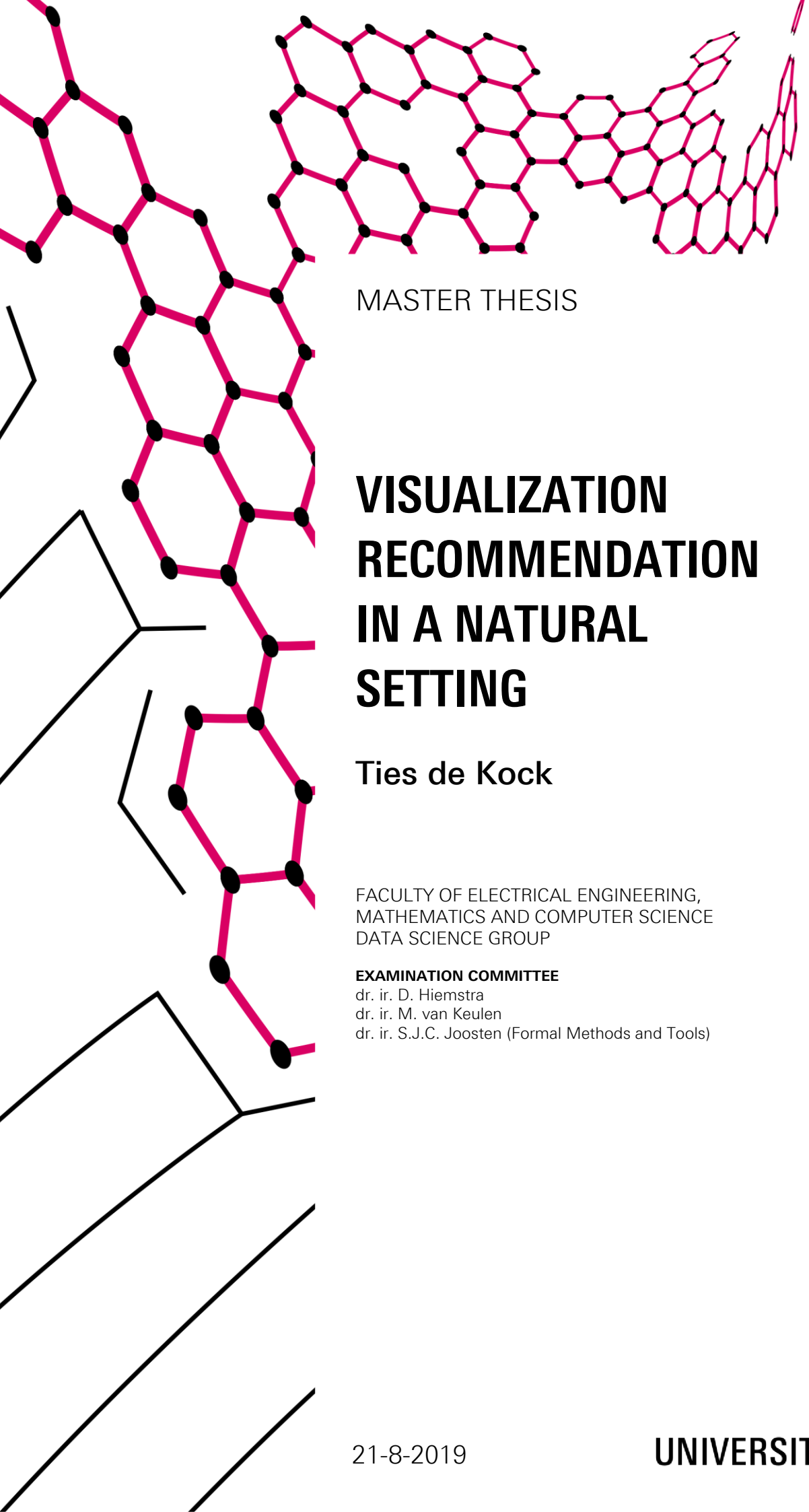


MASTER THESIS

VISUALIZATION RECOMMENDATION IN A NATURAL SETTING

Ties de Kock

FACULTY OF ELECTRICAL ENGINEERING,
MATHEMATICS AND COMPUTER SCIENCE
DATA SCIENCE GROUP

EXAMINATION COMMITTEE
dr. ir. D. Hiemstra
dr. ir. M. van Keulen
dr. ir. S.J.C. Joosten (Formal Methods and Tools)

21-8-2019

UNIVERSITY OF TWENTE.

ABSTRACT

Data visualization is often the first step in data analysis. However, creating visualizations is hard: it depends on both knowledge about the data and design knowledge. While more and more data is becoming available, appropriate visualizations are needed to explore this data and extract information. Knowledge of design guidelines is needed to create useful visualizations, that are easy to understand and communicate information effectively.

Visualization recommendation systems support an analyst in choosing an appropriate visualization by providing visualizations, generated from design guidelines implemented as (design) rules. Finding these visualizations is a non-convex optimization problem where design rules are often mutually exclusive: For example, on a scatter plot, the axes can often be swapped; however, it is common to have time on the x-axis.

We propose a system where design rules are implemented as hard criteria and heuristics encoded as soft criteria that do not need to be satisfied, that guide the system toward effective chart designs. We implement this approach in a visualization recommendation system named OVERLOOK, modeled as an optimization problem implemented with the *Z3* Satisfiability Modulo Theories solver. Solving this multi-objective optimization problem results in a Pareto front of visualizations balancing heuristics, of which the top results were evaluated in a user study using an evaluation scale for the quality of visualizations as well as the low-level component tasks for which they can be used. In evaluation, we did not find a difference in performance between OVERLOOK and a baseline of manually created visualizations for the same datasets.

We demonstrated OVERLOOK, a system that creates visualization prototypes based on formal rules and ranks them using the scores from both hard- and soft criteria. The visualizations from OVERLOOK were evaluated in a user study for quality. We demonstrate that the system can be used in a realistic setting. The results lead to future work on learning weights for partial scores, given a low-level component task, based on the human quality annotations for generated visualizations.

Acknowledgements

This thesis presents the work I have done on data visualization at the Data Science group at the University of Twente. Working on data visualization was an unexpectedly challenging process, which only made me learn more during the project.

First and foremost, I would like to thank my supervisors, Djoerd Hiemstra, Maurice van Keulen, and Sebastiaan Joosten, for their excellent feedback and tough questions. Specifically: Maurice, for your focus on the value of and presentation of results. Sebastiaan, for your attention to detail and support in the last months. And especially Djoerd, for your continued guidance and support, which went beyond what I could reasonably expect from a supervisor. And I want to mention the other staff members who supported me as well. Whether it was through actual support or by helping me sharpen my ideas every time I explained them.

I would also like to thank the participants in the user study for their time and valuable feedback. You made a potentially frustrating process fun, and your contribution was critical to the completion of my work. This made me realize once more — as a computer scientist — that user testing is an essential form of soliciting feedback. When it comes to feedback, I need to mention my peers working on a thesis, for your feedback and for helping me reflect.

Furthermore, my appreciations go out to my friends and colleagues for your support, discussions, and distraction during this process. And most importantly, my parents and sister, for their infinite support during the entirety of my education.

— Ties de Kock

API Application Programming Interface. 1, 2

APT A Presentation Tool. 7–9, 11

ASP Answer Set Programming. 11, 46

ASQ
After-Scenario-Questionnaire. 26

CBS Statistics Netherlands. 1, 3, 5, 6, 18, 35, 44

DBMS
database management system. 11

DCG
Discounted Cumulative Gain. 45

EOU
ease of understanding. ix, 32, 33, 37–40, 44

GIS geographic information system. 28, 36

HCI Human-Computer Interaction. 25

IIR Interactive Information Retrieval. 23, 25, 46

IR Information Retrieval. 23–25, 35, 37, 39, 44–46

ISO International Organization for Standardization. 24

JSON
JavaScript Object Notation. 16

LTR Learning to Rank. 11, 46, 47

NDCG
normalized Discounted Cumulative Gain. 45

ODATA
Open Data Protocol. 1, 2

REST
Representational State Transfer. 1

SAGE
a System for Automatic and Graphical Explanation. 8, 27

SMEQ
Subjective Mental Effort Question. 26, 27

SMT
: Satisfiability Modulo Theories. iii, 17–21, 43, 46

SQL Structured Query Language. 8, 12

StatLine
: CBS StatLine. x, 1, 2, 5, 24, 31, 33, 35, 38–41, 43–45, 47

SUS System Usability Scale. 26, 27

TREC
: Text REtrieval Conference. 45

UME
: Usability Magnitude Estimation. 26, 27

VizRec
: Visualization Recommendation. 7, 15, 16, 18, 19, 36, 43, 46, 47

The amount of digital data that is being created and is available for analysis is increasing. Today, more information is available than ever before. This co-occurred with both an increase in usage of advanced data analysis methods and the democratization of data science. At the same time, exploring information becomes increasingly difficult as the volume of data increases [HS12].

While data processing is automated; reasoning, applying domain knowledge, and interpreting the data is performed by humans. Visualizing data is an important step, both during exploratory data analysis as well as when presenting results. Being able to create useful visualizations, that are relevant and give new insights has become a must-have skill for data analysts.

Analysts use visualizations to explore data, spot trends, etc. Creating visualizations is a mostly manual process, where choices need to be specified by analysts. The resulting visualizations are used by decision-makers in corporations, government, etc. By extension, one could argue that data visualization is an essential skill for the members of the general public or even society at large.

The choices made when designing a visualization depends on multiple variables, including the dataset, selected facts, selected data, type of visualization, and the task at hand. The utility of the resulting visualization depends on its relevance to the task at hand and whether it gives new insights.

While there are many methods for creating visualizations, this thesis will focus on visualizations specified in high-level languages that concisely describe a visualization. These specifications describe how the visualization encodes data, without offering fine-grained control over details.

Creating visualizations is usually a manual process instead of a process where a system recommends visualizations. Wongsuphasawat et al. [Won+16a] group systems that recommend visualizations on two orthogonal axes of recommendations they provide: recommending the data that is queried and recommending visual encodings.

1.1 SETTING

This thesis focuses on *encoding recommendation*, where specifications for visualizations are recommended based on a high-level description of the visualization (what type of visualization, what fields to use) and dataset (meta-)data[1].

These visualizations are created in the setting of statistical data which is available in a data warehouse containing tables with several facts and dimensions, and where for each dataset an example selection (chosen *facts*, *dimensions*, and *filters*) is available.

In this setting, we assume that the data warehouse is accessed through a REST API. This design is common for (open) data sources and implies that retrieving data has a high latency. The meta-data does not change often and can be cached. However, due to the latency of retrieving data, the data for a visualization can not be retrieved while recommending visualizations.

CBS StatLine. One source of such information is Statistics Netherlands (CBS), which provides access to their statistical datasets stored in a data warehouse as open data accessible via a REST API. This API also provides the data for CBS StatLine (StatLine), the Statistics Netherlands (CBS) website for viewing statistical information. The meta-data for both the facts and dimensions of datasets in the data warehouse is machine-readable, with values for dimensions taken from standardized taxonomies. The API is implemented following the Open Data Protocol (OData) standard, which defines both how the data is described (meta-data), as well as how data can be projected and selected.

[1] E.g., "A bar chart containing the *year*, *industry*, and *expected revenue* fields from the dataset *81238ned*

```
https://opendata.cbs.nl/ODataApi/odata/81238ned/TypedDataSet
    ?$select=BedrijfstakkenBranchesSBI2008,Perioden,RegioS,SaldoOmzetKomende3Maanden_26,\
             SaldoVerkoopprijzenTKomende3Mnd_31,SaldoInkoopOrdersKomende3Mnd_41,\
             SaldoPersoneelssterkteKomende3Mnd_80,SaldoEconomischKlimaatKomende3Mnd_105&
    $filter=(
        (BedrijfstakkenBranchesSBI2008 eq "300016") or
        (BedrijfstakkenBranchesSBI2008 eq "307500") or
        (BedrijfstakkenBranchesSBI2008 eq "800037")
    ) and (
        (Perioden eq "2019MM02") or
        (Perioden eq "2019MM03") or
        (Perioden eq "2019MM04") or
        (Perioden eq "2019MM05")
    ) and ((RegioS eq "NL01"))
```

Figure 1.1: CBS StatLine default query for table *81238ned*.

Queries used by StatLine use a subset of OData to select data, almost exclusively using queries in conjunctive normal form. Literals in the queries consist of comparisons and the usage of substring operators. The example query shown in Figure 1.1 selects five facts and three dimensions, and filters the rows by selecting only specific values for the dimensions.

1.2 PROBLEM

Encoding recommendation can be viewed as the process of enumerating and ranking candidate visualizations from the space of possible visualizations. Design knowledge is commonly incorporated in the design by generating candidate visualizations using *expressiveness constraints* that express visualization limitations and by ranking by *effectiveness constraints* based on models of visual encoding effectiveness [Won+16a, p.2].

This is an abstract approach, but simpler models, such as creating visualizations based on templates, are limited since the suitable visualization depends on the data. For example, while "a bar chart with all years on the x-axis" seems sensible, when the data only contains one year this yields a chart with one bar, which is generally seen as ineffective.

Implementations of encoding recommendation systems commonly generate visualizations using the effectiveness- and expressiveness constraints as "ground truth" rules created by experts, grounded in perception research [Won+16a, p.3]. Implementing a system that balances and optimizes these rules is complex: because of implementation complexity, when implemented using a generate and test approach, prior approaches often had to compromise the implementation of effectiveness constraints [Mor+19, p.7].

1.3 RESEARCH GOALS

When applying an encoding recommendation system in a practical setting with information from (open) data sources queried through APIs as inputs, multiple problems arise. The first question encountered is that of information available to the recommendation system, since (open) data sources do typically have meta-data, but access to the information is relatively slow. This setting was introduced earlier in this chapter. This thesis uses StatLine open data as a data source, which allows us to compare our generated visualizations to a baseline of visualizations from StatLine in evaluation.

The primary objective of this study is to investigate how an implementation of an encoding recommendation system performs in this setting. The resulting product needs to be evaluated, accounting for the different use cases and variations of inputs and datasets encountered.

The first research question investigates the literature on automated visualization systems and leads to an overview of the start of the art, as well as a summary of design choices in implementations.

RQ1: What models are used in the implementation of visualization recommendation systems?

After the state of the art is known, a new system is designed which accounts for issues described in the literature review as well as constraints implied by using a real-world situation, with partial visualization prototypes, as input to the system.

RQ2: How can an encoding recommendation system be implemented in order to account for design variation and soft heuristics?

This results in the design and implementation of OVERLOOK, a visualization recommendation system that finds relevant visualizations from a description of the dataset and selected data while adhering to the constraints of the setting.

Afterward, the system is evaluated on its value for users in a user study. To the knowledge of the author, there is no standard evaluation methodology that is applicable for systems that generate sets of visualizations based on a user query. This thesis views this setting as being on the intersection of information visualization and (interactive) information retrieval. This lead to the following two research questions:

RQ3: How can the value to users, for visualizations from a set of visualizations for a given query on a visualization recommendation system, be evaluated?

RQ4: How do the results of OVERLOOK perform compared to the baseline visualizations by CBS?

In aggregate, these questions allow us to answer the main question of whether OVERLOOK provides good visualization support [for users] in a realistic setting. Besides answering the main question, three artifacts are delivered: (i) an implementation of an automated encoding recommendation system, (ii) an evaluation methodology for assessing sets of visualizations for a query, and (iii) a set of annotated charts that can be used in future work (e.g., learning to rank).

1.4 THESIS OUTLINE

The overall structure of this thesis takes the form of six chapters, including this introductory chapter. Chapter 2 begins by presenting the setting of this thesis and laying out related work on models of visualization and visualization recommendation systems. The related work leads to the design of OVERLOOK, presented in Chapter 3. The fourth chapter is concerned with the design of the evaluation materials used in this thesis, which are then first evaluated and validated in Chapter 5. Chapter 6 details the design of the user study and analyzes the results. Finally, the conclusion gives a summary and critique of the findings.

This chapter describes and discusses the methods used in visualization recommendation. The first section introduces the available information for the visualization recommendation algorithm. The next section will provide an overview of the related work on models of visualization (Section 2.2) and visualization recommendation systems (Section 2.3). The final section moves on to describe the concerns and implementation choices shared by the discussed visualization recommendation systems.

2.1 AVAILABLE INFORMATION

This chapter assumes that the dataset to visualize is already selected, either by a user or by another part of a system. The meta-data for the dataset is available, but data (and collection statistics) are not available to this system without performing a call to the data source. The generation of visualizations is performed offline without communicating with data sources. The included data sources in the prototype are CBS and third party datasets hosted by CBS. Looking up data on data sources is expensive[1]; the system can not query the data source while recommending visualizations. This implies that (selection specific) summary statistics are not available.

To decouple the prototype from StatLine, the data source specific meta-data is transformed into an abstract model that is independent of the data source[2]. This model is based on the type of queries supported by data warehouses, with *dimensions* (fields that data are grouped by) and *facts* (fields that contain independent variables) in a star-schema. Figure 2.1 shows the schema for an example dataset containing three dimensions (each with hierarchical levels) and several facts.

Dimensions. The cardinality of dimensions is known and can be restricted by the query if it selects specific values. The type of measurement of a dimension or topic (quantitative, ordinal, nominal) and its specific type (e.g., date, percentage, geographic location) are known.

Facts. For facts (i.e., quantitative fields), the cardinality of selected data is not available during visualization recommendation. The type and unit of values are known.

Visualization meta-data. The data source provides meta-data for visualization. However, the meta-data is not re-usable for our purpose. First of all, in some situations, the current application changes the data that is selected. This can cause semantic differences in the chart (e.g., a bar chart comparing ten regions gets reduced to one bar for the selected region, which is not a sensible visualization).

In addition, there are some practical considerations for the decision to build a new meta-model. Documentation for the provided meta-data is not available, and some unspecified heuristics are used

[1] Queries on data sources are slow because of round trip latency, time taken to perform the query, and time to parse the data after it has been retrieved.

[2] And has been applied to other data sources in earlier iterations.

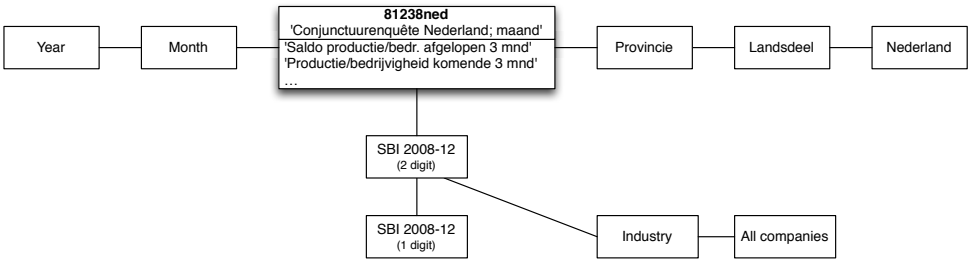


Figure 2.1: *Snowflake schema* of CBS StatLine table *81238ned*.

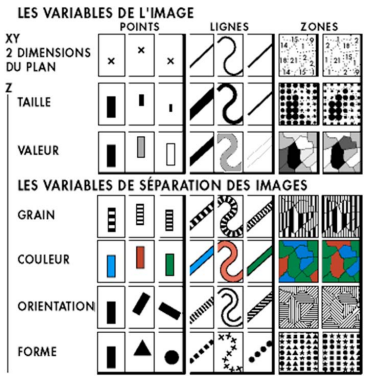


Figure 2.2: Visual variables by Bertin, from [Ber83].

