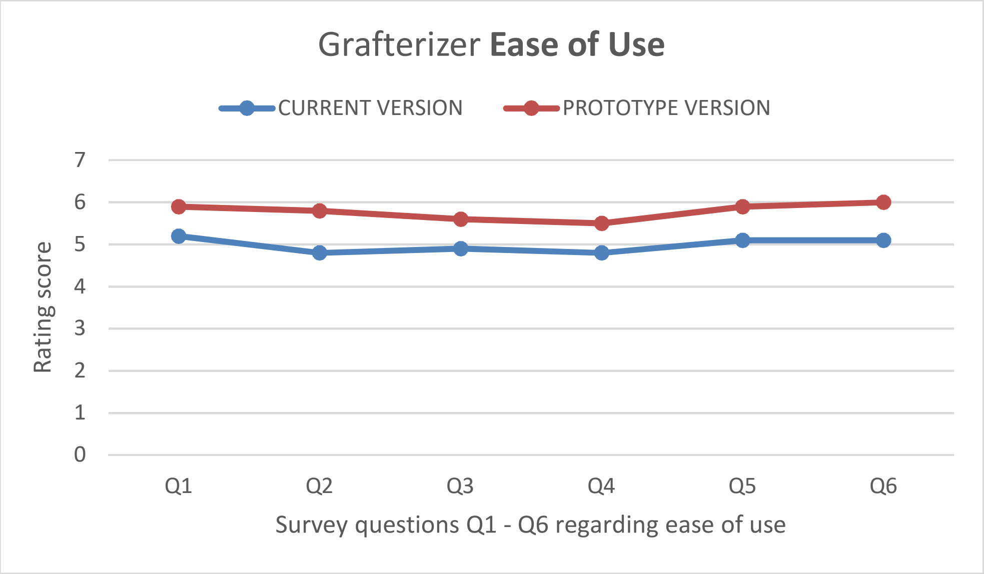


Figure 27: Comparative usability test results (ease of use)

The results that are illustrated in Figure 26 and 27 indicate that the visual data profiling approach consistently is rated higher than the current version of Grafterizer on both dimensions of usefulness and ease of use.

It might be tempting to conclude that these results show that the prototype is better than the current version, but it is insufficient to draw such conclusions based only on the kind of descriptive statistics [20] we find in Figure 26 and 27. We need to determine if this difference between the applications is statistically significant, and if it is larger than we would expect from pure chance [20].

Since the usability test is a within-subject comparison of two applications, and the survey test results are continues values, a paired t-test can be applied to appropriately determine if there is a significant difference between the mean ratings of the two applications [20].

The approach suggested by Sauro and Lewis [20] is applied to compare the mean rating between the prototype and the current version of Grafterizer. This approach is used throughout this section of the chapter.

Analysis of Findings from the Comparative Usability Test in Terms of Usefulness

The following formula can be used to determine the statistical significance:

$$t = \frac{\widehat{D}}{\frac{s_D}{\sqrt{n}}}$$

where

$\widehat{D}$ is the mean of the difference between the scores

s_D is the standard deviation of the difference between the scores

n is the sample size, i.e. the number of survey respondents

t is the test statistic

Equation 7: Paired t-test

Using the t-test from Equation 7 to calculate the test statistic *t* of the values in Table 5 below, we get the following *t* value:

$$t = \frac{4.63}{\frac{4.48}{\sqrt{24}}} = 5.09$$

To determine whether the *t* value is significant, we use the TDIST function in Excel:

TDIST(*t* value, degrees of freedom, one-sided = 1/ two-sided = 2)

Equation 8: TDIST function

In terms of Equation 8, the degrees of freedom are equal to $n - 1$, and we use a two-sided test in the comparison. *n* = 24, which leads to the following calculation:

TDIST(5.09, 23, 2) = 0,000037

This value is very small, and indicates that we can be approximately 99.999% sure that the prototype and the current version have different scores, i.e. the difference is not due to chance. Hence, the prototype's rating score of 30 is statistically significantly higher than the current version's score of 25.4.

We will conclude that the users perceive that the prototype is more useful than the current version of Grafterizer.

Table 5: Survey rating scores, and difference, in terms of usefulness

Survey rating scores for USEFULNESS dimension			
Respondent	**Prototype**	**Current version**	**Difference**
1	6	6	0
2	10	11	-1
3	24	24	0
4	34	35	-1
5	31	24	7
6	26	26	0
7	32	24	8
8	38	36	2
9	19	18	1
10	34	27	7
11	34	32	2
12	26	14	12
13	28	25	3
14	11	10	1
15	30	19	11
16	39	24	15
17	36	30	6
18	35	29	6
19	38	37	1
20	40	32	8
21	42	34	8
22	33	31	2
23	36	32	4
24	38	29	9
Mean	**30**	**25.4**	**4.63**

Analysis of Findings from the Comparative Usability Test in Terms of Ease of Use

In the section of the comparative usability study that compares ease of use between the two applications, we apply the same t-test (i.e. Equation 7) to measure statistical significance. The survey ratings for the prototype and the current version of Grafterizer are listed in Table 6 below.