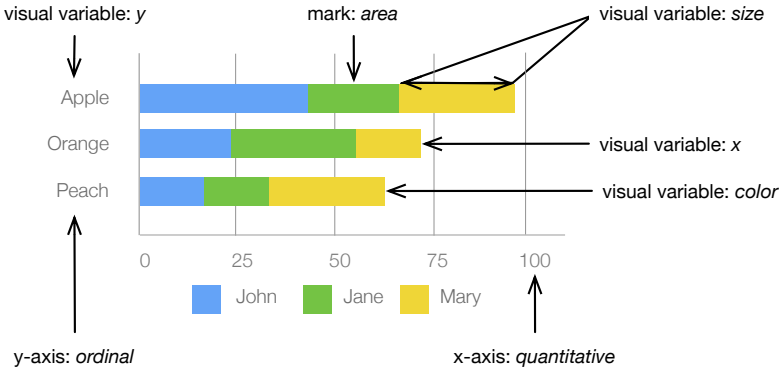


Figure 2.3: Example visualization showing the elements of a graphic annotated using the model of Bertin.

by CBS when creating a visualization[3]. These heuristics are a black box and reverse engineering them caused the first version of the prototype to break when data was updated or when CBS made different choices when annotating new or updated datasets.

Furthermore, multiple concerns (e.g., selecting a supported chart type, merging filters into query, picking fields for an axis) were scattered throughout the code, leading to an implementation that was hard to maintain. That motivated the decision to (a) create a custom meta-model and (b) using a more formal method for recommending (relevant) visualizations. Later in this chapter, Section 2.3 will provide an overview of work on visualization recommendation. Afterward, the next chapter describes the data model used by OVERLOOK and how it implements visualization recommendation.

2.2 MODELS OF VISUALIZATION

Previous research has established an abstract model of graphics. One well-known early study that is often cited in research on information graphics is that of Bertin [Ber83]. It identified three major properties of information graphics: (1) The classification of variables by their type of measurement, (2) the classification of designs by the types plotted on their axes, and (3) the concept of visual variables as properties of marks.

Bertin's model uses the typology by Stevens [Ste46] to classify the scale of measurement of variables as being either *Quantitative*, elements with a constant numerical difference, e.g., integers; *Ordinal*, elements with a natural sequence, e.g., age groups; or *Nominal*, which consists of elements with no inherent order, e.g., gender. The type of variables on axes is used to categorize designs; for example, a Quantitative-Quantitative plot is commonly referred to as a scatter-plot. Elements on a plot (points, lines) were named *marks*. Finally, Bertin defined seven *visual variables* that modify (the appearance of) marks: *position*, *size*, *shape*, *value*, *color*, *orientation*, and *texture*. Figure 2.2 provides an overview of these visual variables.

The model of Bertin provides a vocabulary to describe the visual design of information graphics. Figure 2.2 shows a horizontal bar chart of which the elements have been annotated using this model. The horizontal bar chart uses the visual variables *x*, *y*, *size*, and *mark*. The graphic has *area mark*s and displays a quantitative variable on the *x*-axis and an ordinal (alphabetically sorted) variable on the *y*-axis. Note that this makes it a horizontal bar chart and that all of the (sub)bars (i.e., marks), are using multiple visual variables. Each person is identified by a *color*, the value of a mark is shown by its *size*, and the *x* position is defined by the sum of the values.

Wilkinson was apparently the first to use the term *grammar of graphics*, and view graphics as sentences in a language. The term *grammar* refers to the relationship between components of graphics (instead of the words, the elements). Graphics are specified in a formal language, assembled, and finally displayed [Wil05].

[3] E.g., "Topics on the *x*-axis", "Time on *x*-axis", "Prefer Time over Topics on *x*-axis, "use grouped bars for Topics".

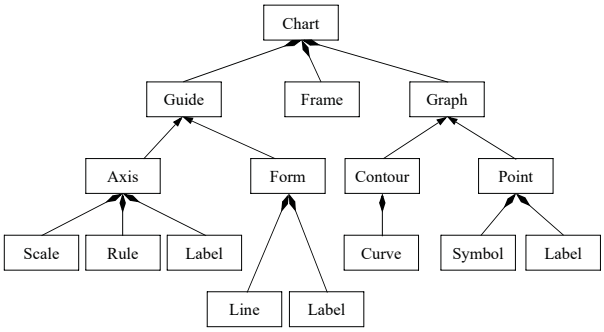


Figure 2.4: "Design tree" for a chart, from [Wic10, p.10].

Group	Visual variables
Marks	Points, lines, areas
Positional	1-D, 2-D, 3-D
Temporal	Animation
Retinal	Color, shape, size, saturation, texture, and orientation

Table 2.1: Bertin's graphical objects and graphical relationships, as reproduced by Mackinlay [Ber83; Mac86].

The specification uses the following elements to declare graphics: *data*, variable *transformations*, *scale* transformations, a *coordinate system*, *element*s (e.g., points) and their aesthetics, and finally *guides* (axes, legends). The components are combined in a hierarchical fashion, as shown in Figure 2.4.

This model of graphics is the basis used in common applications. For example, the *ggplot2* library in *R* implements an algebra based on the grammar of graphics [Wic10], and *Vega-Lite* [Sat+17] is a JavaScript implementation of a grammar of graphics that adds extensions for interaction and uses rules to select "smart defaults" for unspecified values (e.g., the colors of a color scale, font size of labels).

2.3 VISUALIZATION RECOMMENDATION SYSTEMS

Vartak et al. propose the class of Visualization Recommendation (VizRec) as systems that allow users to easily traverse the space of visualizations and focus on the ones most relevant to a task. The recommendations (potentially) include both relevant data and relevant visualizations, with criteria for relevance that include classic *relevance*, for a user given a task; *surprise*, which considers the novelty of a recommendation; and *non-obviousness* which considers whether the recommendation provides new information for a domain expert [Var+17]. Most current systems focus exclusively on either recommendation of data to be queried or of visual encodings. This section will first introduce the history of, and current systems for the recommendation of visual encodings, followed by the introduction of several data (query) recommendation systems.

2.3.1 Visual encoding recommendation systems

Visualization recommendation was first demonstrated by Mackinlay [Mac86]. In his seminal study, Mackinlay reports on the design of an automated system, *A Presentation Tool* (APT), for the presentation of relational information (charts). APT used Bertin's vocabulary of visual variables (Table 2.1).

APT used the effectiveness of specific visual variable for each type of measurement to define an order. The order is based on an extension of the ranking of the accuracy with which users can perform quantitative perceptual tasks by Cleveland and McGill [CM84]. Furthermore, it restricted visual variables to specific scales of measurement, since a visual encoding (e.g., size) may imply an ordering to users that

does not exist in the data. Finally, it introduced *expressiveness criteria*, that ensure that a design can express the given information and *effectiveness criteria* for retinal variables, which determine if a design matches the constraints of the human visual and cognitive system.

APT was developed on a Symbolics LISP Machine using logic programming, with 200 rules that express the expressiveness- and effectiveness criteria. Depth-first-search with backtracking could be used because the effectiveness criteria defined a total ordering over designs. However, Mackinlay notes that this is unlikely to hold when the theory of effectiveness, and transitively the effectiveness criteria become more sophisticated.

Unlike APT, which synthesizes designs from logical rules, S. F. Roth et al. propose *a System for Automatic and Graphical Explanation* (SAGE) that that matches data to visualization prototypes. It uses a library of design prototypes that are then customized for visualization. Compared to APT, SAGE uses a richer representation of the characteristics of data, including scales of measurement, the frame of measurement (quantitative/valuation, coordinate), and complex types (e.g., interval) [Rot+94].

The model of a grammar of graphics was extended in *Polaris* [STH02] where a visual specification is used to display data in relational databases. The specification defines a table with *row*, *column*, and *layer* dimensions where each table entry (cell) is a graphic. The tables are shown as *small multiple* displays, which is the term Tufte [Tuf01] used to refer to a design where each cell in a table contains the same graphical design; viewers only need to understand the design of a single cell to understand the design of all cells.

In Polaris, a visual specification consists of a specification of the data selection, the type of mark used in each cell, and the details of visual encodings. The data selection is defined with a relational algebra that is transformed into SQL.

Polaris evolved into *Tableau*, which distinguishes different *roles* for how fields are used in a graphic (i.e., as a dimension, as an attribute). In Tableau, a graphic is a selection of categorical (subtypes: *data*, *discrete values*, *dimensions*) and quantitative (subtypes: *continuous*, *dependent*, *independent*, *independent: date*) fields from a dataset, mapped to *rows*, *columns*, or *properties* of *marks*. It then defines what chart types are possible given the number and type of selected fields, and defines a default order (indicating a preference) for types of charts.

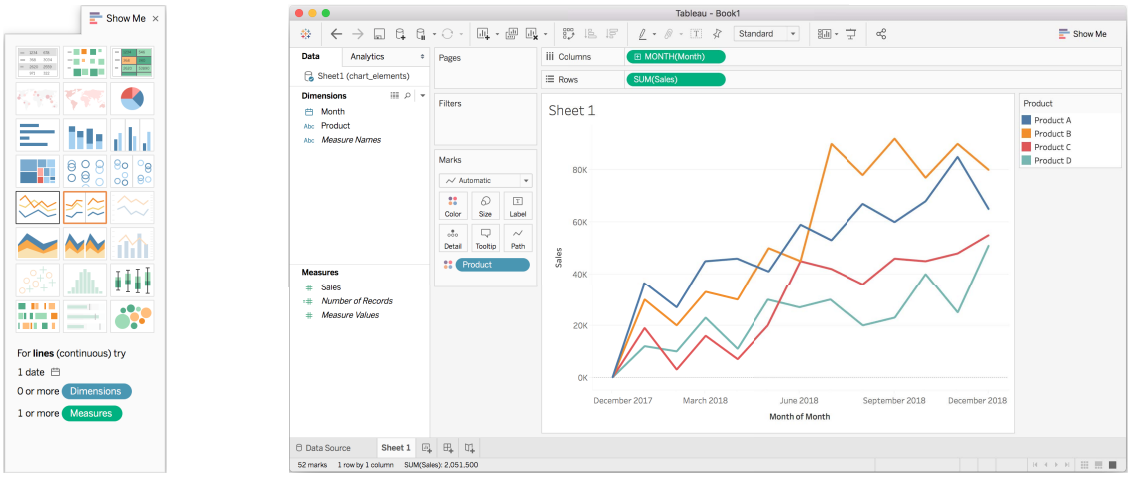
Show me (Figure 2.5(a)) is a user interface element that shows what charts are possible (i.e., "Two quantitative fields can create a scatter plot, bar chart, . . . "). Finally, a graphic is assembled by the user (selection from possible views) or proposed by the system [STH02]. In the user interface (Figure 2.5(b)) fields are grouped by type of measurement as being *dimensions* (categorical) or *measures* (quantitative). Fields are dragged to "shelves" that map to visual variables (of Bertin). The type of chart implies the mark type, the *columns* shelve maps to the x-axis[4], *rows* shelve maps to the y-axis, and the *marks* shelve to retinal properties.

A broader perspective has been adopted by Elzen, Elzen, and Wijk [EEW13] who argue that users do not have an overview of the space of information contained in a dataset and of possible visualizations, and propose a system that *guides* users in the visual data exploration process[5]. When adjusting parameters the systems displays (Figure 2.6) a large view of the current visualization (large single) and small multiples for each value of the parameter being adjusted, and keeps a history of changes to enable users to undo these easily. Participants in a user study preferred the system and explored a larger area of the space of visualizations compared to a baseline system (without small multiples or history).

The view that data exploration is important is shared by Wongsuphasawat et al. [Won+16b], who draw on earlier work on automated presentation and argue that *data variation* (seeing different variable selections

[4]When more axes are selected than is possible for the type of chart, multiple rows or columns of charts are created by faceting on the additional field.

[5]In the model of Van Wijk [Van05]: the activity of gaining knowledge while exploring data by creating visualizations.



(a) *Show me* **(b)** Shelves and Marks

Figure 2.5: The *Tableau* user interface.

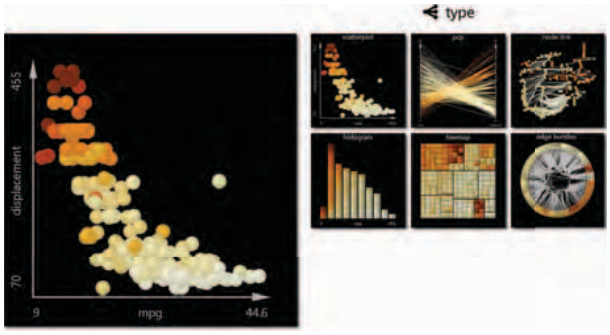


Figure 2.6: [Small multiple] mapping split by visualization type, Figure 3e from [EEW13, p.195].

and encoding) is more important than *design variation* (different visual encodings of the same data). There is a combinatorial explosion of possible design variations. *Voyager* is designed as a *mixed-initiative system*[6] where the system recommends charts by suggesting variables and encodings. Charts are rendered using *Vega-Lite*.

As in APT, the permitted mark types and encoding channels are based on the type of data. However, compared to APT, the model is extended by using the same typology as used by Tableau. In addition, because humans can only easily discriminate a limited number of different *colors*, *shapes*, *rows*, or *columns* at once, the cardinality of a field is taken into account when evaluating permitted encodings, creating more complicated expressiveness constraints. These rules are reproduced in Figure 2.8.

The architecture of the recommendation system (Figure 2.7), named *Compass* implements recommendation as a series of sequential, independent steps: (1) variable selection, (2) data transformation, (3) encoding design, and (4) clustering and ranking. Encodings are scored with a weighted sum of the effectiveness score of features, with manually tuned weights.

In contrast to Voyager (which recommends variables), *Voyager 2* supports exploration steered by the user by augmenting manual specifications, with the main view matching the input and small views for (multiple) alternative encodings. The user specifies what part of the specification is filled in by the system, and the system presents a ranked collection of graphics as output. As an example, a user can specify that each country is plotted on the x-axis and "all other variables" on the y-axis (Figure 2.9). Compared to Voyager, users have more control over the results.

[6] A system that contains agent(s) that provide automation based on guesses of user intent [Hor99].

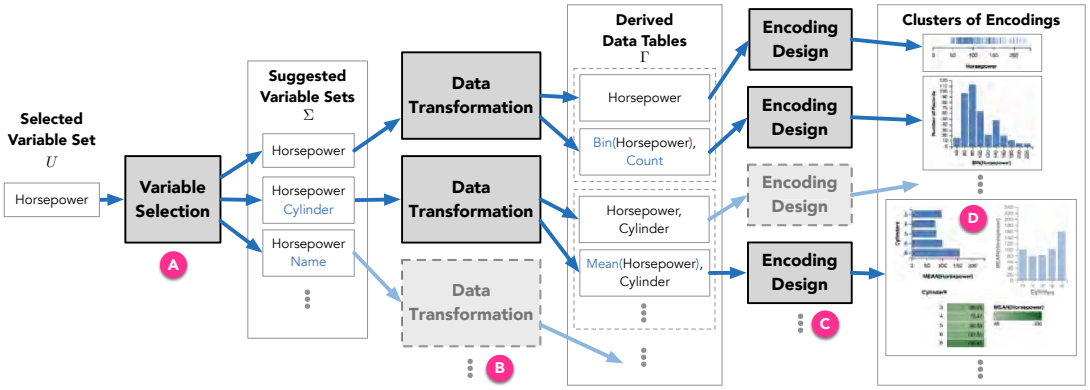


Figure 2.7: Compass's 3-phase recommendation engine, from [Won+16b, p.6].

Data Types	Encoding Channels
quantitative, temporal	x,y > size > color > text
ordinal	x,y > column, row > color > size
nominal	x,y > column, row > color > shape

Table 1. Permitted encoding channels for each data type in Compass, ordered by perceptual effectiveness rankings.

Mark Types	Required Channels	Supported Channels						
		X, Y	Column, Row	Color	Shape	Size	Detail	Text
point	x or y	✓	✓	✓	✓	✓	✓	
tick	x or y	✓	✓	✓				
bar	x or y	✓	✓	✓				
line, area	x and y	✓	✓	✓			✓	
text	text and (row or column)		✓	✓				✓

Table 2. Required and permitted encoding channels by mark type.

Data Types	Mark Types
Q	tick > point > text
(O **or** N) × (O **or** N)	point > text
$Q \times N$	bar > point > text
$Q \times (T$ **or** $O)$	line > bar > point > text
$Q \times Q$	point > text

Table 3. Permitted mark types based on the data types of the x and y channels. N, O, T, Q denote nominal, ordinal, temporal and quantitative types, respectively.

Positions	x, y
Facets	column, row
Level of detail	color (hue), shape, detail
Retinal measures	color (luminance), size

Table 4. Encoding channel groups used to perform clustering.

Figure 2.8: Required and permitted mark and encoding types, from [Won+16b, p.7].

Figure 5. Mapping a *quantitative field wildcard* to *x* and *origin* to *y* (A) produces a gallery of plots. A *wildcard function* enumerates no function (*none*) and *mean* (B-C), generating strip plots of raw values and bar charts of mean values (D). The ? in (A) denotes the wildcard function.

Figure 2.9: Wildcards in *Voyager 2*, from [Won+17, p.4].

The recommendation system implements the *CompassQL* [Won+16a] query language using a derivation of the recommendation engine used in Voyager. The enumeration of chart specifications that adhere to all criteria is implemented with a backtracking algorithm. These are then ranked based on the order specified by the query, with manually tuned weighing factors used when ranking by effectiveness.

A recent study[7] by Moritz et al. [Mor+19] introduced *Draco*, bringing together techniques from logic programming and information visualization. In this innovative study, Moritz et al. point out that prior approaches often had to compromise the implementation of effectiveness criteria due to implementation complexity, and argue that implementing visualization recommendation using logic programming allows designers to focus on describing the design space of visualizations and visualization preferences instead of on re-implementing search algorithms that are available through domain-independent constraint

[7] Published after the design and implementation for this thesis had finished.

solvers [Mor+19, p.8].

The constraint programming problem was implemented as an Answer Set Programming (ASP) program using *Vega-Lite* for visualization specifications. In ASP programs, rules have the form of $A : -L_1, \ldots, L_n$, consisting of a head (A), followed by a body ($L_1, \ldots, L_n$). A rule is true if its body is true. Rules can either define atoms; be integrity constraints; or be soft constraints, which have a weight/cost when they are violated. The cost of a result is the sum of all soft constraint violations multiplied by the count of their violations. The generated ASP program contains rules describing the visualization as well as (optional) rules indicting fields of interest and task. Several base rules are added to implement expressiveness criteria. Solving the ASP program finds solutions that adhere to the constraints and have a minimal cost.

Moritz et al. show that ASP programs can re-create the results of APT (without using soft constraints) and Voyager 2 (with manually tuned weights). Given the difficulty of manually tuning these weights, the authors propose that Learning to Rank (LTR) (linear regression on pairs of soft constraint violation counts) can be used to learn these weights. User preferences between pairs of visualizations from results of graphical perception experiments were re-used as training data. Moritz et al. demonstrate that a system trained on a subset of the annotations from [KH18] and a small subset of [SED18][8] correctly orders 93 % of pairs on the test set held out from training.