Table 6: Survey rating scores, and difference, in terms of ease of use

Survey rating scores for EASE OF USE dimension			
Respondent	**Prototype**	**Current version**	**Difference**
1	38	26	12
2	30	30	0
3	32	34	-2
4	39	36	3
5	28	23	5
6	27	21	6
7	30	24	6
8	40	35	5
9	33	33	0
10	36	30	6
11	38	35	3
12	33	12	21
13	29	31	-2
14	36	36	0
15	36	36	0
16	36	30	6
17	36	30	6
18	27	26	1
19	38	35	3
20	42	36	6
21	42	39	3
22	35	31	4
23	31	30	1
24	38	21	17
Mean	**34.6**	**30**	**4.58**

We use the t-test from Equation 7 to calculate the test statistic *t* of the values in Table 6, and end up with the following *t* value:

$$t = \frac{4.58}{\frac{5.51}{\sqrt{24}}} = 4.07$$

To determine whether the *t* value is significant, we use the TDIST function in Excel. In terms of Equation 8, the degrees of freedom are equal to *n* – 1, while we use a two-sided test to compare the applications. Since *n* = 24, this gives us the following calculation:

TDIST(4.07, 23, 2) = 0.00047

Like the analysis of the usefulness score, this value is small, indicating that we can be approximately 99.999% sure that the prototype and the current version have different scores, i.e. the difference is not due to chance. Hence, the prototype's rating score of 34.6 is statistically significantly higher than the current version's score of 30.

Based on the above analysis, we will conclude that the users perceive that the prototype is easier to use compared to the current version of Grafterizer.

5.2.2 Analysis of Findings from the Cognitive Walkthrough

Revisiting the purpose of using the method to evaluate usability of the visual data profiling approach, we want to understand the learnability of the prototype, i.e. how easy it is for new users to learn how to use the functionality, and develop skills needed to perform basic and necessary tasks [24]. The purpose of applying the evaluation method in the context of this thesis, is to identify usability issues that are introduced by the visual data profiling approach in data cleaning and transformation processes. The results of this evaluation are used to identify and propose future research initiatives in Chapter 6 that would contribute to extending and improving visual data profiling approaches.

The two groups of expert reviewers went through user scenarios that are divided into tasks of the following format:

Task 1

I want to set first row as header.

Expert evaluation (questions answered by reviewers):

- **a.** *Will the user know what to do next?*
- **b.** *Will the user get appropriate feedback if the correct action is taken?*

The sessions resulted in an eight pages long document that describes the responses from the reviewers, and includes a discussion of the findings. The document is omitted from this thesis to preserve privacy of expert reviewers, but the questions that each group received, can be found in the Appendices section as Appendix B.

To categorize and analyze the findings from the streamlined cognitive walkthrough sessions, we differentiate between two methods, based on the approach of Barnum [8]:

- A *top-down* approach that starts with predefined categories and codes.
- A *bottom-up* approach which starts with individual findings that are clustered into groups and labeled according to category. Categories emerge as the process moves forward.

A bottom-up approach is used in this thesis to organize and analyze the findings from the sessions. By using this method, we emphasize the advantage it provides by keeping the researcher open to the results the process will reveal. The method requires more time to organize and analyze than would a top-down approach that starts with predefined concepts, but this disadvantage is outweighed by the potential of identifying more usability issues.

The main findings from the reviews are summarized and categorized in Table 7 below. With each type of usability issue follows a suggestion on how the issue could be corrected. The suggestions are discussed as further research opportunities in Chapter 6.

Table 7: Identified usability issues and suggestions for further research

CATEGORY	USABILITY ISSUES	SUGGESTIONS FOR FURTHER RESEARCH
Visual data profiling	• Some of the charts are not domain specific enough. • The functionality and purpose of each visual data profiling chart are not clear. • Outlier detection and correction of missing values are too generic.	• Explore visual recommender system approaches to suggest relevant and domain specific charts to the user. • Explore approaches that include multivariate data profiling (i.e. by profiling two or more columns to reveal relevant information related to data cleaning and transformation).
'Excel' table view	• Missing information about data type of selected values. Lack of possibility to specify parameters directly in the table view.	• Explore direct table manipulation approaches to data cleaning and transformation to extend capabilities of the tabular table view.
'Suggested transformations' sidebar	• The sidebar is overlooked/ ignored in several cases because the suggested transformations are too generic and not specifically aimed at the current dataset. • Users also prefer to use the right-clicking functionality of the Excel-like table view.	• Explore approaches within predictive data cleaning and transformation, based on machine learning techniques, to provide more intelligent and relevant suggestions.