Draco demonstrates that multiple encoding recommendation systems can be implemented as ASP programs and that (effectiveness) scores can be learned by re-using results of graphical perception experiments. Moritz et al. propose that future work could re-rank visualizations using low-level features, use multi-objective (Pareto) optimization to enumerate the frontier of designs that make a trade-off or add a richer task taxonomy to capture latent information (i.e., the task is in the user's mind).

2.3.2 Data (query) recommendation systems

Contrary to the studies discussed in the previous section, which have a background in information visualization, finding a relevant visualization has also been approached from the perspective of data management and/or database research.

A premise is that the design space of possible visualizations for a dataset is too large, but that an analyst needs to explore the relevant area in order to extract relevant information from data [EEW13; Won+17; Var+15; Sid+16]. The following systems were designed to support this process.

In [Var+15] Vartak et al. present *SeeDB*, a system that finds the visualizations of a dataset with the highest utility. Data is retrieved from a generic DBMS using select-project-join queries on a snowflake schema. The utility is defined as the deviation from a reference, defaulting to creating a normalized histogram of selected data and using earth mover's distance as a metric. Both the metric and the reference query are specified by the user.

Comparing all selections of data is computationally expensive; therefore, multiple optimizations are used. Data is processed in partitions. After each partition, candidates visualizations for which the upper bound of the confidence interval of the expected utility is outside the top K are pruned. In addition, by using a multi-armed-bandit approach, candidates that are very likely in the top K are kept without additional computation.

The prototype was evaluated in a user study with a within-subject design (2×2 visualization tool $\times$ dataset) using a think-aloud protocol. Participants had prior data analysis experience. During the experiment, participants answered a survey per task. Afterward, they participated in an exit interview.

Another approach is used by *zenvisage* [Sid+16], which instead searches data for visualizations with a desired pattern. Zenvisage uses a model that views visualizations as being defined by the following five

[8] Selecting only the *value* and *summary* tasks from Saket, Endert, and C. Demiralp [SED18].

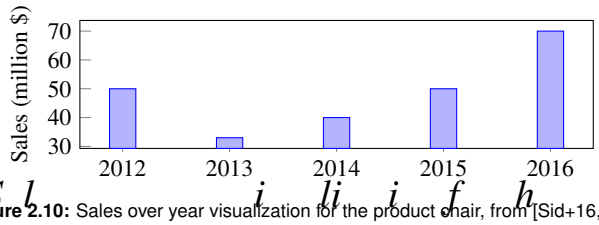


Figure 2.10: Sales over year visualization for the product chair, from [Sid+16, p.2].

Name	X	Y	Z	Viz
*f1	'year'	'sales'	'product'.'chair'	bar.(y=agg('sum'))

Table 2.2: Query for the bar chart of sales over year for the product chair, from [Sid+16, p. 3].

components: x-axis attribute, y-axis attribute, subset of data used, type of visualization (e.g., bar chart, scatter plot), and binning and aggregation functions used.

Visualizations are queried by a query in *ZQL*, that binds these components of visualizations to part of the queries. A query selects axes (x, y), data (z), and visual properties (Viz). A ZQL query and its resulting visualization are shown in Table 2.2 and Figure 2.10.

Using ZQL, it is possible to perform queries that specify collections of visualizations and perform operations on these collections. ZQL supports wild-cards ("evaluate every column for the z-axis"), and queries can depend on the result of an earlier query. *zenvisage* is a database-oriented system; thus, implementation and evaluation focus on *how* the queries are executed and their performance[9]. For *zenvisage* automation of visualization was out of scope, but the data model for the user interface was based on a grammar of graphics and used *Vega-Lite* for the implementation of the user interface.

The prototype was evaluated in a user study with a within-subject design, with 12 participants with data analysis experience. The tasks were based on interviews with experts and performed on a dataset that participants could relate to (housing data). After a familiarization period, participants performed tasks on both zenvisage and a baseline system. Follow-up questions were asked by e-mail afterward if needed. In evaluation, participants valued the possibility to search for attributes that match a trend (i.e., the generations of sets of visualizations instead of enumerating attributes manually) for finding correlations.

2.3.3 Summary

This section provided a brief summary of literature relating to two imports aspects of visualization recommendation systems: models of visualization, and how previous studies designed visualization recommendation systems. Studies that focused solely on the visualization of data (i.e., without visualization recommendation) were not included.

The included studies have reported the ubiquitous usage of two concepts. Most research on visualization systems has emphasized the use of a model of visualization based on the model of Bertin [Ber83], which more recent work implemented as implementations of a grammar of graphics [Wic10]. Furthermore, almost every paper on visualization recommendation includes the notion of expressiveness and effectiveness criteria for visualizations as introduced by Mackinlay [Mac86]. Together, these studies provide valuable insights into the design of visualization recommendation systems, and are in general agreement on the following concerns[10]:

[9] I.e., runtime and number of SQL queries issued.
[10] List of citations for each concern is not exhaustive and generally follows their first usage.

DESCRIBING PLOT TYPES BY SCALES OF MEASUREMENT

Plot types are distinguished by the scales of measurement of the variables on their axes [Ber83].

MARK TYPES

Systems distinguish mark types [Ber83; Mac86].

USING SCALES OF MEASUREMENT

Scales of measurement are used to describe the type of variables [Ste46; Ber83; Mac86] and often specialized with subtypes [Rot+94; STH02].

VISUAL EFFECTIVENESS

The effectiveness of (retinal) encodings differs and can be used to rank them [Mac86; CM84].

SCALE OF MEASUREMENT AND CARDINALITY INFLUENCE ENCODING

Scales of measurement [Mac86], and the cardinality of a variable influence its possible encodings [STH02; Won+17].

VISUALIZATION INFLUENCES QUERY

The requirements of the visualization adept/lead to the query needed to retrieve the data [STH02].

LOGIC OR CONSTRAINT PROGRAMMING

Searching for the best encoding is a non-convex problem and is implemented using logic- or constraint programming [Mac86; Won+16b; Mor+19].

LEARNING TO RANK

Learning to rank is used to learn weights to rank visualizations [Mor+19].

TASK Recognizes the influence of task on the suitability of a visualization and uses this in recommendation [Mor+19].

This introductory section provides a brief overview of the rationale behind the implementation of Visualization Recommendation (VizRec) in the prototype system named OVERLOOK. The chapter then goes on to describe the structure of the implemented solution. What follows is a detailed explanation of the steps of the implementation.

Before proceeding to examine the implementation of VizRec in OVERLOOK, it helps to take a moment to re-introduce choosing a visualization from the perspective of a search problem. As explained earlier in Section 2.2, there is a common language for describing visualizations[1]. This language describes a subset of all possible visualizations; some visualizations are not (intuitively) expressible in this (formal) language. A notable example of this is Minard's *Carte Figurative* (Figure 3.1), which Wickham [Wic10, p.18] provides as an example. While this visualization can be approximated using *ggplot2*, it is not intuitive to do so.

In addition to the limitation that not all visualizations can be expressed using a grammar of graphics, there is a sub-set of all visualizations that is *expressive* and communicates the pattern in the data (e.g., "expressive", "good", "intuitive", ... visualizations).

The *expressiveness* and *effectiveness* criteria, as introduced by Mackinlay [Mac86] are a method of formalizing knowledge about what makes an expressive visualization. In turn, the set of graphics that adhere to these criteria make up the space of visualizations considered by such an automated visualization system. Not all of the visualizations considered are possible visualizations.

An automated visualization system has the goal of creating visualizations that are in the intersection of (a) the language of the implementation of a grammar of graphics it uses, (b) expressive visualizations, and (c) visualizations considered by the system.

Most of the criteria are logical for humans. For example, for a chart to make sense, the essential axes are used (e.g., x-axis, y-axis), and all retinal variables are used at most once. Besides, there are aspects of good charts, for example, that a chart should prefer an effective encoding over a less effective encoding (e.g., $color$ over $shape$) that can be encoded as criteria.

[1] The (formal) grammar of graphics defines the language of valid graphics in that language.

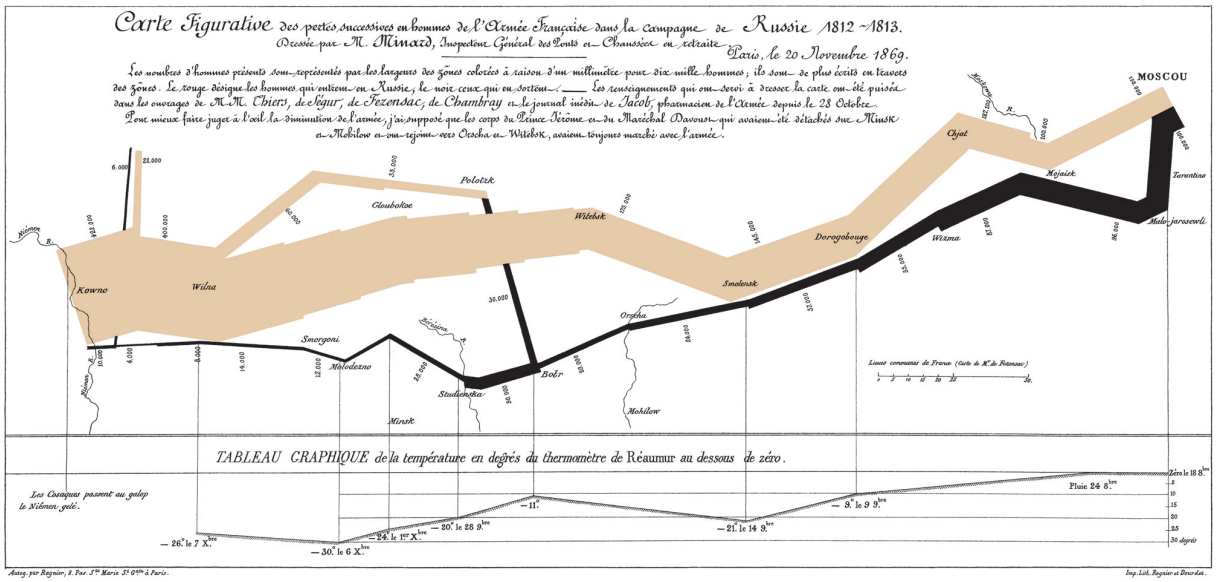


Figure 3.1: Charles Minard's *Carte Figurative*, Wikimedia Commons [Min69].

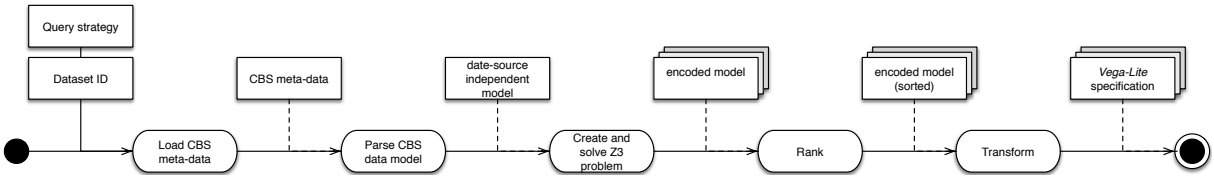
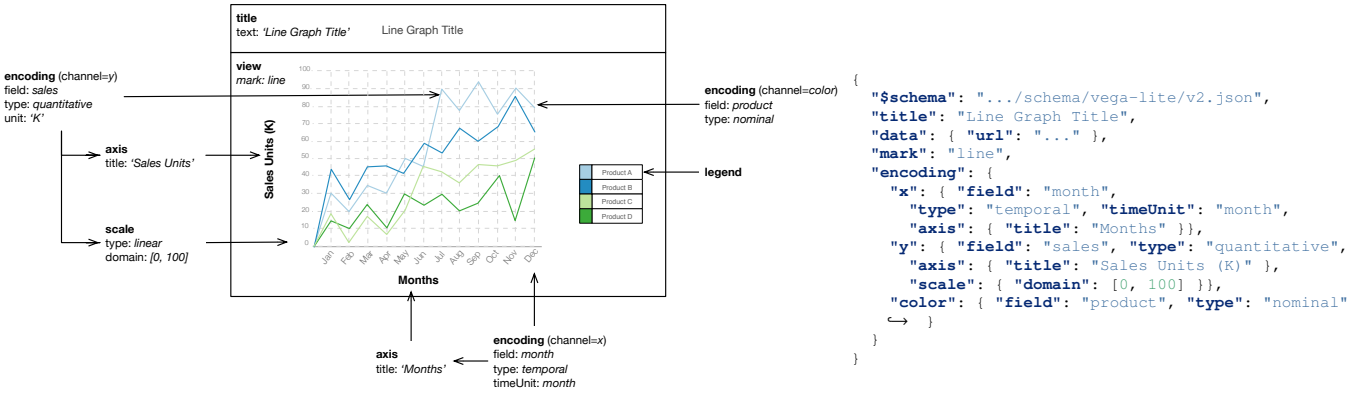


Figure 3.2: Steps in the implementation of VizRec in OVERLOOK.



(a) Annotated with *Vega-Lite* attributes

(b) Specified with *Vega-Lite*

Figure 3.3: Elements of a chart.

3.1 HIGH-LEVEL DESCRIPTION OF OVERLOOK

OVERLOOK implements VizRec as a constraint optimization problem. The problem is solved using Z3 and is implemented in multiple steps (see Figure 3.2). The solutions of the optimization problem form a Pareto frontier[2], with each solution being a prototype for a visualization. Each solution is translated into a visualization specification in an intermediate model. These specifications contain the allocations made, as well as scores for the heuristics matched and allocations made in the visualization. These visualization specifications are then re-ranked based on the sum of their scores. Finally, these visualization prototypes are transformed into a *Vega-Lite* specification, which is rendered in a user interface using *Vega-Lite*.

Vega-Lite is an implementation of a grammar of graphics, that uses a declarative JSON specification to define a visualization. A specification (see Figure 3.3(b)) contains *encodings* for multiple axes, a *title*, a type of *marks*, and definitions of axes. The *mark* attribute defines the type of visualization and thus implies available encoding channels. Figure 3.3(a) shows these elements annotated on an example chart.

3.2 CONSTRAINT-BASED MODEL OF VISUALIZATION

VizRec can be viewed as a constraint optimization problem, using expressiveness criteria to find valid visualizations, and effectiveness criteria to order them. Expressiveness criteria can be viewed as constraints that are required to be satisfied for a visualization to be valid, and effectiveness and other heuristic goals can be seen as optimization goals for the quality of the visualization.

The descriptions of expressiveness criteria and the priorities of possible encodings (i.e., effectiveness criteria) differ in literature. The data model and priorities used in OVERLOOK are based on the model of *Vega-Lite* since this is used in the implementation of the user interface.

For each type of chart, the possible axes are known[3]. The visual variables map to *encoding channels* in *Vega-Lite*[4]. The possible encodings are restricted by the properties of a field, including the cardinality and

[2] A set of allocations that are each optimal for one or more criteria, more formally introduced later in Section 3.3.3.
[3] E.g., "a bar chart has a slot for x, y".
[4] `https://vega.github.io/vega-lite/docs/encoding.html`, retrieved on 2019-01-22.

Name	Description
Expressiveness criteria	
possible encodings	Possible encodings for a field.
used	Every selected field is encoded.
per type	Only one encoding of each type (e.g., retinal) is used per field.
mutually exclusive	Each encoding can only be used once.
sharing	Shared encodings all have the same visual variable.
required axes	The required axes are used.
color or saturation	Color and Saturation can not be used at the same time.
Effectiveness criteria	
score encoding	Use the most effective visual variable (maximize the score of the encoding).
Heuristics	
time	Prefer time on main axes.
topics	Prefer topics on main axes.
preference	Prefer time over topics.

Table 3.1: Criteria and heuristics included in OVERLOOK.

type of measurement, e.g., "colors can be used for up to 7 nominal values". These are hard constraints or expressiveness criteria.

The effectiveness criteria are a different type of constraint. Determining the possible encodings given a mark type (chart of type) and a set of fields is not a concave problem; choosing the best encoding for the first field could mean that the remaining best choice for the second field leads to a lower overall utility. This implies that a greedy approach does not work and that in order to find the best encoding, all possible encodings need to be evaluated.

The number of constraints is dynamic, and a possible result needs to adhere to all restrictions. Additionally, several heuristics are used to guide the solver toward good charts. For example, there is the convention that *time* is displayed on the x-axis. These heuristics are soft constraints and have different utilities (scores); some heuristics take priority over others. In literature this process is often implemented using logic- or constraint-programming [Mac86; Won+16b; Mor+19].

OVERLOOK describes encoding recommendation as a Satisfiability Modulo Theories (SMT) problem and uses the Z3 theorem prover [MB08] to solve this problem using Pareto optimization. In this problem, the expressiveness criteria are encoded as hard constraints on solutions. The effectiveness criteria (such as the utility of a visual encoding) are encoded using optimization objectives. A part of the effectiveness criteria is implemented in Python code. For example, the lookup of the possible encodings for a field, given its cardinality and scale of measurement, is implemented this fashion. The results are equivalent to if this was encoded this in the SMT problem. The expressiveness criteria, effectiveness criteria, and heuristics included in OVERLOOK are listed in Table 3.1.

When this system is solved, Z3 yields results that adhere to all hard constraints. The results are on the Pareto frontier of optimal allocations, given the constraints[5]. The heuristics are independent constraints of which multiple can apply for a solution. A distance function is used to sort all the possible solutions and pick one of the optimal ones. For the top-ranking solutions, a chart object is constructed. Finally, this chart object is transformed into a *Vega-Lite* specification.

[5]For example: A scatter-plot where the x-axis and y-axis are switched.

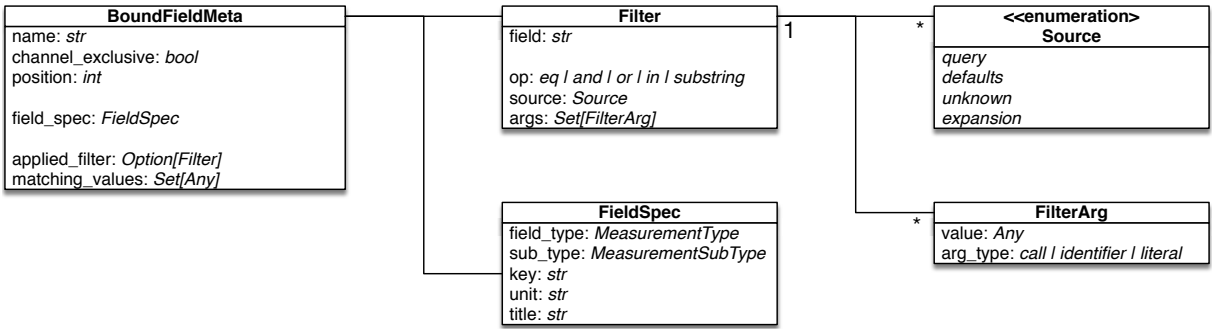


Figure 3.4: Diagram of the data source independent data model.

Scale of measurement	Subtypes
Ordinal	region, time
Nominal	string, topics
Quantitative	topic_values, monetary, monetary_per_unit, percentage, number, relative

Table 3.2: Subtypes of values.

3.3 IMPLEMENTATION OF VISUALIZATION RECOMMENDATION IN OVERLOOK

Before proceeding with the introduction of the steps in the implementation of VizRec in OVERLOOK, it will be necessary to introduce the data-model and corollary functions used. The paragraphs that follow describe (1) the basic structure of the SMT problem; (2) the logical constraints created for fields; (3) the global constraints, for heuristics; (4) how the solutions to the SMT problem are transformed into visualization specifications; and finally (5) how these solutions are ranked.