In terms of *learnability* of the visual data profiling approach, the expert reviews show that the system needs to recommend charts that are domain specific and relevant to the user. This improvement will probably increase the speed, and ease of use, of learning new and basic functionality to perform the specific data cleaning and transformation tasks. Advanced capabilities (i.e. clicking and zooming charts to display detailed information) are not intuitive, and should be considered moved up one level in the user interface hierarchy to be visible always (e.g. by providing access to detailed information in a drop-down menu). The expert reviews also identified a need for a more consistent pattern of visual data profiling sequences (e.g. every time a user clicks a table column, he or she would know what happens next in the visual data profiling view).

Furthermore, the table view and 'Suggested transformation sidebar' need to be consistent by displaying the exact same range of data cleaning and transformation options. Users were confused when only a subset of options were available when right-clicking the table view. The approach should also consider including a mode where the sidebar 'Suggested transformations' can be hidden on demand by the user to free up more space for the table view.

In general, the expert reviews indicate that users are satisfied with the immediate feedback that the visual data profiling approach provides. Feedback includes information such as status of missing values, potential extreme values, and number of distinct values. Still, the partial lack of explicit feedback after clicking columns and rows of the table view, leads to uncertainty about which parts of the dataset has been profiled. Hence, the visual data profiling approach should provide immediate feedback to the user by indicating which columns or rows have been selected, and indicate the data type of the values).

6 Conclusion

6.1 Evaluation of the Visual Data Profiling Approach in Terms of Requirements

With the increasing amounts of data in today's organizations and businesses, proper data quality has become essential to extract and analyze content from large volume data sources. Incorrect or inconsistent data may distort the results of analysis processes, and reduce the potential benefits of applying data-driven approaches in organizations. Furthermore, data scientists spend more than half of their time on preparing data for analysis. Hence, there are considerable research opportunities to ease the process of data cleaning and transformation, and improve data quality.

As a response to the demand for solutions that improve data quality and reduces time spent on cleaning and transforming data, this thesis proposes a visual data profiling approach that implements powerful visual data profiling capabilities. The visual data profiling approach has been evaluated in terms of usability, and found to be perceived useful and easy to use by users. Furthermore, critical usability issues have been identified and proposed as further work in future iterations of the prototype. The thesis has also contributed to proposing a visual data profiling approach that can be further researched and implemented on the DataGraft platform to extend, or replace, the current version of Grafterizer.

The following discussion sums up whether the requirements from Chapter 3 have been satisfied by the realization of the visual data profiling approach in a software prototype. According to the requirements, the prototype should:

R1 *Provide visual data profiling capabilities.* The visual data profiling capabilities have been successfully implemented in Chapter 4.3 to include common profiling functionality, i.e. outlier detection and identification of missing values.

R2 *Provide data cleaning and transformation functionality.* The prototype has successfully implemented the needed functionality to run two selected scenarios in the validation sessions (Chapter 4.3).

R3 *Provide data cleaning and transformation suggestions.* A rule-based matrix has been implemented in Chapter 4.3 to suggest data cleaning and transformation procedures to users.

R4 *Provide a pipeline that reflects applied data cleaning and transformation steps.* A pipeline has been developed in Chapter 4.3 to reflect applied data cleaning and transformation steps.

R5 *Provide a solution that is useful to the user.* The evaluation of usability requirements in Chapter 5.2 indicates that users perceive the prototype to be more useful than the current version of Grafterizer.

R6 *Provide a solution that is easy to use.* The evaluation of usability requirements in Chapter 5.2 (i.e. the comparative usability test and survey) suggests that users perceive the prototype to be easier to use than the current version of Grafterizer. Still, the expert reviews have identified usability issues that provide future research opportunities (discussed in Chapter 6.2) within visual data profiling approaches.

The evaluation of the work performed as part of this thesis, shows that the proposed visual data profiling approach provides a useful framework for data scientists to clean and transform data effectively and easily.

6.2 Further Work

We have seen from Chapter 6.1 that the prototype meets the stated requirements by providing a visual data profiling approach that is perceived useful and easy to use. Still, the approach can be extended and improved by conducting continued research within the areas that are discussed below.

Visual Recommender System for Data Profiling

Chapter 2 introduced two approaches to developing a visual data profiling system. First, the system could be rule-based only. Second, the system could implement intelligent logic to recommend charts that are relevant to the user. The prototype implements the first approach.

Further research should explore the second approach since a visual recommender system will automatically generate personalized data profiling visualizations. This will again provide an approach that is perceived more useful, domain specific, and easier to use. Figure 28 provides a schematic overview of the approach from user interaction and data input to the final rendering and display of personalized, recommended visualizations.

The recommender system pipeline illustrated in Figure 28, includes the following three steps [37]–[40]:

1. **Preprocessing:** Syntactic and semantic preprocessing of data. Syntax analysis identifies and categorizes data according to standard data types (i.e. nominal, ordinal, and numeric) while semantic analysis defines specific data types (e.g. geographical coordinates from numeric fields).
2. **Stage one: Rule-based recommendations**. An encoding algorithm uses an ontology of patterns to map data to specific chart types. Each data type is mapped according to rules for permitted visual encoding channels (e.g. x/ y scale, size, and color) and allowed chart types (e.g. bar, area, line).

3. **Stage two: Personalized recommendations.** An appropriate recommender system approach and algorithm are applied to filter and validate visualizations from stage one - providing top-N suitable combinations of possible charts to a specific user.

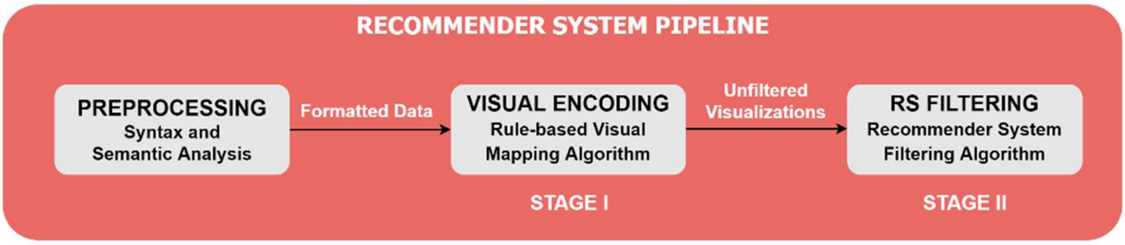


Figure 28: Data profiling visual recommender system pipeline

Multivariate Data Profiling

Multivariate data profiling provides an assessment of data quality in multiple combined columns. The visual data profiling approach that has been proposed in this thesis applies univariate data profiling, i.e. one column at a time. While univariate analysis is capable of uncovering many data quality issues such as missing values and univariate outliers, more useful information is available from multivariate analysis that could assist the user in cleaning and transforming data [1], [5]. As an example, consider the weather data scenario that was used throughout this thesis. We would expect that the temperature on days with snow would be around 0 degrees Celsius, and that the two attributes ‘snow’ and temperature are correlated. If one or several days with snow also display high temperatures, these rows in a dataset might be flagged as potential outliers.

Multivariate analysis would introduce several relevant visualization techniques and charts to the visual data profiling approach. As an example, the scatter plot in Figure 29 [5] shows positive and negative correlations between attributes.

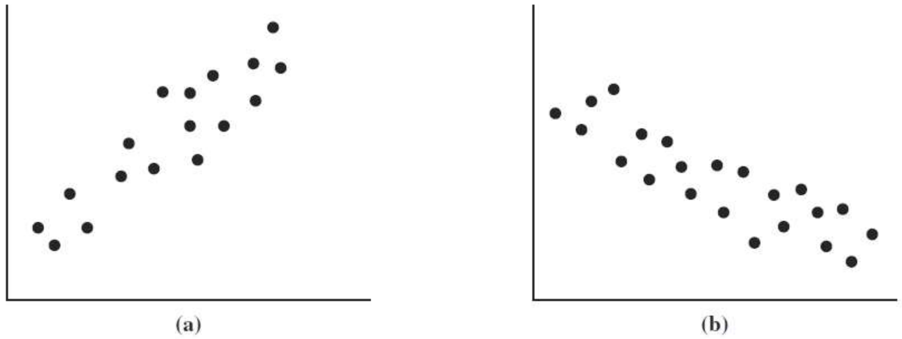


Figure 29: Scatter plots – positive (a) and negative (b) correlations between attributes

Predictive, Intelligent Data Cleaning and Transformation

Data cleaning and transformation processes are most often a domain specific problem that focus on the statistical properties, semantics and structure of data [6]. A visual data profiling approach would benefit from combining a visual recommender system and an intelligent approach to the domain-specific data cleaning and transformation problem. Hence, based on the current column/ row selection in a dataset, the data scientist would be presented only relevant data profiling charts and suggestions for data cleaning and transformation.

An intelligent approach could be incremental, like the one that is applied in the Grafterizer framework, in which the user iteratively applies transformations in a pipeline process. Still, Grafterizer, and the proposed visual data profiling approach, lack an intelligent system that learns from previous tasks to predict useful data cleaning and transformation actions.

Such an approach could be based on the work of Heer et al. [6] that involves a framework for predictive interaction. This framework relieves the burden of technical specification in a domain specific language, and guides the user through an incremental process of cleaning and transforming data. The system intelligently predicts the next interaction, and the user is involved to judge whether the next step is relevant or needs to be modified.