3.3.1 Data model

As explained earlier in Section 2.1, the meta-data model used by OVERLOOK was designed to be independent of the data source. A chart is specified by the type of visualization and a set of *BoundFieldMeta* objects. Together with the related objects (see Figure 3.4), each of these *BoundFieldMeta* objects describes a field in the dataset used in the visualization.

The *BoundFieldMeta* objects provide an abstraction for both the dataset and the query. It contains information on the field (*FieldSpec*), information on the query (*Filter*), and information on how the field is used for this chart (position, name, selected values, whether the field can share its axis with another field).

The data source performs pre-processing to create these objects. Some fields in the data source may be split up into two objects if they are used both as a topic and a dimension[6]. For *ordinal* and *nominal* fields, this pre-processing includes calculating the number of matching values. All these operations are performed "offline" — without interaction with the data source.

FieldSpec objects contain the information needed to display a field as either an axis or item of the legend. The object includes a textual description as well as the scale of measurement. Sub-types of scales of measurement were added for more precision (Table 3.2). For now, this information is not used in the SMT problem; however, it is used while creating the visualization specification.

[6]This is applied to "topics" from CBS.

Channel	Axis channel	Visual variables
Position	✓	x, y, x_2, y_2, region
Row	✓	row
MarkProperty		color, opacity, shape, size
TextTooltip		text, tooltip
LevelOfDetail		detail
Order		order

Table 3.3: Visual variables for each of the (Vega-Lite) channels, ordered by preference (descending).

Chart type	Visual variables
Line	x, y
Bar	x, y, column
Map	region, column
Table	x, y, x_2, y_2

Table 3.4: Visual variables on the axes of chart types by order of preference.

3.3.2 Corollary functions

Previous studies of VizRec systems typically include preferences and limitations on mark and encoding types. A similar component is included in OVERLOOK. These apply to *all visualizations* and are independent of the implementation as a SMT problem (which will use them).

Channels. The visualization is rendered with *Vega-Lite*, which defines the possible channels for charts and possible mark types (Table 3.3). These are ordered by the visual quality of the channels (top-bottom) and within a channel (left-right).

axisEncodings. This function uses the table of channels and visual variables (Table 3.3) to define what visual variables are an axis of a chart:

$$\text{axisEncodings}(chartType : ChartType) \rightarrow \text{Set[Encoding]}$$

isPrimaryAxisChannel. Another function indicates if an encoding is a required axis of a chart. An encoding is required if the visual variable is on an axis of the chart type (Table 3.4) and it is x, y, or $region$.

$$\text{isAxisChannel}(encoding : \text{Encoding}) \rightarrow \text{bool}$$

possibleEncodings. The possible (distinguishable) visual variables for a field depend on the scale of measurement of the field and the number of values. A viewer should be able to distinguish the order of elements (for an ordinal scale) or individual elements (for a nominal scale). Human perceptual capabilities limit the number of different values that can be distinguished and thus be used by retinal encodings [STH02, p.8]. The mapping used by possibleEncodings is listed in Table 3.5.

$$\text{possibleEncodings}(measurementType : \text{MeasurementType}, values : \text{Sized}) \rightarrow \text{Set[Encoding]}$$

Scale of measurement	Number of values	Visual variables
Nominal	< 7	x, x_2, y, y_2, row, column, *color*, size, shape, region
Nominal	$>= 7$	x, x_2, y, y_2, row, column, size, shape, region
Ordinal	< 6	x, x_2, y, y_2, row, column, color, *size*, shape, region
Ordinal	6	x, x_2, y, y_2, row, column, color, *shape*, region
Ordinal	$7 <= \|values\| < 12$	x, x_2, y, y_2, row, column, color, region
Ordinal	$>= 12$	x, x_2, y, y_2, row, column, shape, region
Quantitative	∞	x, x_2, y, y_2, size, color, text, region

Table 3.5: Available visual variables by scale of measurement and number of values.

scoreEncoding. Another method evaluates the quality of an encoding. The heuristic used in function is that a field should use the most perceptually effective encoding (locally), but that quantitative fields have priority over ordinal/nominal fields for the most effective encodings. This leads to a preference for quantitative fields on the axes (the highest quality encodings).

The index in the ordering of *all* encodings is used as the base of the score. This index is then used to calculate a score that prefers *all* encodings for quantitative fields over ordinal and nominal fields. The scores for quantitative fields range from $\left|encodings\right| + 1$ to $2\left|encodings\right|$, while for ordinal and nominal fields the scores range from 0 to $\left|encodings\right|$.

$$\text{scoreEncoding}(measurementType : \text{MeasurementType}, encoding : \text{Encoding}) \rightarrow \text{int}$$

3.3.3 Steps of visualization recommendation

Problem structure. The basic structure of the SMT problem consists of a *const* for each field used in the visualization (e_i) and an enumeration of all the channels of the visualization (C). A solution assigns encodings to fields while adhering to the constraints. In the pseudo-code in this chapter, model. add($goal$) adds a hard constraint to the SMT problem and model. optimize($goal$) adds an optimization goal. In the final SMT instance, we defined the following variables[7]:

C enumeration (type) of all the *Vega-Lite* channels.

e_i state variable indicating the encoding for field i.

The solver is set-up for Pareto optimization in order to find all possible matching solutions on the Pareto-front. A Pareto optimal allocation is an "optimal" allocation $\{x_1, \ldots, x_i\}$ where there is no allocation $\{x'_1, \ldots, x'_i\}$ where for **each** i, $u(x'_i) > u(x_i)$[8]. Note that this considers the optimal utility *for each* criterion without considering the exact utility for a criterion[9]. Since part of the SMT problem is encoded with heuristics that are implemented with scores, the Pareto front contains solutions with differing (total) scores that are all Pareto-optimal. OVERLOOK ranks visualizations by the sum of their component scores.

Problem setup. After the solver is set up, the constraints are added. These can be grouped into constraints for fields, heuristics on fields, and global constraints. The SMT problem is built using the algorithm set

[7] These are not used to in the pseudo-code in this chapter. However, in the implementation, the constraints added to the Z3 model are defined in terms of these variables.

[8] With $u(x_i)$ defined as the utility of x_i.

[9] I.e., criterion are independent, the exact values are not considered.

Algorithm 1 Main SMT problem setup.

Precondition: $chartType$ is the chart type, $fields$ a set of field objects, and $model$ is a Z3 context.

```
function encodeProblem(chartType, model, fields)
    for f ∈ fields do
        constrainFields(model, field)              ▷ Add constraints for each field.
    constrainDistinctFields(model, fields)         ▷ Ensure fields have distinct encodings.
    constrainRequiredAxes(chartType, model, fields)  ▷ Ensure required axes are used.
```

Type	Visual variable	
	x	$column$
Time	110	55
Topics	100	50

Table 3.6: Scores for *Time* and *Topics* heuristics.

out in Algorithm 1, of which the structure is described below. In the algorithms, the implementations of criteria and heuristics have been emphasized[10].

For each field, the possible encodings, and an optimization goal for the optimal encoding are added (Algorithm 2). If applicable, a heuristic score (see Table 3.6) is added as an optimization goal. Afterward, global constraints are added. This is performed in two steps. In the first step, the grouping of fields is considered. For fields that can be grouped, a constraint is added that ensures that all fields in a group have the same value, and a characteristic element from the group is picked. In the second step, a constraint is added that ensures that non-grouped fields and the characteristic elements of the groups are distinct. This guarantees that visual variables are only used once (Algorithm 3). Finally, a constraint is added that ensures that the axes of the chart are used by one of the fields (Algorithm 4).

Solving. Once the model is set up, its solutions are used to create candidate visualization specifications. Z3 checks for *satisfiability* of the model and all Pareto-equal solutions are gathered. Each solution contains both *assignments* for the variables as well as *score* on objectives (heuristics, quality of assignments). Only the Pareto front is considered instead of enumerating all possible solutions. When this process finishes, the result is a set of possible visualization specifications. As explained earlier, these are Pareto-equal (per objective) but have different global utility.

Ranking and transformation. Finally, the solutions are ranked and transformed into a *Vega-Lite* specification. Many approaches are suitable for ranking the solutions. Overlook sorts the solutions on the Pareto front by the sum of their component scores. This specific method could be added to the Z3 model; however, it is performed in a separate step for extensibility. Afterward, the best candidates are transformed into *Vega-Lite* specifications.

[10]Note that the *per type* and *color or saturation* expressiveness criteria are implied by the fact that overlook only assigns a single visual variable to a field.

Algorithm 2 Sets up the constraints and heuristics for a single field.

Precondition: $fields$ is a set of field objects, and $model$ is a Z3 context.

function ConstrainFields($model, fields$)
  for $f \in fields$ **do**
    $E \leftarrow$ possibleEncodings($field$) ▷ *Expressiveness criterion*: possible encodings.
    $oneOf \leftarrow \bigvee_{e_i \in E} f = e_i$ ▷ *Expressiveness criterion*: used.
    $bestEncoding \leftarrow \sum_{e_i \in E} \begin{cases} \text{scoreEncoding}(f, e_i) & \text{if } f = e_i \\ 0 & \text{otherwise.} \end{cases}$
    model. optimize($bestEncoding$) ▷ *Effectiveness criterion*: score encoding.
    model. add($oneOf$)
    if type(f) $\in$ {Topics, Time} **then**
      $condScore \leftarrow \begin{cases} 100, & \text{if type}(f) = \text{Topics} \wedge f = \text{x;} \\ 55, & \text{if type}(f) = \text{Topics} \wedge f = \text{column;} \\ 110, & \text{if type}(f) = \text{Time} \wedge f = \text{x;} \\ 55, & \text{if type}(f) = \text{Time} \wedge f = \text{column;} \\ 0, & \text{otherwise.} \end{cases}$ ▷ *Heuristic*: preference.
      model. optimize($condScore$) ▷ *Heuristics*: time, topics.

Algorithm 3 Global constraints: encodings are only used once unless otherwise specified.

Precondition: $fields$ is a set of Field objects, $model$ is a Z3 context

function ConstrainDistinctFields($model, fields$)
  $f_d \leftarrow \{y \in fields \mid y.channelExclusive\}$
  $f_{nd} \leftarrow \{y \in fields \mid \neg y.channelExclusive\}$
  $distinct \leftarrow f_d$
  $byType \leftarrow$ GroupBy($f_{nd}, (f) \rightarrow f.subType$) ▷ Group fields by their types.
  for $(type, fields) \in byType$ **do**
    $first \leftarrow fields[0]$
    $eqToFirst \leftarrow []$
    for $f \in fields$ **do**
      if $f \neq first$ **then**
        eqToFirst. append($f == first$) ▷ $f == first$ is a Z3 expression (c.f. boolean)
    model. add(And($*eqToFirst$)) ▷ *Expressiveness criterion*: sharing.
    distinct. append($first$) ▷ The first element is used as a representative of a group.
  model. add(Distinct($*distinct$)) ▷ *Expressiveness criterion*: mutually exclusive.

Algorithm 4 Ensures the required axes are used.

Precondition: $chartType$ is the chart type, $fields$ a set of field objects, and $model$ is a Z3 context.

function ConstrainRequiredAxes($chartType, model, fields$)
  $requiredAxes \leftarrow \{c :$ axisEncodings($chartType$) $\mid isPrimaryAxisChannel(c)\}$
  for $r \in requiredAxes$ **do**
    $oneOf \leftarrow \bigvee_{f \in fields} f = r$ ▷ *Expressiveness criterion*: required axes.
    model. add($oneOf$)

An important aspect of evaluation is the design of the data collection instruments. The evaluation of OVERLOOK investigates the usability of visualizations. This evaluation is different from classical settings (the usability of a system, in human-computer interaction; the relevance of documents, in information retrieval) while being very similar to evaluations for interactive information retrieval.

This chapter will first position the evaluation approach. It then explores the goals of the evaluation, what elements the evaluation contains, and finally, the design of the instruments.

4.1 EVALUATION IN INFORMATION VISUALIZATION

Evaluation in information visualization is complicated since it considers the tool under study, the process that the tool supports, and the visualizations simultaneously. The problem of how to carry out specific evaluation methods has been extensively studied. However, few studies focus on when to choose a specific evaluation type [Lam+12].

In a comprehensive literature review of evaluation scenarios, Lam et al. included 850 papers from four information visualization publication venues, 361 of which contained at least one evaluation. The authors distinguish four scenarios that focus on data analysis and three scenario's that focus on evaluating visualization performance. Most of the papers focus on visualization performance with *evaluating user performance* in 33 %, *evaluating user experience* in 34 %, and *evaluating visualization algorithms* in 22 % of papers that include an evaluation.

It is evident that evaluation is rare in information visualization. Most evaluations use a within-subject design with a limited number of selections from known datasets and a limited number of participants. The dataset and task are generally chosen so that participants can relate to the data[1]

A recent trend is the usage of online user experiments instead of a laboratory setting — with the number of participants magnitudes higher than in laboratory studies. This makes it possible to investigate more variables and/or use different research designs.

In their large-scale online user study of the effectiveness of scatterplots, Kim and Jeffrey Heer [KH18] performed a mixed design study assigning visual encoding within-subject and task and data distribution between-subject. The data was created by sampling from US daily weather data to create datasets with specific cardinalities and distribution shapes. Compared to a laboratory setting, this type of online user experiment uses simpler questions (binary) and generally only provide (detailed) quantitative results[2].

In contrast to these studies, in the evaluation of OVERLOOK, [characteristics of the] datasets are varied as inputs to the visualization recommendation system. Furthermore, there is no specific component task for the visualization recommendation system except to "*generate understandable visualization prototypes*". Suitable component tasks for visualization types are a result instead of an input.

4.2 EVALUATION OF OVERLOOK

The evaluation of OVERLOOK is positioned between the evaluation of a visualization system, evaluating user experience; and Information Retrieval (IR) evaluation, where the performance of a system on multiple search topics is measured. This approach is similar to Interactive Information Retrieval (IIR) evaluation and uses user-oriented methods to evaluate system performance.

The evaluation is designed as a within-subject usability study in a laboratory setting which evaluates the interactions of *visualization type*, *query heuristics*, and *dataset* and compares the utility of the visualizations

[1] Similar to the usage of *simulated work tasks*, by Borlund and Ingwersen [BI97], in Interactive Information Retrieval (IIR).
[2] Cf. a combination of quantitative and qualitative results.

to a baseline result from StatLine. All results are evaluated using instruments that will be introduced in Section 4.3.

The system under test is a non-interactive system that is early in its design process. The goal of the evaluation is twofold: validate that the system works for multiple datasets, and investigate the influence of inputs on the quality of results. This goal led to the choice to explore a high number of items with a limited number of participants, without balancing effects or a between-subject design, instead choosing to have a low number of overlapping ratings on items (as is common in IR). This method was chosen because it is particularly useful when studying a system where the influence of different parameters is still unknown.

Measuring usability. The beginning of this section left open the conceptualization of performance because it would distract from the discussion of the goals. Performance is viewed as usability. However, a precise definition of usability has proved elusive. The International Organization for Standardization (ISO) [Int98, p.2] defines usability as the extent "*to which the users of products are able to work effectively, efficiently and with satisfaction.*". The ISO definition is the most commonly used definition of usability and defines usability as a concept that includes measures for effectiveness, efficiency, and satisfaction. These concepts are presented below:

EFFECTIVENESS "the accuracy and completeness with which these goals can be achieved."

EFFICIENCY "the level of effectiveness achieved to the expenditure of resources"

SATISFACTION "the extent to which users are free from discomfort, and their attitudes towards the use of the product"

A good summary of the evaluation of usability and interactive information retrieval prototypes has been provided in the work of Kelly [Kel09]. In this thesis, effectiveness will be measured by measuring whether subjects indicated errors in the visualization and by their indication of how hard it was to understand a visualization. Efficiency will be measured by measuring the time between a visualization being displayed and the user beginning his annotation of the visualization. Finally, satisfaction will be measured by open questions and their indicated preferences between OVERLOOK and the baseline.

4.3 INSTRUMENTS

To measure usability, three questionnaires were designed. These are included in Appendixes 7.4 and 7.4. During the study three common elements of the structure of a usability experiment [Kel09, p.97] were used: a demographic survey in order to gather descriptive statistics of the population, a post-task questionnaire to rate each document, and an exit survey. The text in this chapter describes the instruments as used in the final evaluation. An earlier iteration of these instruments was evaluated in the preliminary evaluation, which is presented in Chapter 5.

Usability will be measured using the measures listed below. These measures are grouped by the instrument used for data collection. A detailed account of the design of the post-task questionnaire is given in Section 4.4.

Demographic questionnaire

AGE OF SUBJECT 15–25, 25–50, 50+

EDUCATION FIELD OF SUBJECT science, technology, engineering and, mathematics or other

EDUCATION LEVEL OF SUBJECT high school, Bachelor of Science, Master of science

Visualization experience	basic, intermediate, expert, machine-learning. Based on examples
Experience rating	1=strongly disagree, 7=strongly agree
Experience (time)	Years of experience with data visualization

The personal details were optional. Despite its common inclusion in descriptive statistics of populations in literature, a question on gender was not included.

Post-Task questionnaire

Errors	(missing data, sparse data, visualization is not displayed properly)
Usability	(ease of understanding, Likert-type scale, 1=strongly disagree, 7=strongly agree)
Suitability for task	(selection of set of suitable component tasks)

Exit questionnaire

Utility	what stood out in the visualizations?
Design rules	the system seems to follow the following design rules. . .

System logs

Interaction	time to start answering the questionnaire after visualization is displayed.
Interaction	time until submitting the questionnaire.

4.4 design of post-task questionnaire

Previous studies either consider the usability of a system (in Human-Computer Interaction (HCI)), or the relevance of documents returned (in IR), or the experience while using a search system (in IIR). This evaluation considers the usability of a document returned by system returning ranked results.

The design of the post-task questionnaire was based on standard instruments when possible. However, because of the position at this intersection of fields and the need for system-specific questions, a single standard questionnaire could not be used.

It was decided after a preliminary evaluation that the best method to adopt for this investigation was to measure three separate concepts for each visualization that is generated. The first consists of possible errors in the visualization. The second concerns the usability of the visualization. A final question concerns the component task the visualization is suitable for. The following paragraphs will introduce literature on- and discuss the design of the questions for each of the concerns included in the questionnaire.

4.4.1 Error types

During the development of the system, "broken visualizations" were common. While developing the evaluation interface, it became apparent that this had multiple causes. The error types and their description were first added and then iteratively refined during development and initial testing of the evaluation interface.

Within these errors, two groups can be distinguished. The first group of errors, data errors, are caused by the selection of data that cannot create a good visualization. The second group, visualization errors, are the result of a poorly designed visualization. The following four error types were included, with the accompanying descriptions:

Missing data	Data is missing
Visualization not displayed properly	The visualization is not displayed properly; it does not fit in the available area without scrolling or was truncated.
Data not displayed properly	The data is not displayed properly. Marks are not distinguishable or overlap. In some situations, (part of the) data is not visible because of this.
Sparse data	The data is sparse; some marks (bars, dots, . . .) are missing and this leads to a bad visualization.

4.4.2 Usability

A post-task rating of difficulty was chosen as a measure for usability. This type of rating is commonly used to provide diagnostic information and provide an estimate of usability. The rating needs to be both reliable and easy to use.