Direct Manipulation Interfaces

Direct manipulation interfaces (e.g. Microsoft Excel spreadsheet) is well known to most data scientists. The visual data profiling approach proposed in this thesis provides a direct manipulation interface in a spreadsheet style table view that dynamically integrates with the approach. Furthermore, the expert reviews indicate that users prefer a direct manipulation spreadsheet for basic data cleaning and transformation operations.

The advantage of using a direct manipulation interface is that it reduces the distance between a user's intention and the capabilities provided by the system [52]. Hence, the efforts required to reach a goal is reduced. As an example, a user that wants to delete a column in a dataset, would probably want to click the column directly to find an applicable action. An interface that does not rely on direct manipulation, could require the user to specify the intended action in a domain specific language.

The direct manipulation approach could provide a solution that is easy to use, and reduce the learning curve for new users. Still, not all types of data operations will benefit from direct manipulation interfaces [52]. As an example, complex, or repeated, data cleaning and transformation sequences that require parameters may not be suitable. Future research initiatives should explore the use of direct manipulation interfaces in visual data profiling approaches, and determine which data cleaning and transformation processes that can be executed directly in a spreadsheet table view.

References

[1] J. M. Hellerstein, "Quantitative Data Cleaning for Large Databases," *United Nations Economic Commission for Europe (UNECE)*, Feb. 2008.

[2] S. Kandel, R. Parikh, A. Paepcke, J. M. Hellerstein, and J. Heer, "Profiler: Integrated Statistical Analysis and Visualization for Data Quality Assessment," in *Proceedings of the International Working Conference on Advanced Visual Interfaces*, New York, NY, USA, 2012, pp. 547–554.

[3] T. C. Redman, "Bad Data Costs the U.S. $3 Trillion Per Year," *Harvard Business Review*, 22-Sep-2016. [Online]. Available: https://hbr.org/2016/09/bad-data-costs-the-u-s-3-trillion-per-year. [Accessed: 18-Mar-2017].

[4] "CrowdFlower | 2016 Data Science Report." [Online]. Available: //visit.crowdflower.com/data-science-report. [Accessed: 19-Mar-2017].

[5] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011.

[6] J. Heer, J. M. Hellerstein, and S. Kandel, "Predictive Interaction for Data Transformation.," in *CIDR*, 2015.

[7] F. D. Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technology," *MIS quarterly*, pp. 319–340, 1989.

[8] C. M. Barnum, *Usability testing essentials: ready, set... test!* Elsevier, 2010.

[9] J. Nielsen, *Usability 101: Introduction to usability*. 2003.

[10] D. Roman *et al.*, "DataGraft: One-Stop-Shop for Open Data Management," Technical Report, January 2016. Available at http://www. semantic-web-journal. net/system/files/swj1285. pdf.

[11] W. Dai, I. Wardlaw, Y. Cui, K. Mehdi, Y. Li, and J. Long, "Data Profiling Technology of Data Governance Regarding Big Data: Review and Rethinking," in *Information Technology: New Generations*, Springer, Cham, 2016, pp. 439–450.

[12] J. W. Creswell, *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications, 2013.

[13] D. Sukhobok *et al.*, "Tabular Data Cleaning and Linked Data Generation with Grafterizer," in *International Semantic Web Conference*, 2016, pp. 134–139.

[14] V. Venkatesh, S. A. Brown, and H. Bala, "Bridging the qualitative-quantitative divide: Guidelines for conducting mixed methods research in information systems," *MIS Quarterly*, vol. Vol. 37 No. 1, pp. 21–54, Mar. 2013.

[15] I. Solheim and K. Stølen, *Technology Research Explained*. 2007.

[16] I. Sommerville, *Software Engineering*. Pearson, 2011.

[17] B. Hanington and B. Martin, *Universal methods of design: 100 ways to research complex problems, develop innovative ideas, and design effective solutions*. Rockport Publishers, 2012.

[18] “The Guide to Prototyping Process & Fidelity,” *Studio by UXPin*. [Online]. Available: https://www.uxpin.com/studio/ebooks/prototyping-process-fidelity-guide/. [Accessed: 13-Apr-2017].

[19] “The Ultimate Guide to Prototyping,” *Studio by UXPin*. [Online]. Available: https://www.uxpin.com/studio/ebooks/guide-to-prototyping/. [Accessed: 13-Apr-2017].

[20] J. Sauro and J. R. Lewis, *Quantifying the user experience: Practical statistics for user research*. Morgan Kaufmann, 2016.

[21] J. Nielsen, “Usability inspection methods,” in *Conference companion on Human factors in computing systems*, 1994, pp. 413–414.

[22] R. Spencer, “The streamlined cognitive walkthrough method, working around social constraints encountered in a software development company,” 2000, pp. 353–359.

[23] T. Mahatody, M. Sagar, and C. Kolski, “State of the art on the cognitive walkthrough method, its variants and evolutions,” *Intl. Journal of Human–Computer Interaction*, vol. 26, no. 8, pp. 741–785, 2010.