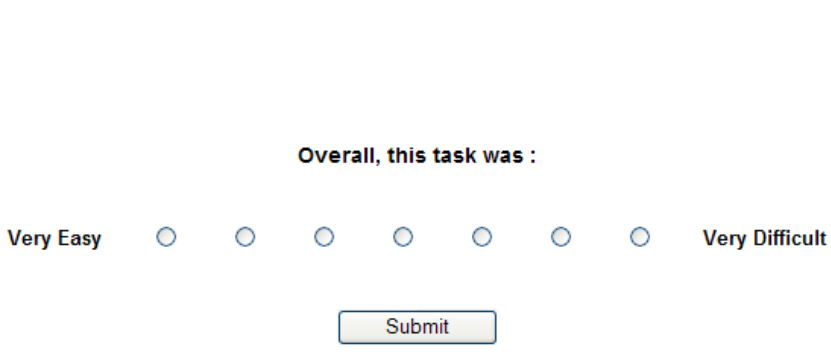
At the system-level, the System Usability Scale (SUS) by Brooke [Bro96] is one of the most commonly used scales for measuring user satisfaction. The SUS consists of ten statements scored on a Likert-type scale (1=Strongly disagree, 5=Strongly agree, the score ranging from 0–100 is a weighted sum of the items with defined weights). Its primary use is estimating and classifying the usability of a system, and monitoring the usability of a system over time [BKM08]. An alternative at the system level is heuristic evaluation [NM90]. While the SUS is found to be reliable, it is not suitable as a post-task questionnaire due to its length. Besides, measuring user satisfaction immediately after an event potentially increases its validity [SD09].

J. R. Lewis [Lew90; Lew93] was one of the first to examine the development and evaluation of standardized questionnaires for subjective usability using psychometric methods[3]. The resulting questionnaire, the After-Scenario-Questionnaire (ASQ) is a three-item questionnaire that addresses ease of task completion, time to complete a task, and adequacy of support information answered using a 7-point graphic scale (1=Strongly disagree, 7=Strongly agree), with an option for not applicable ("N/A") positioned outside the scale.

Tedesco and Tullis [TT06] evaluated multiple methods of eliciting subjective user feedback. Included conditions included two out of three ASQ questions (support information was not deemed to be relevant); two variants that inquire about self-assessment of task difficulty ("Overall this task was . . . ", 1=very easy, 5=very difficult), "before and after" questions, for rating expectations of task difficulty; and usability magnitude estimation, which measures ratios between subjective ratings for multiple situations. In an online experiment with 1131 participants where each participant was assigned to one out of five conditions and performed seven tasks on a system. All rating techniques correlated with user performance on the tasks performed. While all rating techniques identified significant differences between the difficulty of the tasks performed, the single question was most consistent at small sample sizes.

Sauro and Dumas [SD09] evaluated three one-question rating types in a within-subject experiment with 26 participants performing five tasks on two systems, evaluating a Likert-type scale with 7 points (Figure 4.1(a)); Usability Magnitude Estimation (UME), where the difficulty is compared relative to a baseline task; and an online variant of the Subjective Mental Effort Question (SMEQ), which displays a continuous linear scale with no upper boundary (as shown in Figure 4.1(b)). The rationale for the SMEQ is that subjects can re-interpret their difficulty scale after encountering earlier samples. There was

[3] According to a definition provided by J. R. Lewis, "The goal of psychometrics is to establish the quality of psychological measures (Nunnally, 1978). Is a measure reliable in the sense that it is consistent? Given a reliable measure, is it valid (measures the intended attribute)? Finally, is the measure appropriately sensitive to experimental manipulations?" [Lew93, p.2].



(a) Variant of Likert-type scale

(b) The Subjective Mental Effort Question

Figure 4.1: Examples of scales, from [SD09].

a significant difference in the performance of the SMEQ and the Likert-type scale. UME was hard for participants to learn and had a lower sensitivity. However, with small sample sizes (below ten to twelve), none of the post-task question types (nor post-test questionnaires such as the SUS) have high detection rates.

The questionnaire used in the evaluation of OVERLOOK included a single Likert-type question on perceived difficulty ("How easy was it to understand this visualization") with seven points and "N/A" displayed beside the scale. The choice for seven points was made because of its higher reliability.

4.4.3 Component tasks

The following section will first introduce an early paper on visualization tasks and then describe multiple (grounded) taxonomies that were used in the synthesis of the classification used in this study.

Several attempts have been made to create a task taxonomy of information visualizations. One early study that is often cited in research is that of Shneiderman [Shn96] which introduced "overview first, zoom and filter, then details-on-demand" as a mantra for the user interface design for visualization prototypes, and postulated seven component tasks (that relate to user interface interactions) that relate to the tasks in other taxonomies.

OVERVIEW	Gain an overview of the entire collection
ZOOM	zoom in on items of interest
FILTER	filter out uninteresting items
DETAILS-ON-DEMAND	Select an item or group and get details when needed
RELATE	View relationships among items
HISTORY	Keep a history of actions to support undo, replay, and progressive refinement
EXTRACT	Allow extraction of sub-collections and of the query parameters

In their work on data characterization that introduced the foundation of SAGE, S. F. Roth et al. distinguished several information-seeking goals[4]. These were: *accurate value lookup*, *comparison of values*, *pairwise or n-wise comparison* (between series in a dataset), *distribution of values*, *correlations*, and *indexing a structure by an element* [RM90; Rot+94].

[4]Named "display goals".

Wehrend and C. Lewis [WL90] draw a distinction between object classes and operation classes and align their operation classes with these of S. F. Roth and Mattis [RM90]. The operation classes are abstracted from an analysis of 90 representation problems and distinguish the classes *identify*, *locate*, *distinguish*, *categorize*, *cluster*, *distribution*, *rank*, *compare within and between relationships*, *associate*, and *correlate*.

In a study which set out to determine *components* of visual activity Amar, Eagan, and Stasko [AES05] found ten clusters of analysis tasks. Students proposed 196 valid analysis tasks which were then clustered, providing both a taxonomy of tasks as well as example tasks for each cluster. Only primitive tasks were included, tasks that could be composed of primitive tasks were not included.

Retrieve value	given a set of specific cases, find attributes of those cases
Filter	given some concrete conditions on attribute values, find data case satisfying those conditions
Compute derived value	given a set of data cases, compute an aggregate numeric representation of those data cases
Find extremum	Find data cases possessing an extreme value of an attribute over its range within the data set
Sort	
Determine range	Given a set of data cases and an attribute of interest, find the span of values within the set
Characterize distribution	Given a set of data cases and a quantitative attribute of interest, characterize the distribution of that attribute's value of the data set
Find anomalies	Identify any anomalousness within a given set of data cases with respect to a given relationship or expectation, e.g., statistical outliers
Cluster	Given a set of data cases, find clusters of similar attribute values.
Correlate	Given a set of data cases and two attributes, determine useful relationships between values of those attributes

A more systematic study of interaction primitives was reported by R. E. Roth in 2013 [Rot13]. This study views the interactions with visualizations as an aspect that is shared between geographic sciences (GIS, cartography, . . .) and information visualization. Roth interviewed 21 expert interactive map users using semi-structured interviews to elicit statements on user tasks (*operations*) or interactive functionality (*operators*). These statements (for both operations and operators) were then grouped in a card sorting study by 15 (other) expert interactive map designers.

The study distinguishes operands, what the action is performed on (e.g., "space in time: . . . interactions with the temporal component of the map"); interaction goals, goals of complete interactions (e.g., "predict: . . . interactions that are performed to forecast what may occur in the future based on current conditions"); objective primitives, primitives that are part of goals (e.g., "compare . . . interactions that determine the similarities and differences between two map features"); and operation primitives, which are actions (e.g., "annotate . . . interactions that add graphic markings and textual notes to the visualization"). For this work, we use the object primitives since these align with the primitives from the other taxonomies.

identify	describes interactions that examine an individual map feature

COMPARE describes interactions that determine similarities and differences between two map features.

RANK determine the order or relative position of three or more map features

ASSOCIATE characterize the relationship between multiple map features

DELINEATE interactions that are performed to organize map features into a logical structure

USED COMPONENT TASKS Due to semantic differences between, and differing numbers of elements of the taxonomies, it is not possible to align the primitives from all discussed taxonomies. However, three of the described taxonomies align very well, with S. F. Roth and Mattis [RM90] proposing low-level primitives, Amar, Eagan, and Stasko [AES05] providing similar primitives as well as a set of examples for each primitive, and R. E. Roth [Rot13] providing a systematically validated set of primitives.

In the evaluation instruments, the following component tasks were used. The set of component tasks consists of four of the objective primitives by R. E. Roth (excluding delineate), with an added "*overview*" primitive. Each of the primitives was reworded so as not to refer to maps. The component tasks can be seen as compositions of visual activities. Examples for the visual activities from Amar, Eagan, and Stasko [AES05] are used to provide examples for the visual activities. For example, "rank" contains the "sort" activity, for which "Rank the cereals by calories" is an example. These examples were included in the evaluation instructions, which are reproduced in Appendix 7.4.

OVERVIEW Gain an overview of the entire collection

IDENTIFY [describes] interactions that examine an individual visualization feature

COMPARE describes interactions that determine similarities and differences between two visualization features.

RANK determine the order or relative position of three or more visualization features

ASSOCIATE characterize the relationship between multiple visualization features

In the previous chapters, we introduced the design of a visualization recommendation system and the instruments for evaluating such a system. Before proceeding to the full evaluation of the system, the evaluation materials need to be tested.

This chapter describes and discusses a preliminary evaluation of the system and data collection instruments. The preliminary evaluation investigates the quality of the evaluation interface, the prototype system, and the instruments. Furthermore, the results of the preliminary evaluation were used to motivate the chosen variations included in the final evaluation.

The first section of this chapter will start with a description of the parameters used for the evaluation and the evaluation protocol. The next section presents the results of the preliminary evaluation. Finally, we will discuss the changes to the instruments and the evaluation protocol made after the preliminary evaluation.

5.1 EVALUATION PARAMETERS

In order to evaluate whether the visualization generation and query handling were working, we evaluated all variations of chart types and query types. Since our experience during development was that the quality of the visualizations was highly dependent on the dataset, we selected five datasets, which are listed in the table in Appendix 7.4. Five chart types were included (*line*, *trail line*[1], *bar*, *horizontal bar*, and *circle*). These types include two pairs of closely related charts (*line* and *trail line*, *bar* and *horizontal bar*) which differ in their heuristics. Furthermore, five query types were tested (*default*, *most recent*, *all years*, *quartiles*, *most common*)[2]. StatLine changes the query when switching between views[3]. These query types attempt to have a similar effect.

The two top-ranked results from OVERLOOK for each of these permutations were included. This lead to 5 (datasets) ×5 (chart types) ×5 (query types) ×2 (results per setting) = 250 visualizations to be evaluated. A participant annotated these in a single session. This participant did not participate in the final evaluation.

5.2 ANALYSIS

In the analysis of these preliminary results we are investigating (a) what query types should be included in the final evaluation, (b) what chart types to include in the final evaluation, and (c) the questions used for the evaluation. Before analyzing the detailed results, during exploratory analysis, we verified that datasets, chart types, and errors influence the measured variables. Afterward, we investigated multiple measurements in the results. These are discussed in the paragraphs below.

Influence of chart- and query type. There are no significant results to use as a basis for the decision of what chart- and query-types to include in the final evaluation. However, there are other considerations. The *trail line* and *circle* chart types were excluded because they do not have equivalent visualizations in StatLine.

Effectiveness and relevance metric. The dataset contains two measurements for the effectiveness of a visualization. These are *ease of understanding* and *relevance*. The hypothesis is that these variables contain different information. In the questionnaire, *relevance* was an optional metric and not filled when a result was unusable. For this analysis, missing elements and *N/A* were filled with the lowest value possible for

[1] Chart with a line of varying thickness.

[2] The *most common* query type selects a single point in time. The other query types select all values with a certain interval from a time series.

[3] For example, when switching from a bar to line chart, years are added on the x-axis.

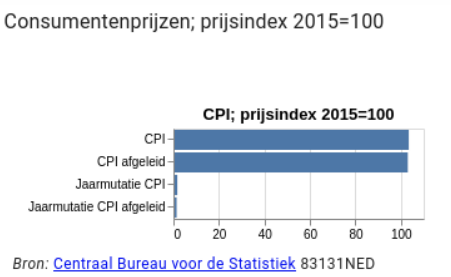


Figure 5.1: An example of a chart that is easy to understand but does not provide relevant information.

Question	Suitable task					
	compare	identify	not applicable	overview	rank	NULL
Most suitable task	40	16	17	76	3	138
Next suitable task	28	5	1	20	5	231
Total	68	21	18	96	8	269

Table 5.1: Most suitable component tasks.

the field. In order to decide which of these metrics to keep, we investigate whether they measure different results.

From the Shapiro-Wilk test, it follows that the p-values are less than the significance level ($p = 1.308 \times 10^{-14}$, $p = 4.784 \times 10^{-12}$), implying that the alternative hypothesis is true and that relevance and ease of understanding are not approximately normally distributed. Because the data is not approximately distributed, we use Kendall's rank correlation test to test for independence. From the result of Kendall's rank correlation test we find that the p-value is lower than the significance level ($p < 2.2 \times 10^{-16}$) and find that the alternative hypothesis is true; relevance and ease of understanding are correlated.

However, we have seen examples (such as Figure 5.1) of visualizations that are easy to understand but not relevant. This counter-example shows that while the measurements are correlated, they measure different concepts. For the final evaluation, both items are included, and both will be required.

Suitable component tasks. When testing the evaluation system subjects gave feedback that it was harder to rank the (abstract) component tasks than to choose which applied. In addition, as Table 5.1 shows, the data for the suitable tasks was sparse. Due to the question being optional, two values for not applicable were possible (explicit N/A. or by a NULL value). Based on this, for final evaluation, this question was changed into a field where the set of applicable component tasks is selected.

Error types. Errors in either the data or the visualization have a considerable influence on the rating of a visualization. Figure 5.2 shows the difference in mean ease of understanding (EOU) over all datasets when errors are present, split up by combination of chart- and query type. From the chart, it can be seen that on average, the existence of errors dominates the effect of the chart- and query type.

Factor analysis of the error variables is not possible due to it being dichotomous variables. Principal component analysis[4] finds that four principal components are needed to explain > 95% of the variance with each component explaining > 20% of the variance, indicating that none of the items is redundant.

Implicit data: Timing. Measured data includes the time it takes a participant to start filling the questionnaire after the visualization is displayed. This data contains a large number of outliers and is skewed right.

[4]Corrected for zero mean, unit variance of variables.

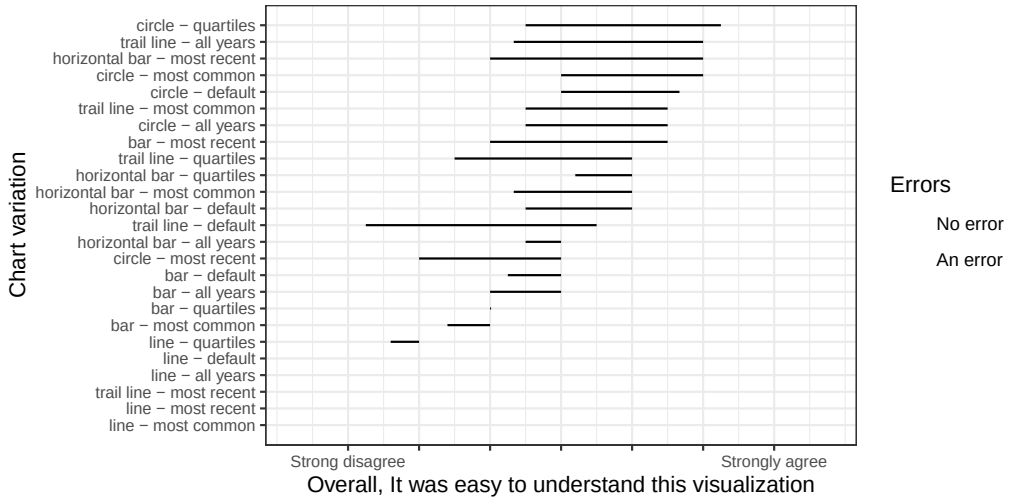


Figure 5.2: Dot plot showing the influence of errors on the EOU of the top-ranked result.

The mean time to start an answer is 73.0 s, the median is 3.2 s, and the maximum is 5486.4 s. When a user takes a break, this results in a very long duration.

In order to increase the quality of the timing data in the final evaluation, two measures were taken. First of all, participants will be instructed to reload the visualization after a break. Second, the amount of work performed by each participant will be reduced, so a session can be finished without taking breaks.

5.3 CONCLUSIONS

Based on the preliminary evaluation, we made several choices for the evaluation of OVERLOOK, resulting in changes to the evaluation protocol, data collection instruments, and the parameters included in the evaluation. Furthermore, we made several changes to OVERLOOK and the evaluation interface. The version of OVERLOOK evaluated in this chapter differs from the version used in the final evaluation.

Participants will be instructed to answer visualizations without taking breaks before answering the questionnaire for a visualization, to reduce the number of outliers in the measured durations caused by pauses. The number of visualizations evaluated by each participant will be reduced to reduce the time it takes to participate in the experiment.

To measure the quality of a visualization, the questionnaire includes both EOU as 7-element Likert scale item and the question on relevance. In contrast to the preliminary evaluation, an answer for both questions is required. The questions on error types and suitability for tasks were kept. The former is used to check the performance of the system on datasets, and the latter provides empirical data on task-chart suitability given aspects of the data. The error types have been (re-)grouped by type (data errors first, followed by visualization errors). Task suitability will be measured with a checkbox for each option instead of as an ordered preference (two sets of radio buttons).

Besides these changes to the data collection instruments, we changed the variations of OVERLOOK we evaluated. We removed the *trail line* chart, due to its inconsistent performance and lack of similar chart from StatLine; and the *circle* chart because there is no similar chart from StatLine. The levels of the query type were reduced to have similar behavior as StatLine: *default* is always included, the *most recent* type is used for bar charts, and *most common* for line charts.

We conducted a user study to assess OVERLOOK's ability to recommend visualizations for CBS datasets based on their meta-data. Participants evaluated the quality of visualizations using the instruments designed in Chapter 4, which we adjusted after the preliminary evaluation described in Chapter 5. We compared visualizations generated by OVERLOOK with visualizations from StatLine for identical datasets. To adjust for the data selection implicitly performed by StatLine when choosing a visualization type, we added conditions that used a similar adjustment to the data selection.

6.1 STUDY DESIGN

Visualizations depend on visualization *type*, *query type*, and visualization *tool*. Dataset is an independent variable. In this chapter, we will refer to a combination of a *dataset*, visualization *type*, and *query type* as [a] *query* [to the visualization recommendation system]. From the perspective of IR, the recommended visualization specifications are documents. Our study included 16 (dataset) ×3 (chart type) ×2 (variations) = 96 queries, with four visualizations each as documents.

Our study employed a within-subject design (concerning chart type, query type, and tool) with random assignment to conditions. We chose this design over a balanced design due to the need to explore multiple datasets while still taking a reasonable amount of effort for each participant. This design is similar to how IR test-collections are created in with pooling. In a test run, it took a participant fifteen minutes to annotate the results of six queries (24 visualizations).

We assigned each participant in the study to six conditions/queries, at random without replacement, from the queries with the least number of assignments. It follows that a participant was very likely to be assigned to multiple datasets and chart- and query types. Due to the drop-out of participants after assigning queries to a participant, queries did not have the same number of participants.

Visualization types. In the evaluation *line*, *bar*, and *vertical bar* charts were included, since these are the types that are supported by StatLine, and thus can be compared.

Visualization Tools. Both OVERLOOK, and StatLine were included in the evaluation. StatLine provided one visualization for a condition; for OVERLOOK the top-three results were evaluated.

Datasets. We selected datasets from CBS for either being recent, or important[1]. After selection, each dataset was manually checked to validate that *most* of its visualizations in StatLine were working. From an initial selection of twenty datasets[2], four datasets were excluded: Two datasets were excluded because they did not have *any* working visualizations in StatLine and two were excluded because OVERLOOK did not support the query in the meta-data (time not formatted according to the standard format and query not in conjunctive normal form, respectively). Out of the sixteen selected datasets (listed in the table in Appendix 2), three had one broken visualization when viewed using StatLine.

We did not test the selected datasets in OVERLOOK before evaluation. This procedure was designed to ensure that selected datasets from CBS were valid, without introducing a bias *against* StatLine by tuning after the evaluation set was known.

Query type. Only OVERLOOK has different query types. This condition was added to approximate the change of selection that StatLine implicitly applies when choosing a different visualization type after having selected data. These query types alter the default data selection provided by CBS. For line charts,

[1] Indicated by being included on the overview page at `https://www.cbs.nl/nl-nl/cijfers`, retrieved on 2019-05-23.
[2] Of which two important datasets were included in the preliminary evaluation.

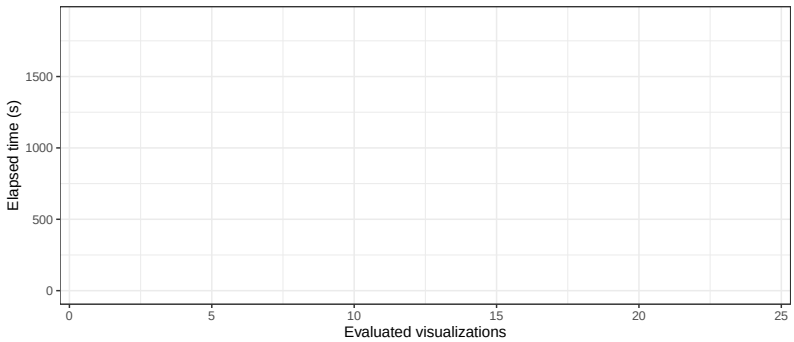


Figure 6.1: Participant progress over time.

the default query was used, as well as a query which finds the *most common* unit of time and selects all values of that unit (e.g., all years, all quarters). For (horizontal) bar charts, the default query was used as well as a variation that selected the *most recent* period.

Participants. We recruited sixteen participants, including 11 (former) graduate students and 13 participants with a background in a technical[3] field. All participants had prior data visualization experience (9: 0–2 years, 4: 2–5 years, 3: 5+ years). Almost all participants (14) had experience using Excel, eight with more specialized tools (*Tableau*, *D3*, *Vega-Lite*), nine used data-oriented work-flows (GIS, data science, open data, *Jupyter*, *R*, . . .), and six had experience with machine learning. On average participants agreed that they had data visualization experience[4].

When inviting potential participants to the user study, we informed them that they would be evaluating the quality of visualizations from a VizRec system and that participating would take fifteen to twenty minutes. Participants did not receive any compensation. The median duration of participation was 22 minutes, with two participants taking more than 30 minutes. During the experiment, each participant evaluated between 19 and 24 visualizations. Figure 6.1 provides an overview of the progress of participants over time.

Study protocol. At the start of the session, potential participants received the information sheet describing the study and were asked to read it (Appendix 7.4). In addition to providing the information sheet, the researcher verbally summarized the study protocol (including the right to withdraw consent) and data usage. After this step, potential participants were asked for consent and asked to sign a consent form, followed by a demographic questionnaire (Appendices 7.4 and 7.4).

When a potential participant agreed to participate, the researcher provided them with a printed copy of the evaluation instructions (Appendix 7.4). Using the figure on the evaluation instructions, participants were verbally introduced to the evaluation interface, and the questions it contained. Before starting the tasks, it was verbally repeated that evaluating the visualizations takes fifteen to twenty minutes when making steady progress and that it would be followed by two verbal exit-interview questions[5].

After this introduction, the participant started by evaluating the first of their visualizations, using the evaluation interface shown in Appendix 7.4. A think-aloud protocol was not used during the experiment due to the high number of visualizations a participant needs to assess during the study. When a participant had questions during the experiment, questions about the evaluation questions were answered. Questions that considered the systems under test, or specific visualizations, were acknowledged and answered once the participation was finished.

[3] I.e., Science, Technology, Engineering, and Mathematics.

[4] Self assessment on a scale of one through seven, mean: 5.5, first quartile: 3, median: 5, third quartile: 6.

[5] *"What did stand out in the visualizations?"* and *"The system used rules to guide visualizations. What design rules do you think were incorporated?"*.

	Ease of understanding							Relevance			
	−3	−2	−1	0	1	2	3	0	1	2	3
All (n = 381)											
n	119	55	29	34	59	48	37	138	92	84	67
%	31.2	14.4	7.6	8.9	15.5	12.6	9.7	36.2	24.1	22.0	17.6
Inter-rater (n = 233)											
n	63	41	16	25	30	24	34	81	51	53	37
%	28.4	18.5	7.21	11.3	13.5	10.8	10.4	36.5	23.0	23.9	16.7

Table 6.1: Distribution of ratings.

All sessions were held in a laboratory setting. Participants used the evaluation interface in a full-screen browser window (Safari 12.1.1) on a 15" retina MacBook Pro running OS X 10.14.5 with a resolution of 2880 × 1800 pixels (effective resolution: 1440 × 900).

6.2 ANALYSIS

Sixteen participants were recruited for the user study[6]. Fifteen participants evaluated six queries each. One participant experienced in visualization evaluated twelve queries over two separate sessions in order to allow us to evaluate inter-rater reliability on these queries.

The data were preprocessed before analysis. The levels for the ease of understanding (EOU) metric were centered around 0, with −3: strongly disagree and 3: strongly agree. The levels for relevance range from 0 . . . 3 with "not applicable" mapped to 0, with 0: irrelevant and 3: highly relevant.

Inter-rater reliability. During the experiment, 266 unique visualizations were evaluated, with participants creating a total of 381 assessments. Out of the 266 visualizations 159 were evaluated once, 99 were evaluated twice, and 8 were evaluated three times.

Both EOU and relevance were measured on an ordinal scale. Since each participant only evaluated a part of the collection, and there is no evaluation shared between all participants, Fleiss' kappa could not be used[7]. Instead, Krippendorff's alpha (α) was used since it can be used for all common scales of measurement and in the presence of missing/sparse data [Kri04]. Alpha is defined as $Agreement = 1 - \frac{D_o}{D_e} = 1 - \frac{\text{Observed Disagreement}}{\text{Expected disagreement}}$[8]. All alpha values were calculated using the *irr* package in R.

Krippendorff [Kri04] argues that the level of acceptable agreement differs per application, but that for scholarly usage in content analysis $\alpha \geq 0.800$ could be a threshold, and an agreement of $\alpha \geq 0.667$ allows for tentative conclusions. In IR literature, Damessie et al. [Dam+17] compared multiple types of judgments to a gold standard. They found an agreement of $\alpha = 0.687$ for judgments created in a laboratory study, $\alpha = 0.407$ for crowd-sourced judgments where participants were paid after judging a complete topic, and $\alpha = 0.561$ where crowd-sourced assessors were paid per document. Schaer [Sch12] found a substantially lower average alpha of 0.145 for relevance assessments created by undergraduate students.

The distribution of the ratings, as shown in Table 6.1, indicate that Krippendorff's alpha can be used. None of the levels is scarce, which prevents substantial changes in Krippendorff's alpha caused by disagreement on an infrequent level. Furthermore, we can see that the distribution of ratings of the items with inter-rater overlap, on which the inter-rater agreement is measured, is similar to the overall distribution of ratings.

[6] The author did not participate in the user study as a particpant.
[7] Maximum overlap between two participants eight visualizations.
[8] With D_o and D_e being functions that calculate disagreement using a different metric function for each scale of measurement.

			Krippendorff's alpha	
Overlap	n	Raters	Ease of understanding	Relevance
All ↔ all	246	16	0.357	0.377
All ↔ expert	23	6	0.483	0.769

Table 6.2: Inter-rater reliability.

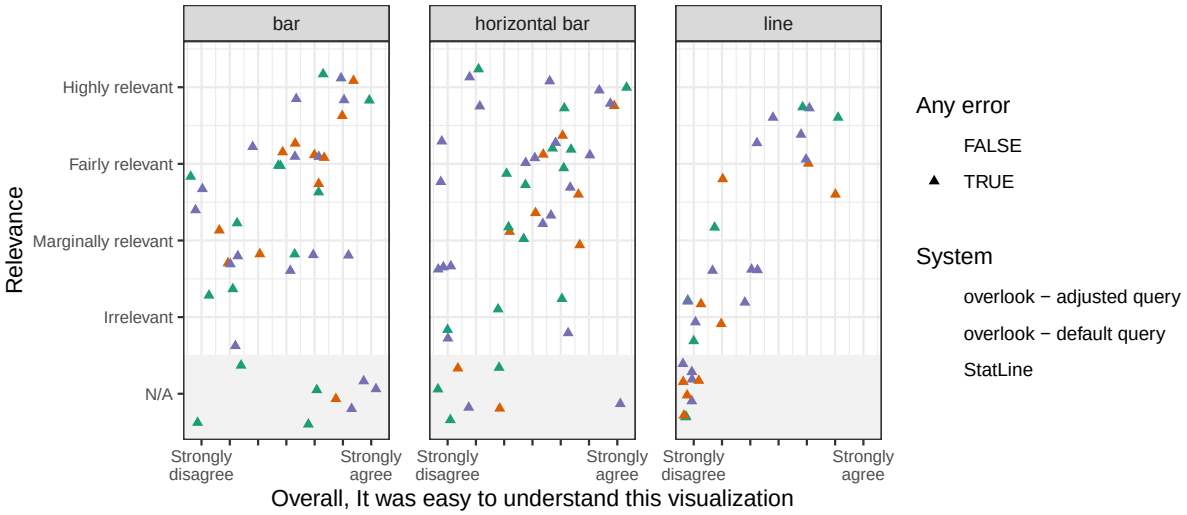


Figure 6.2: Ease of understanding and relevance by chart type and system — top-ranked visualization per system.

Inter-rater agreement was low. This indicates that evaluating visualizations is a hard task for annotators. As Table 6.2 shows the inter-rater agreement between all participants is lower than the threshold used in content analysis and close to that found by Damessie et al. What is interesting in this data is that the inter-agreement between a visualization expert and all other participants is higher ($\alpha = 0.483$ for EOU, $\alpha = 0.769$ for relevance), even though the participants were relatively experienced with visualization, especially compared to the general public.

Quantitative results. Having investigated the reliability of the ratings, we will now move on to investigate the ratings themselves. We will first present an overview of the data, followed by the precision at differing relevance thresholds. Finally, the number of situations in which overlook performs better than StatLine is analyzed.

To give an overview, the ratings for the top-ranked visualizations for all queries are shown in Figure 6.2. This figure is quite revealing in several ways. First of all, it can be seen that the quality of visualizations is highly variable: it is highly dependent on the dataset and the existence of errors. Second, it shows that, per the evaluation instructions, only visualizations with errors were rated with a relevance rating of not applicable. Finally, it shows that a visualization that shows irrelevant information can be easy to understand.

For both systems, the majority of the visualizations had one or more errors in how they were displayed, with more errors in visualizations generated by overlook (77.8 %) than those of StatLine (52.0 %). Table 6.3 presents the summary statistics for the ratings. From this data, we can see that for all variations of overlook the mean (for EOU and relevance) is greater than the median, the distributions are positively skewed. As Table 6.4 shows, the Shapiro-Wilk test rejects the null hypotheses ($p < 0.001$), there is evidence that EOU and relevance ratings for any combination of system and chart type are not normally distributed. Therefore, it is not possible to test for significant differences between systems (per chart type) using unpaired two-samples t-test.

			Any error		Visualization error		Ease of understanding			Relevance		
System	Query type	n	n	%	n	%	$\bar{x}$	s	M	$\bar{x}$	s	M
bar												
StatLine	default	36	17	47.2	8	22.2	0.8	1.8	1	1.8	1.1	2
overlook	default	45	39	86.7	14	31.1	−0.7	1.7	−1	1.3	0.9	1
overlook	most recent	57	45	78.9	23	40.4	−0.9	2.0	−2	1.1	1.1	1
horizontal bar												
StatLine	default	38	21	55.3	9	23.7	0.4	2.1	1	1.8	1.0	2
overlook	default	39	29	74.4	18	46.2	−0.1	2.2	0	1.3	1.1	1
overlook	most recent	63	49	77.8	24	38.1	−0.8	2.1	−2	1.0	1.1	1
line												
StatLine	default	28	15	53.6	13	46.4	0.3	2.4	1	1.6	1.2	2
overlook	default	45	34	75.6	18	40.0	−1.9	1.8	−3	0.7	1.0	0
overlook	most common	30	21	70.0	14	46.7	−2.0	1.8	−3	0.6	1.1	0

Table 6.3: Scores over all visualizations.

		Relevance		Ease of understanding	
System	Query type	w	p	w	p
bar					
StatLine	default	0.850	< 0.001	0.879	0.001
overlook	default	0.877	< 0.001	0.915	0.003
overlook	most recent	0.841	< 0.001	0.856	< 0.001
horizontal bar					
StatLine	default	0.860	< 0.001	0.885	0.001
overlook	default	0.866	< 0.001	0.880	0.001
overlook	most recent	0.782	< 0.001	0.856	< 0.001
line					
StatLine	default	0.835	< 0.001	0.853	0.001
overlook	default	0.712	< 0.001	0.665	< 0.001
overlook	most common	0.600	< 0.001	0.600	< 0.001

Table 6.4: Shapiro-Wilk test for normality, h_0: sample came from a normally distributed population.

However, information retrieval style metrics are commonly used in this situation. In the field of IR, the generalized non-binary precision is defined as $P = \sum_{d \in R} r(d)/n$, for documents d from a set of results R, with $r(d)$ looking up the relevance score for a document [KJ02].

The average of the precision[9] for all datasets, for both EOU and recall, was calculated for varying relevance thresholds. The threshold t was used to map the ratings $r(d) \geq t$ to 1, 0 otherwise. Figures 6.3 and 6.4 present the precision for the top-ranked visualizations, and Figures 6.5 and 6.6 for all visualizations. From these figures, it is apparent that the top-ranked result for overlook performs better than the average of the top 3. The full precision results, shown in the tables in Appendix 7.4, indicate that the performance of overlook is close to that of StatLine. There is no apparent difference in precision between the default- and adjusted queries.

When comparing between overlook and StatLine, the results are mixed. For some datasets and query methods, the former performs better, for some the latter. The top-ranked result from overlook has an equal or higher rating than StatLine for 56 out of 102 situations (*chart type* × *query type*) for EOU and 59 out of 102 for relevance. To distinguish between the different query types, these were split out. From the data in Table 6.5, it is apparent that the default query performs better than the adjusted query.

[9] Note that this differs from the average precision used in IR, which is the average of the precision at the position of each relevant document.

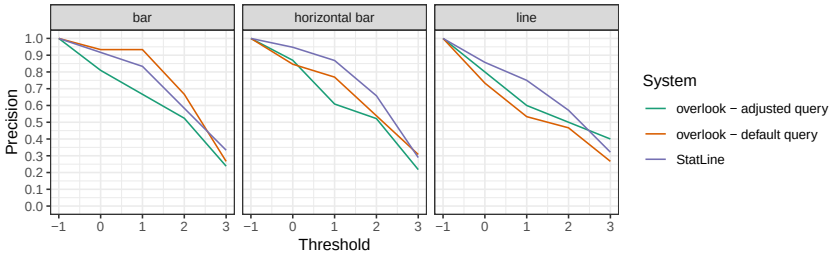


Figure 6.3: Precision on relevance for top-ranked visualizations.

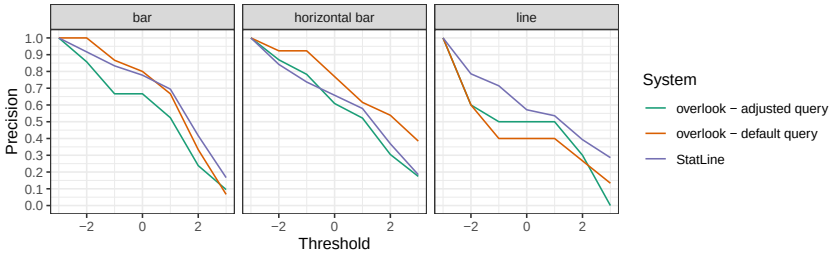


Figure 6.4: Precision on ease of understanding for top-ranked visualizations.

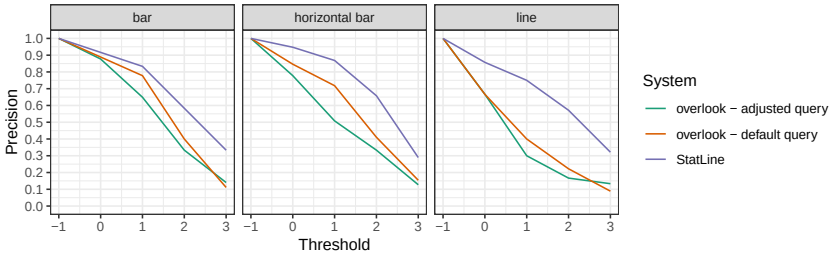


Figure 6.5: Precision on relevance for all visualizations.

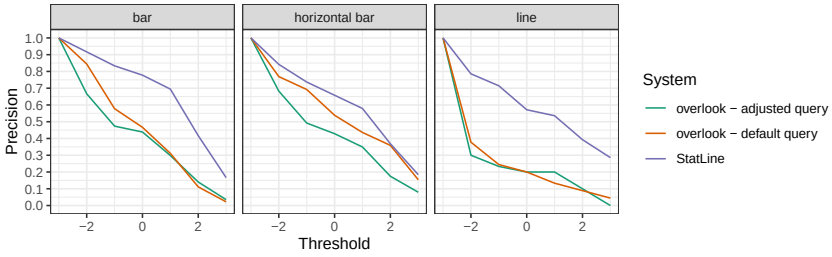


Figure 6.6: Precision on ease of understanding for all visualizations.

		Relevance				Ease of understanding			
		=		≥		=		≥	
Query type	n	n	%	n	%	n	%	n	%
bar									
default	15	3	20.000	10	66.667	5	33.333	9	60.000
most recent	21	7	33.333	12	57.143	7	33.333	11	52.381
horizontal bar									
default	15	5	33.333	9	60.000	2	13.333	10	66.667
most recent	23	8	34.783	13	56.522	6	26.087	15	65.217
line									
default	15	3	20.000	8	53.333	3	20.000	7	46.667
most common	13	2	15.385	6	46.154	2	15.385	4	30.769

Table 6.5: Comparison of per-item ratings for OVERLOOK compared to StatLine.