[24] “Cognitive Walkthrough | Usability Body of Knowledge.” [Online]. Available: http://www.usabilitybok.org/cognitive-walkthrough. [Accessed: 10-May-2017].

[25] S. Chen, “Six Core Data Wrangling Activities eBook,” *Trifacta*, 23-Nov-2015. .

[26] C. Batini and M. Scannapieco, *Data Quality: Concepts, Methodologies and Techniques (Data-Centric Systems and Applications)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[27] “ISO 9241-11:1998(en), Ergonomic requirements for office work with visual display terminals (VDTs) — Part 11: Guidance on usability.” [Online]. Available: https://www.iso.org/obp/ui/#iso:std:iso:9241:-11:ed-1:v1:en. [Accessed: 05-May-2017].

[28] “The Guide to Usability Testing - Free e-book by UXPin,” *Studio by UXPin*. [Online]. Available: https://www.uxpin.com/studio/ebooks/guide-to-usability-testing/. [Accessed: 13-Apr-2017].

[29] Z. Hussain, W. Slany, and A. Holzinger, “Current state of agile user-centered design: A survey,” in *Symposium of the Austrian HCI and Usability Engineering Group*, 2009, pp. 416–427.

[30] C. Stolte, D. Tang, and P. Hanrahan, “Polaris: a system for query, analysis, and visualization of multidimensional relational databases,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, no. 1, pp. 52–65, Jan. 2002.

[31] S. Kandel, A. Paepcke, J. Hellerstein, and J. Heer, “Wrangler: Interactive visual specification of data transformation scripts,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2011, pp. 3363–3372.

[32] J. Mackinlay, “Automating the Design of Graphical Presentations of Relational Information,” *ACM Trans. Graph.*, vol. 5, no. 2, pp. 110–141, Apr. 1986.

[33] J. Bertin, “Semiology of graphics: Diagrams, networks, maps, trans,” *WJ Berg. Madison, WI: The University of Wisconsin Press, Ltd*, 1983.

[34] W. S. Cleveland and R. McGill, “Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods,” *Journal of the American Statistical Association*, vol. 79, no. 387, pp. 531–554, Sep. 1984.

[35] M. O. Ward, G. Grinstein, and D. Keim, *Interactive data visualization: foundations, techniques, and applications*. CRC Press, 2015.

[36] S. Few, *Now You See It: Simple Visualization Techniques for Quantitative Analysis*, 1st ed. USA: Analytics Press, 2009.

[37] B. Mutlu, E. Veas, C. Trattner, and V. Sabol, “VizRec: A Two-Stage Recommender System for Personalized Visualizations,” in *Proceedings of the 20th International Conference on Intelligent User Interfaces Companion*, New York, NY, USA, 2015, pp. 49–52.

[38] M. Voigt, M. Franke, and K. Meissner, “Using expert and empirical knowledge for context-aware recommendation of visualization components,” *Int. J. Adv. Life Sci*, vol. 5, pp. 27–41, 2013.

[39] B. Mutlu, E. Veas, C. Trattner, and V. Sabol, “Towards a Recommender Engine for Personalized Visualizations,” in *International Conference on User Modeling, Adaptation, and Personalization*, 2015, pp. 169–182.

[40] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer, “Voyager: Exploratory Analysis via Faceted Browsing of Visualization Recommendations,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 649–658, Jan. 2016.

[41] “Vega-Lite.” [Online]. Available: https://vega.github.io/vega-lite/. [Accessed: 19-Mar-2017].

[42] L. Wilkinson, *The grammar of graphics*. Springer Science & Business Media, 2006.

[43] H. Wickham, *ggplot2: elegant graphics for data analysis*. Springer, 2016.

[44] J. Mackinlay, P. Hanrahan, and C. Stolte, “Show Me: Automatic Presentation for Visual Analysis,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1137–1144, Nov. 2007.

[45] A. Satyanarayan, R. Russell, J. Hoffswell, and J. Heer, “Reactive Vega: A Streaming Dataflow Architecture for Declarative Interactive Visualization,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 659–668, Jan. 2016.

[46] C. Little, “Forrester Wave for Data Preparation Tools,” Mar. 2017.

[47] A. Wilson, “A New Cloud-Based Data Preparation Solution from Google & Trifacta...,” *Trifacta*, 09-Mar-2017. .

[48] E. Bakke and D. R. Karger, “Expressive query construction through direct manipulation of nested relational results,” in *Proceedings of the 2016 International Conference on Management of Data*, 2016, pp. 1377–1392.

[49] B. Familiar, *Microservices, IoT and Azure: Leveraging DevOps and Microservice Architecture to deliver SaaS Solutions*. Apress, 2015.

[50] “Angular.” [Online]. Available: https://angular.io/docs/ts/latest/guide/architecture.html. [Accessed: 16-Apr-2017].