Participant feedback. After rating the visualizations, participants were asked what stood out in the visualizations and asked to speculate about the design rules used by the system. The majority of the feedback (15 participants) was given in Dutch; therefore, the included quotations have been translated. Statements indicating similar themes were grouped, leading to six themes about the visualizations and five on the rules used by the visualization system.

The most common view among participants, noted by nine participants, was that there were issues in how labels were displayed for visualizations and that this made visualizations hard to understand. Five participants noted that there was a low number of useful visualizations. As one interviewee put it: "Hardly anything works!". Two participants noted that they had trouble assessing the relevance of the information displayed in a visualization without knowing the dataset. Three participants noted the lack of usage of colors by OVERLOOK: "It is a fan of the color blue.". When it comes to the difference between systems, five participants explicitly indicated that StatLine was much clearer, specifying that it made better use of colors, had better labels, and clear visualizations in general. Finally, five participants commented on visualization types, and in specific trellis plots, which are a matrix of scatter/line plots.

Trellis plots were confusing/unknown to one participant, while two (more experienced) participants valued them. One participant indicated that trellis plots were familiar, but that they were complex to interpret if you do not know what you are looking for.

When it comes to feedback on the rules used by the system, six participants could not make guesses for or provide examples of rules or heuristics used by the system. Three participants noted that the system distinguishes between categorical and quantitative fields or dependent and independent fields in the dataset. Four participants noted that there were sequential results were axis were swapped (e.g., x with y), showing one first, followed by another variation. One participant noted that sometimes, first, a single bar chart is shown, and then later a time-series. Finally, one individual indicated that they did not expect the visualizations to have been generated by an automated system.

In this thesis, we investigated visualization recommendation systems, with the high-level goal of supporting data exploration. More specifically, we examined how these systems perform in a real-world setting. We designed OVERLOOK, a system that uses heuristic criteria as optimization criteria; designed an evaluation methodology for sets of visualizations from a VizRec system; and evaluated OVERLOOK on real-world datasets, comparing it to a deployed system which uses curated visualizations.

The final chapter of this thesis has four parts. The first section reviews the most important findings of this thesis. The next section moves on to reflect the limitations of this work. The third section proposes future work. The final section concludes this thesis.

7.1 RESEARCH QUESTIONS

The first research question of this study was to identify what models are used in the implementation of visualization (recommendation) systems. To answer this research question, we performed a literature review, which led to a list of concerns addressed in other systems. In turn, we used these concerns as design criteria for OVERLOOK.

As mentioned in the literature review in Chapter 2, there is a consensus on the models for *visualizations*, but the implementation of VizRec systems differs. Visualization systems commonly use the model of visualization by Bertin, with a growing number of systems using an implementation of a grammar of graphics. VizRec systems commonly distinguish variable types and scale of measurement and use this in determining encodings, use models of perceptual effectiveness to search for effectiveness visualizations, and are implemented using search algorithms or constraint programming techniques.

The second question in this study was to create a design/architecture for a VizRec system that accounts for design variation and supports soft heuristics. This design, introduced in Chapter 3, incorporated the concerns found in the literature review and views VizRec as a constraint optimization problem. This constraint optimization problem consists of hard and soft objectives (for heuristics), the solution of which is a Pareto front of visualizations that are ranked by their objective scores. This design was implemented in OVERLOOK using the Z3 SMT solver.

The third question in this study was how the value to users of visualizations in the set of results of a VizRec system for a given query can be evaluated. We developed (Chapter 4) data collection instruments based on questionnaires used in literature and evaluated (Chapter 5) the instruments by performing a preliminary user study. Based on the results of this user study, we created a new iteration of the instruments. Our instruments evaluate visualizations on how easy it was to understand the visualization, the relevance of the data shown, and component tasks the visualization is suitable for.

Our final research question was to evaluate how OVERLOOK compares to the visualizations of StatLine. As described in Chapter 6, a set of 16 StatLine datasets was selected to evaluate the systems in a user study. Because StatLine adjusts the data selection when switching chart type, a similar mechanism was implemented in OVERLOOK, leading to two query types per type of chart. For each of these datasets, the *bar*, *horizontal bar*, and *line* chart from StatLine were compared to the three top-ranked visualizations of the same type from OVERLOOK.

We evaluated these visualizations in a user study in a laboratory setting with 16 participants. This study had a within-subject design, where each participant evaluated the visualization from both systems for six combinations of dataset, data selection method, and chart type. In total, 381 assessments were made during the user study.

Based on the data collected in the user study, we first investigated the inter-rater reliability. We concluded that inter-rater reliability was low, which indicates that assessing visualizations is a hard task.

We then continued our analysis by investigating the distribution of the ratings. From this, it became clear that a large number of visualizations, for both overlook and to a lesser extent StatLine had errors. We concluded that the presence of errors influenced the rating of the visualizations and that the questions for EOU and relevance measure different concepts.

Finally, we compared the performance of StatLine and the two variations of overlook. We concluded that the top-ranked result for overlook performs better than the lower-ranked results. When investigating precision, with varying thresholds for relevance, all three systems have similar performance, with no clear improvement by adjusting the query. When comparing the rating for StatLine with overlook for a dataset, the default query performs better than the adjusted query, with overlook performing better or equal to StatLine for most visualizations. The relative performance of overlook was better for the two types of *bar* charts compared to the *line* charts. However, with the small sample size, caution must be applied when interpreting this result.

Most participants indicated that there were issues in how labels were displayed and that this made visualizations hard to understand. A third of the participants commented that a lot of the visualizations by overlook were broken. Finally, five participants explicitly indicated that the visualizations by StatLine were clearer.

While not making a statement to the size of the effect, we conclude that overlook creates visualizations of similar quality to the visualizations by StatLine, with the handling of labels and selection of data as the major limitation of overlook.

7.2 limitations

So far, this chapter has focused on reviewing the findings of this thesis. However, there are several limitations to our study, which can be improved. These can be grouped by limitations in the prototype, in the evaluation, and in the metrics used. The following paragraphs will discuss each of these in turn.

Prototype. During the evaluation, we used a high fidelity prototype (overlook), which resembled a finished product. While a high fidelity prototype is needed to investigate the performance in a realistic setting, limitations of the prototype influence the ratings of visualizations.

The first limitation of this prototype are errors that are not caused by the visualization algorithm. In 39.8 % of visualizations, an error in the data selection was indicated. This fraction can likely be reduced by improving how meta-data from CBS is interpreted. Also, the majority of participants indicated that labels were not displayed correctly. Overlapping or cluttered labels can be improved by displaying them differently or even considering the length of labels during visualization recommendation. An alternative approach is to display the visualizations from StatLine using *Vega-Lite* instead of using a screenshot from StatLine. This change would remove a bias introduced by the different styles of the visualizations.

Another limitation is in the set-up of the evaluation, which introduces a bias towards StatLine. First of all, the datasets were checked for working visualizations in StatLine, but not in overlook before being included. Furthermore, the data-selection between charts may differ slightly. The format of the meta-data, and data selection behavior of StatLine was undocumented. This added complexity in implementation, with query transformations to attempt to select equivalent data. In turn, this led to (subtly) different data selection and thus visualizations of different data.

Evaluation. Our experimental results were also affected by choices in the evaluation setting, where we chose to perform a user study with IR style aspects. Moreover, we chose to approximate a realistic setting which is disadvantageous compared to other studies, which often do not include a user study, or use known datasets.

When comparing to IR evaluation, our evaluation included a low number of topics (datasets), whereas 25–50 topics are common in IR [Voo09]. In visualization studies a substantially lower number of datasets is commonly used, enabling a counterbalanced design.

There is also a potential ordering effect present in the evaluation. First, (ordered) OVERLOOK visualizations for a dataset were shown, followed by the StatLine chart of the data dataset. We recommend that new experiments randomize this order.

Finally, due to the drop-out of participants after assigning visualizations described in Section 6.1, visualizations have differing numbers of evaluations. This non-uniform distribution weighs the average scores towards the visualizations where this occurs.

Metrics. The evaluation compared two systems, where one has a single result, and the other has multiple results. This led to difficulties compared to standard IR evaluation protocols. For example, Text REtrieval Conference (TREC) tracks commonly define the evaluation metrics used, a collection of documents, a set of topics to be tested, and a relevance assessment process. The evaluation process commonly collects the top documents from participating systems for each topic, and assessors evaluate the top 100–200 documents for each system. Pooling all documents from multiple systems gives an estimate of the best documents in the collection, which can be used in evaluation metrics. In contrast, there is no standard approach and metric for evaluating sets of visualizations leading to a number of novel choices.

The first issues we encountered was that during the final evaluation, the time between displaying the visualization, and the first interaction with the evaluation form was not measured due to a bug. This type of time measurement is commonly used to measure efficiency — one of the three aspects of usability [Kel09, p.118]. Another limitation is in the limited number of topics (datasets), and documents (visualizations) tested. This is a result of both time constraints and the laboratory setting and was a conscious decision given that we wished to perform exit interviews.

Other limitations were external to the system and are pervasive in IR experiments. Models of information-seeking behavior assume that the relevance of documents decreases as searcher knowledge increases. This process occurs when evaluating multiple visualizations about one dataset in sequence. Moreover, the evaluation instructions we used did not give instructions on how to handle insights that overlapped with other visualizations. A dataset contains multiple clusters of visualizations with each cluster showing relevant insights, similar to instance-recall (distinct correct answers) in the TREC question answering track.

Finally, since the experiment just evaluated two systems, one of which had a single result, evaluation metrics for ranked retrieval could not be used. Precision at K illustrates this point clearly. For precision at K, for $K > 1$, StatLine's maximum possible score would be $\frac{1}{K}$. With Discounted Cumulative Gain (DCG), any information gain after the first document would introduce a bias against StatLine.

In the hypothetical situation where pooled results would be available from multiple systems, another issue would arise when calculating the normalized Discounted Cumulative Gain (nDCG). Systems (potentially) create distinct visualizations that other systems can not generate[1]. nDCG normalizes by dividing by the ideal DCG, which would contain visualizations that a single system could never produce since the (pooled) collection used for ideal DCG differs from that of the system.

7.3 FUTURE WORK

While results from this study were promising, this research has thrown up many questions in need of further investigation. The areas of further research can be categorized into three categories. These areas are described in the paragraphs that follow.

[1] E.g., visualizations created by different libraries.

Prototype. While OVERLOOK is a working system, it is a proof of concept system that can be expanded upon. The main area of improvement is in the included heuristics. Various errors in visualizations can be expressed as constraints or heuristics. For example, the product of the cardinality of the datasets fields, when mapped to the row and $column$ channel can be constrained, limiting the number of elements of trellis plots.

When these rules interact with the data used for the visualization, it is idiomatic to implement them in python. When this is not required, we propose to follow the approach of Moritz et al., who showed that logical rules can be expressively expressed in ASP programs, and provide a database of design guidelines expressed in ASP programs. ASP programs can be parsed and included in the Z3 model.

In evaluation, OVERLOOK recommended a set of *bar*, *horizontal bar*, or *line* charts. This allowed for comparison to the baseline system. The implementation can be expanded to support additional visualization types, or to recommend multiple visualization types in one query. Visualizations of different types are disjunct; thus, the number of states in the Z3 model scales linearly with the number of visualization types requested.

A natural progression from this is to remove the restriction that fields are only used for a single visual variable and allow the assignment of a field from the dataset to multiple visual variables. When this is supported, a regularization penalty could be added as an optimization goal to ensure efficient allocation of visual variables.

With regard to the information available to the VizRec algorithm, OVERLOOK does not query data sources during recommendation. That implies that (selection specific) summary statistics of datasets are not available. Future research might explore methods for incorporating the shape of the data during recommendation, for example, by sampling collection statistics.

Evaluation. This study has used an evaluation approach positioned between the evaluation of a visualization system and IR evaluation. While this is valuable, when evaluating detailed features of a system, or when a large number of responses is needed, further research might choose another approach. Two approaches are common.

When investigating how users experience a system, other studies often use mixed methods designs with multiple treatments. This approach can evaluate multiple interventions and allows for a balanced design. It is also common to design a task that is similar to IIR, where users need to select or save results out of all results. This data can then be used as training data. A disadvantage of this approach is that it is not feasible to evaluate on multiple datasets since the number of respondents needed scales with the number of datasets.

Another approach is to use crowdsourcing[2] to gather assessments. Crowdsourcing is especially useful when a high number of assessments and or assessors is needed. For example, Kim and Jeffrey Heer use crowdsourcing to measure the error rate of assessors when interpreting visual encodings, for various primitive tasks, and use that to rank encodings by visual effectiveness [KH18]. Ç. Demiralp, Bernstein, and Heer evaluate the usage of different judgment types to create an effectiveness ranking for different shapes for shape marks using crowdsourcing.

Ranking. After the SMT problem is solved, OVERLOOK ranks the candidate visualizations by the sum of their heuristic scores. A system that presents results to a user, ordered by a utility function, is using a ranking function [Bur+05]. When labeled examples are available, machine learning can be used to learn a ranking function. The term Learning to Rank (LTR) refers to this application of machine learning.

LTR has been applied in VizRec [Luo+18; Mor+19]. Using this technique, visualization assessments from this study, visualization ratings from other studies (as used by Moritz et al. [Mor+19]), or user

[2] Also called: micro tasks.

behavior[3] can be used to improve the ranking of visualizations and make the system increasingly more effective after initial deployment. For overlook in specific, LTR can be used to learn weights for heuristic scores or to re-rank visualizations based on how visualizations use fields (e.g., number of values on the x-axis).

Another ranking technique to be considered is using heuristic scores to include diversity in the ranked results. For example, a heuristic can express the difference in encoding between a clustered bar chart and a stacked bar chart.

Finally, further research might investigate the usage of machine learning to learn the primitive tasks a visualization is suitable for. This classifier can then be used as either decision support, where a system suggests what tasks a visualization is suitable for; or as a ranking problem, where a system returns a list of visualizations ranked for the task(s) given.

7.4 conclusion

This study set out to investigate the usage of Visualization Recommendation (VizRec) system in a realistic setting. To achieve this goal, we implemented overlook, a system that views VizRec as a constraint optimization problem, with design guidelines and heuristics implemented as hard- and soft criteria. Solving this problem results in a Pareto front of visualizations that are each optimal for one of the soft criteria.

We evaluated the system in a user study where participants compared *bar*, *horizontal bar*, and *line* chart from overlook to the same type of chart from StatLine for 16 datasets. This evaluation performs a more challenging comparison (cross-system) in a more difficult setting (real datasets) than in other works, which commonly use synthetic or known datasets in a mixed-method design to evaluate features of a system. We contribute data collection instruments for VizRec systems in a realistic setting and show that evaluating visualizations is a hard task for annotators.

This study has shown that overlook can be used in a realistic setting and did not find a difference in visualization quality between overlook and the baseline system. This demonstrates that visualizations generated by overlook are an alternative to manually created visualizations and that the system has the potential to support data analysts by generating visualizations based on design guidelines. In contrast to the baseline system, overlook can generate visualizations during data exploration, closing the loop, and making data exploration more interactive.

The evidence from this study suggests that overlook can be deployed in a real-world setting, increasing the value of data by supporting non-experts during data exploration.

[3] E.g., implicit feedback such as interacting with a visualization, or explicit feedback such as saving, rating, or bookmarking a visualization.

Appendices

INFORMATION SHEET

You are being invited to participate in a research study titled overlook. This study is being done by Ties de Kock from the Faculty of Electrical Engineering, Mathematics and Computer Science at the University of Twente.

You are being invited for this study since you are likely to have some experience with data visualization, and likely been exposed to information from similar types of datasets as the datasets from Statistics Netherlands (CBS) used in this study. You are free to discuss both your choice to participate in this study, as well as the content seen during the study with anyone. You can take your time to reflect on your choice of whether you participate in this study. If you have any questions about the study or the process, I will try to answer your questions to the best of my ability. You can ask additional questions at any time, both before and after participating.

The purpose of this research study is to evaluate the quality of results from an automated visualization system. It will take you approximately 15 minutes to complete. During your participation you will evaluate the quality of visualizations for multiple datasets from CBS in order to provide a ground truth for the quality of the results of the system, using a questionnaire for each visualization. You will evaluate multiple visualizations per dataset.

Before the study starts you will be asked to answer a survey that will be used to provide a summary of the demographics of the participants. After the study ends, the researcher will ask two questions about the results you have seen. This exit interview is optional.

Participation in the study is voluntary and you have the right to refuse to participate, or withdraw from the study at any time, both before the study starts, as well while, and after participating. Please note that, as is customary, any information you provide before withdrawing from the study, and that has already been processed before withdrawing, may still be used.

There a no risks involved in this study. Participating in the study does not provide direct benefits. No financial compensation will be offered for participation.

The study does not collect personally identifiable information (except for the name and signature on the consent form). The consent forms cannot be directly linked to the pseudonyms used during the study, and consent forms are confidential.

Responses on the demographic survey are anonymous and will be used in aggregate to describe the study population. Questionnaire responses (multiple choice) during the study are stored anonymously and will be used to evaluate the quality of the visualization systems. (Anonymous) quotes from the exit-interview, written down as notes by the researcher, may be used in the report.

The results of the study will be used in a master thesis that is due to be finished in this academic year. The results and/or data gathered during this study, may be used for future research and learning.

Participation is voluntary and you have the right to withdraw. Data you provide, from the moment of consent, up to the moment of withdrawal (of consent), can be used in research. You have the right to view, and/or alter data provided during your participation.

If you want more information about this study, now or in the future, you can contact Ties de Kock, e-mail: ties@tiesdekock.nl.

If you have questions about your rights as a research participant, or wish to obtain information, ask questions, or discuss any concerns about this study with someone other than the researcher(s), please contact the Secretary of the Ethics Committee of Faculty of Electrical Engineering, Mathematics and Computer Science at the University of Twente, mrs J Rebel-de Boer by e-mail: ethics-comm-ewi@utwente.nl.

overlook – information sheet, 12-06-2019

CONSENT FORM

Consent Form for overlook

YOU WILL BE GIVEN A COPY OF THIS INFORMED CONSENT FORM

Please tick the appropriate boxes	**Yes**	**No**
Taking part in the study		
I have read and understood the study information dated 12-06-2019, or it has been read to me. I have been able to ask questions about the study and my questions have been answered to my satisfaction.	□	□
I consent voluntarily to be a participant in this study and understand that I can refuse to answer questions and I can withdraw from the study at any time, without having to give a reason.	□	□
I understand that information I provide will be used for a master thesis, and possibly in other publications.	□	□
I understand that personal information collected about me that can identify me, such as my name, will not be shared beyond the study team.	□	□
I agree that my information can be quoted in research outputs.	□	□
I give permission for the data I provide (e.g. questionnaires rating visualizations) to be archived so it can be used for future research and learning.	□	□

Signatures

……..	……	……
Name of participant	Signature	Date

I have handed out the information sheet to the potential participant and, to the best of my ability, ensured that the participant understands to what they are freely consenting.

Ties de Kock	……	……
Researcher name	Signature	Date

Study contact details for further information: Ties de Kock, e-mail: t.dekock@student.utwente.nl

If you have questions about your rights as a research participant, or wish to obtain information, ask questions, or discuss any concerns about this study with someone other than the researcher(s), please contact the Secretary of the Ethics Committee of Faculty of Electrical Engineering, Mathematics and Computer Science at the University of Twente, mrs J Rebel-de Boer by e-mail: ethics-comm-ewi@utwente.nl.