[51] “DataGraft | Documentation.” [Online]. Available: http://dapaas.github.io/documentation/. [Accessed: 17-Apr-2017].

[52] E. L. Hutchins, J. D. Hollan, and D. A. Norman, “Direct manipulation interfaces,” *Human–Computer Interaction*, vol. 1, no. 4, pp. 311–338, 1985.

Appendices

Appendix A

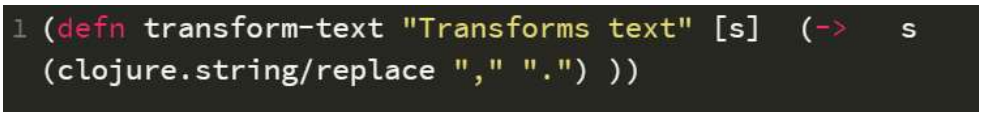
Statsbygg User Scenario

Make a dataset

1. **Function**: *set-first-row-header*
   Column(s): N/A
   Explanation: Set first row as header

MAPC – Map transformation to column(s) to modify values

2. **Function**: *transform-text*
   Column(s): BRUTTO_BTA_SUM
   Explanation: Replace ',' with '.'

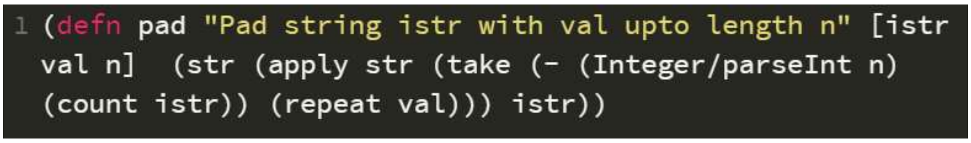


```
(defn transform-text "Transforms text" [s]  (->   s
(clojure.string/replace "," ".") ))
```

Figure 30: Example of user defined Clojure code in Grafterizer

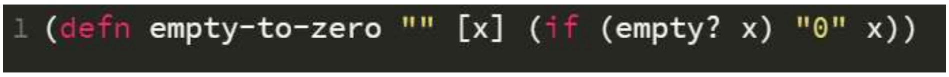
3. **Function**: *upper-case*
   Column(s): BYGGEIERFORHOLD
   Explanation: Convert ownership type to the unified form, I.e. upper case letters

4. **Function**: *pad*
   Column(s): KOMMUNE
   Explanation: Pad municipality code with 0 to 4 digits



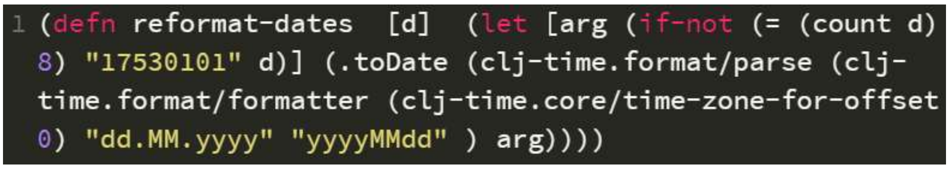
```
(defn pad "Pad string istr with val upto length n" [istr
val n]  (str (apply str (take (- (Integer/parseInt n)
(count istr)) (repeat val))) istr))
```

5. **Function**: *empty-to-zero*
   Column(s): GNR, BNR, FNR, SNR
   Explanation: Add zero to empty cells



```
(defn empty-to-zero "" [x] (if (empty? x) "0" x))
```

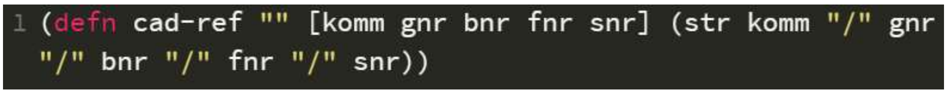
6. **Function**: *reformat-dates*
   Column(s): ERVERVAAR
   Explanation: Reformat dates to format 'dd.MM.yyyy'. If empty cell → set to 01.01.1753.
   Else → reformat according to specified format dd.MM.yyyy.



```
(defn reformat-dates  [d]  (let [arg (if-not (= (count d) 8) "17530101" d)] (.toDate (clj-time.format/parse (clj-time.format/formatter (clj-time.core/time-zone-for-offset 0) "dd.MM.yyyy" "yyyyMMdd" ) arg))))
```

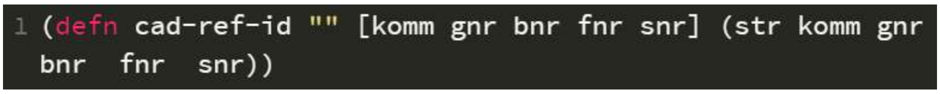
<u>Derive column – Create a new column by applying transformation to existing columns</u>

7. **Function**: *cad-ref*
   Column(s): KOMMUNE, GNR, BNR, FNR, SNR
   Explanation: Concatenate to string, separate with '/'.



```
(defn cad-ref "" [komm gnr bnr fnr snr] (str komm "/" gnr "/" bnr "/" fnr "/" snr))
```

8. **Function**: *cad-ref-id*
   Column(s): KOMMUNE, GNR, BNR, FNR, SNR
   Explanation: Concatenate to string.



```
(defn cad-ref-id "" [komm gnr bnr fnr snr] (str komm gnr bnr  fnr  snr))
```

Detailed Statsbygg User Scenario

For demonstration purposes of visual data profiling capabilities in the prototype, a pre-defined sequence of data cleaning and transformation steps are applied to the dataset. The numbering of each scenario step below corresponds to the numbering of functions specified above.

1. Click ‘Suggested Transformations’ tab, select transformation ‘Set first row as header’.

2. Select column ‘BRUTTO_BTA_SUM’, choose transformation ‘Replace (,) with (.)’ to replace all commas in number values with periods (e.g. ‘10770,4’ → ‘10770.4’).

3. Select column ‘BYGGEIERFORHOLD’, choose transformation ‘Set text to uppercase’ (e.g. ‘Eid’ → ‘EID’).

4. Select column ‘KOMMUNE’, choose transformation ‘Add (0) to the end of values, max length is (4)’ to add up to three zero(s) to any values that contains less than four digits (e.g. ‘214’ → ‘2140’).

5. Select columns ‘GNR’, ‘BNR’, ‘FNR’ and ‘SNR’, choose transformation ‘Set empty cells to value (0). This will check every cell for missing values and replace empty cells with a zero.

6. Select column ‘EIEFORHOLD’, choose transformation ‘Convert to standard format’. This is a custom function defined by Statsbygg that converts pre-defined values to a standardized format (e.g. ‘EID’ → ‘HJEMMELSHAVER’).

7. Select column ‘ERVERVAAR’, choose transformation ‘Reformat dates’. This is a custom function defined by Statsbygg that converts dates to a standardized format of type ‘dd.MM.yyyy’ (e.g. ‘18540101’ → ’01.01.1845’).

8. Select columns ‘KOMMUNE’, ‘GNR’, ‘BNR’, ‘FNR’ and ‘SNR’, choose transformation ‘Concatenate cells’. This is a custom function defined by Statsbygg that combines cell values of each row into a string that separates values by a ‘/’ (e.g. ‘2012/28/5/0/0’).

9. Select columns ‘KOMMUNE’, ‘GNR’, ‘BNR’, ‘FNR’ and ‘SNR’, choose transformation ‘Concatenate cells cad-ref-id’. This is a custom function defined by Statsbygg that combines cell values of each row into a string (e.g. ‘201228500’).

Appendix B

In terms of the streamlined cognitive walkthrough sessions, the two groups of expert reviewers went through user scenarios that are divided into tasks of the following format:

Task 1

I want to set first row as header.

Expert evaluation (questions answered by reviewers):

c. *Will the user know what to do next?*
d. *Will the user get appropriate feedback if the correct action is taken?*

Since all tasks include the same type of evaluation questions, the evaluation questions will be omitted in the scenarios below. The real scenarios conducted during walkthrough sessions include evaluation questions on each task.

The SINTEF HCI experts completed the following user scenario:

Scenario: Cleaning and transforming weather data

Task 1: I want to set first row as header.

Task 2: I want to delete the first column.

Task 3: I am now considering the 'weather' column only. I want to view:

- Total number of values
- Number of distinct values
- How many weather observations (1 row = 1 observation) there are on sunny days

Task 4: I am now considering the 'wind' column only. I want to replace all invalid cells, I.e. empty cells, with zero.

Task 5: I am once more considering the 'wind' column only. I want to replace all invalid cells, I.e. empty cells, with zero by using an alternative approach to the one used in task 4. Is that possible and how will the action be performed?

Task 6: I want to change the value in first row of column 'wind' to the value '6'.

Task 7: I am now considering the 'wind' column. I want to zoom in on the suspected outliers (I.e. red dots) in the box plot.

Task 8: I want to set all values in column 'weather' to uppercase letters.

Task 9: I want to undo all my actions and go back to the second step of my applied transformations.

The LogID domain experts completed the following user scenario:

Scenario: Cleaning and transforming real estate data: State of Estate

Task 1: I want to set first row as header.

Task 2: I want to delete the first column.

Task 3: I am now considering the 'BNR' column only. I want to view:

- Total number of values
- Number of distinct values
- The number of real estate properties (1 row = 1 property)

Task 4: I am now considering the 'FNR' column only. I want to replace all invalid cells, I.e. empty cells, with zero.

Task 5: I want to change the value in first row of column 'SNR' to the value '6'.

Task 6: I am now considering the 'BNR' column. I want to assess possible outliers. What is the value of the most extreme outlier?