DEMOGRAPHIC QUESTIONNAIRE

Participant

Participation id	
Date	
Time	

Personalia[1]

Age	○ **15-25** ○ **25-50** ○ **50-65** ○ **65+**
Education (ongoing or finished)	○ Science, technology, engineering, and mathematics ○ Other ○ Havo, vwo ○ Hbo-, wo-bachelor ○ Wo-master, doctor ○ Unknown/none of the above/decline to answer

I consider myself to have experience with (multiple answers possible):

- ○ Creating charts in Excel
- ○ More specialized visualization tools (tableau, D3, vega-lite, …)
- ○ Data oriented workflows (GIS, data science, open data, Jupyter, R, …)
- ○ Machine learning

I have experience with data visualization:

Strongly disagree						Strongly agree
○	○	○	○	○	○	○

Years of experience:

- ○ 0-2
- ○ 2-5
- ○ 5+

Visualization is a(n) …:

- ○ Technology
- ○ Art
- ○ Science

[1] This information will be used for descriptive statistics of the population of participants, if wanted questions can be left unanswered.

EVALUATION INSTRUCTIONS

Evaluation instructions

During the experiment you will evaluate multiple visualizations for their effectiveness by answering multiple questions. This is done using a web interface that shows the title of the visualization, the visualization, and a questionnaire. The user interface is shown in the figure below.

During the experiment you will evaluate visualiations for six combinations of datasets and visualization. For each combination, up to four visualizations are shown.

The answers for the questions are introduced on the next page. This document can be kept as a reference while answering the questions. In addition, feel free to ask additional questions now or as the experiment progresses.

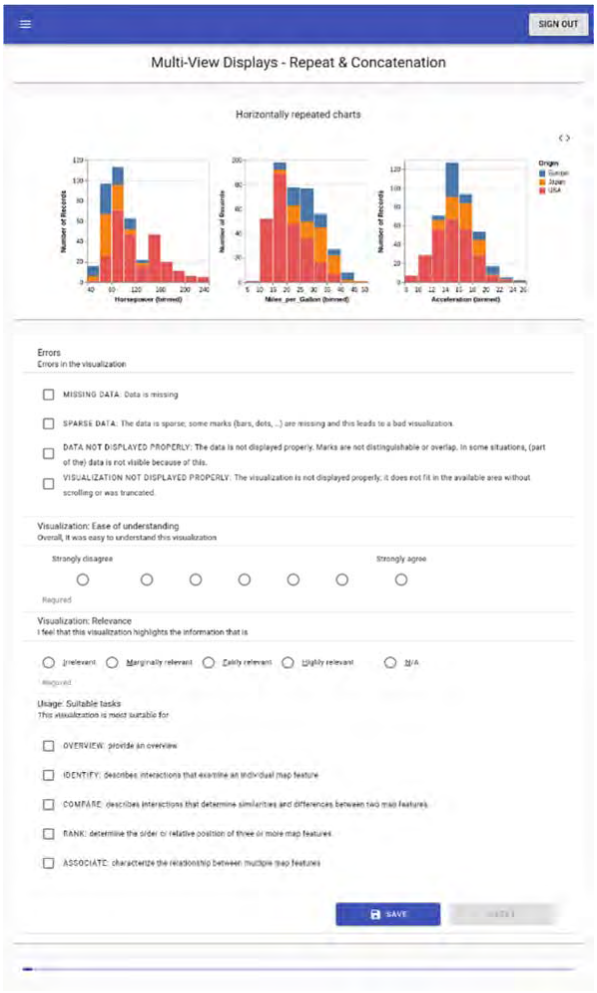


Figure 1: layout of the evaluation form

Questions

Errors

The first question investigates potential *errors* in the visualization.

It consists of checkboxes that for different types of errors, two on the data in the visualization, and two on how it is displayed (error in the visualization itself, or in how a potentially valid visualization is displayed in the tool). Please tick any error type that is applicable.

Ease of understanding

The second question investigates how hard it was to understand the visualization.

Was the visualization clear? Did you understand how it was structured?

Relevance

The third question measures the relevance of the visualization. A relevant visualization from a dataset should show a pattern or trend; a chart showing a single bar would not be relevant.

A visualization that *gives insight* into [some] information that would help when writing report on [its title/topic] is relevant.

When a visualization is broken and you can not understand it, answer N/A.

Suitable tasks

The final question is on the task the visualization is suitable for. Examples for the tasks are given below.

Answer these questions with your best estimate and check the types that apply, multiple options can be checked.

overview

The visualization gives a general overview of the data in the dataset.

identify

"*What is the mileage per gallon of the Audi TT?*"

"*What comedies have won awards?*"

compare

"*Are there exceptions to the relationship between horsepower and acceleration?*"

"*What is the car with the highest MPG?*"

rank

"*Rank the cereals by calories*"

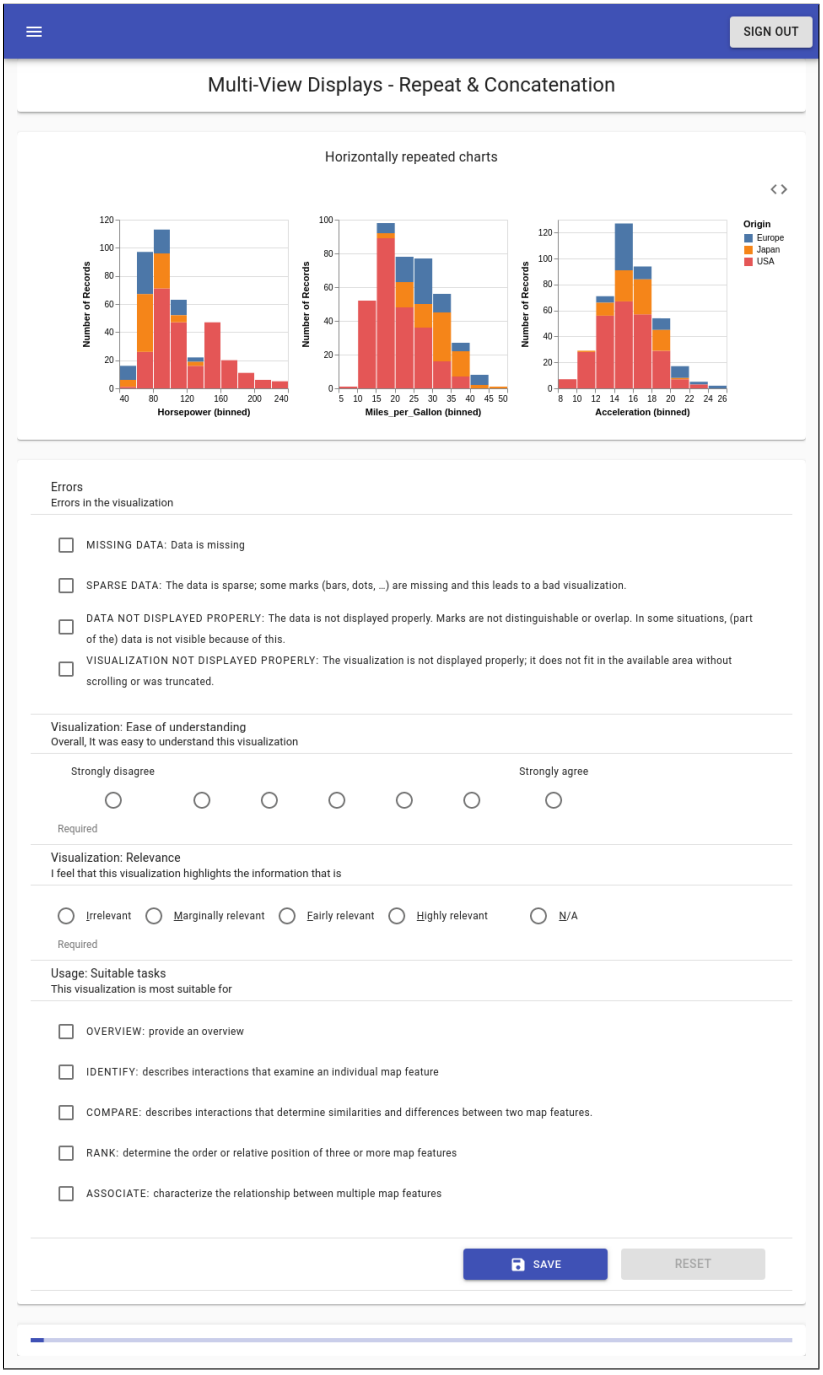
"*What actresses are in the data set?*"

"*What is the distribution of carbohydrates in cereals?*"

associate

"*Is there a trend of increasing film length over the years?*"

"*Are there groups of cereals with similar fat/calories/sugar?*"


SIGN OUT
Multi-View Displays - Repeat & Concatenation
Horizontally repeated charts
Number of Records
Horsepower (binned)
Miles_per_Gallon (binned)
Acceleration (binned)
Origin
Europe
Japan
USA
Errors
Errors in the visualization
MISSING DATA: Data is missing
SPARSE DATA: The data is sparse; some marks (bars, dots, ...) are missing and this leads to a bad visualization.
DATA NOT DISPLAYED PROPERLY: The data is not displayed properly. Marks are not distinguishable or overlap. In some situations, (part of the) data is not visible because of this.
VISUALIZATION NOT DISPLAYED PROPERLY: The visualization is not displayed properly; it does not fit in the available area without scrolling or was truncated.
Visualization: Ease of understanding
Overall, It was easy to understand this visualization
Strongly disagree
Strongly agree
Required
Visualization: Relevance
I feel that this visualization highlights the information that is
Irrelevant
Marginally relevant
Fairly relevant
Highly relevant
N/A
Required
Usage: Suitable tasks
This visualization is most suitable for
OVERVIEW: provide an overview
IDENTIFY: describes interactions that examine an individual map feature
COMPARE: describes interactions that determine similarities and differences between two map features.
RANK: determine the order or relative position of three or more map features
ASSOCIATE: characterize the relationship between multiple map features
SAVE
RESET

SELECTED DATASETS

Dataset		Chart type		
Title	ID	line	bar	hor. bar
Overheid; ontvangen belastingen	82569NED	✓	✓	✓
Consumentenprijzen; prijsindex 2015=100	83131NED	✓	✓	✓
Consumentenprijzen; basisjaren vanaf 1969	83136NED	✓	✓	✓
Consumentenprijzen; werknemers laag, 1969-1995	83433NED	✓	✓	✓
Beloning en arbeidsvolume van werknemers	82577NED	✓	✓	✓
Bevolking; generatie, geslacht, leeftijd en migratieachtergrond	37325	✓	✓	✓

Table 1: Selected datasets for preliminary evaluation.

SELECTED DATASETS

Dataset		Chart type		
Title	ID	line	bar	hor. bar
Faillissementen, zittingsdaggecorrigeerd	83085NED	✓	✓	✓
Bouwvergunningen woonruimten; aantal en index	83668NED	✓	✓	✓
Vacatures; vacaturegraad naar SBI 2008	80567ned	✓	✓	✓
Bevolking; hoogstbehaald onderwijsniveau en onderwijsrichting	82816ned	✓	✓	✓
Invoer en uitvoer volgens eigendomsoverdracht; volumeontwikkelingen	84264NED		✓	✓
Uitzendbureaus en arbeidsbemiddeling; ontwikkeling omzet, 2015=100	83853ned	✓	✓	✓
Zorguitgaven internationaal vergelijkbaar; functies en financiering	84043ned	✓	✓	✓
Basisverzekering (Zwv); kosten per persoon, inkomen	81827ned	✓	✓	✓
Sociale zekerheid; kerncijfers, uitkeringen naar uitkeringssoort	37789ksz	✓	✓	✓
Arbeidsdeelname; kerncijfers	82309ned	✓	✓	✓
Bedrijfsgegevens; omzetontwikkeling (stijgers-dalers), SBI 2008	82190ned	✓	✓	✓
Huishoudens; grootte, samenstelling, positie in het huishouden, 1 januari	82905ned	✓	✓	✓
Totale reizigerskilometers in Nederland per jaar; vervoerwijzen, regio's	83497ned	✓	✓	✓
Jaarmutatie consumentenprijsndex; vanaf 1963	70936ned		✓	✓
Consumentenprijzen; prijsindex 2015=100	83131ned		✓	✓
Beloning en arbeidsvolume van werknemers; kwartalen, nr, 1995–2018	82577NED	✓	✓	✓

Table 2: Selected datasets for user study.

PRECISION

			Ease of understanding							Relevance				
System	Query type	n	−3	−2	−1	0	1	2	3	*N/A*	0	1	2	3
bar														
StatLine	default	36	1	0.92	0.83	0.78	0.69	0.42	0.17	1	0.92	0.83	0.58	0.33
overlook	default	45	1	0.84	0.58	0.47	0.31	0.11	0.02	1	0.89	0.78	0.40	0.11
overlook	most recent	57	1	0.67	0.47	0.44	0.30	0.14	0.04	1	0.88	0.65	0.33	0.14
horizontal bar														
StatLine	default	38	1	0.84	0.74	0.66	0.58	0.37	0.18	1	0.95	0.87	0.66	0.29
overlook	default	39	1	0.77	0.69	0.54	0.44	0.36	0.15	1	0.85	0.72	0.41	0.15
overlook	most recent	63	1	0.68	0.49	0.43	0.35	0.17	0.08	1	0.78	0.51	0.33	0.13
line														
StatLine	default	28	1	0.79	0.71	0.57	0.54	0.39	0.29	1	0.86	0.75	0.57	0.32
overlook	default	45	1	0.38	0.24	0.20	0.13	0.09	0.04	1	0.67	0.40	0.22	0.09
overlook	most common	30	1	0.30	0.23	0.20	0.20	0.10	0.00	1	0.67	0.30	0.17	0.13

Table 3: Mapped precision (rating ≥ indicated value = 1.) for all visualizations

			Ease of understanding							Relevance				
System	Query type	n	−3	−2	−1	0	1	2	3	*N/A*	0	1	2	3
bar														
StatLine	default	36	1	0.92	0.83	0.78	0.69	0.42	0.17	1	0.92	0.83	0.58	0.33
overlook	default	15	1	1.00	0.87	0.80	0.67	0.33	0.07	1	0.93	0.93	0.67	0.27
overlook	most recent	21	1	0.86	0.67	0.67	0.52	0.24	0.10	1	0.81	0.67	0.52	0.24
horizontal bar														
StatLine	default	38	1	0.84	0.74	0.66	0.58	0.37	0.18	1	0.95	0.87	0.66	0.29
overlook	default	13	1	0.92	0.92	0.77	0.62	0.54	0.38	1	0.85	0.77	0.54	0.31
overlook	most recent	23	1	0.87	0.78	0.61	0.52	0.30	0.17	1	0.87	0.61	0.52	0.22
line														
StatLine	default	28	1	0.79	0.71	0.57	0.54	0.39	0.29	1	0.86	0.75	0.57	0.32
overlook	default	15	1	0.60	0.40	0.40	0.40	0.27	0.13	1	0.73	0.53	0.47	0.27
overlook	most common	10	1	0.60	0.50	0.50	0.50	0.30	0.00	1	0.80	0.60	0.50	0.40

Table 4: Mapped precision (rating ≥ indicated value = 1) for top-ranked visualizations.

[AES05] R. Amar, J. Eagan, and J. Stasko. "Low-Level Components of Analytic Activity in Information Visualization". In: *Proceedings of the 2005 IEEE Symposium on Information Visualization (INFOVIS'05)*. INFOVIS '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 15–15. ISBN: 0-7803-9464-X. DOI: 10.1109/INFOVIS.2005.24.

[Ber83] Jacques Bertin. *Semiology of graphics: diagrams, networks, maps*. 1983, p. 456. ISBN: 978-1-58948-261-6.

[BI97] Pia Borlund and Peter Ingwersen. "The development of a method for the evaluation of interactive information retrieval systems". In: *Journal of Documentation* 53.3 (1997), pp. 225–250. DOI: 10.1108/EUM0000000007198.

[BKM08] Aaron Bangor, Philip T. Kortum, and James T. Miller. "An empirical evaluation of the system usability scale". In: *International Journal of Human-Computer Interaction* 24.6 (2008), pp. 574–594. DOI: 10.1080/10447310802205776.

[Bro96] John Brooke. "SUS - A quick and dirty usability scale". In: *Usability evaluation in industry* 189.194 (1996), pp. 4–7.

[Bur+05] Chris Burges et al. "Learning to Rank Using Gradient Descent". In: *Proceedings of the 22Nd International Conference on Machine Learning*. ICML '05. New York, NY, USA: ACM, 2005, pp. 89–96. ISBN: 1-59593-180-5. DOI: 10.1145/1102351.1102363.

[CM84] William S Cleveland and Robert McGill. "Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods". In: *Journal of the American Statistical Association* 79.387 (1984), pp. 531–554.

[Dam+17] Tadele T. Damessie et al. "Gauging the Quality of Relevance Assessments using Inter-Rater Agreement". In: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '17*. New York, New York, USA: ACM Press, 2017, pp. 1089–1092. ISBN: 9781450350228. DOI: 10.1145/3077136.3080729.

[DBH14] Ç Demiralp, M S Bernstein, and J Heer. "Learning Perceptual Kernels for Visualization Design". In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 1933–1942. ISSN: 1077-2626. DOI: 10.1109/TVCG.2014.2346978.

[EEW13] Stef van den Elzen, Stef van den Elzen, and Jarke J. van Wijk. "Small multiples, large singles: A new approach for visual data exploration". In: *Computer Graphics Forum* 32.3 PART2 (2013), pp. 191–200. DOI: 10.1111/cgf.12106.

[Hor99] Eric Horvitz. "Principles of mixed-initiative user interfaces". In: *Proceedings of the SIGCHI conference on Human factors in computing systems the CHI is the limit - CHI '99* (1999), pp. 159–166. ISSN: 10525505. DOI: 10.1145/302979.303030.

[HS12] Jeffrey Heer and Ben Shneiderman. "Interactive Dynamics for Visual Analysis". In: *Communications of the ACM* 55.4 (2012), pp. 45–54. ISSN: 0001-0782. DOI: 10.1145/2133806.2133821.

[Int98] International Organization for Standardization. *Iso 9241-11*. Tech. rep. 1998, p. 22.

[Kel09] Diane Kelly. "Methods for Evaluating Interactive Information Retrieval Systems with Users". In: *Foundations and Trends in Information Retrieval* 3.1-2 (2009), pp. 1–224. ISSN: 1554-0669. DOI: 10.1561/1500000012.

[KH18] Younghoon Kim and Jeffrey Heer. "Assessing Effects of Task and Data Distribution on the Effectiveness of Visual Encodings". In: *Computer Graphics Forum* 37.3 (2018), pp. 157–167. ISSN: 14678659. DOI: 10.1111/cgf.13409.

[KJ02] Jaana Kekäläinen and Kalervo Järvelin. "Using graded relevance assessments in IR evaluation". In: *Journal of the American Society for Information Science and Technology* 53.13 (2002), pp. 1120–1129. ISSN: 15322882. DOI: 10.1002/asi.10137. URL: http://doi.wiley.com/10.1002/asi.10137.

[Kri04] Klaus Krippendorff. "Reliability in Content Analysis." In: *Human Communication Research* (2004). ISSN: 0360-3989. DOI: 10.1111/j.1468-2958.2004.tb00738.x.

[Lam+12] Heidi Lam et al. "Empirical studies in information visualization: Seven scenarios". In: *IEEE Transactions on Visualization and Computer Graphics* 18.9 (2012), pp. 1520–1536. ISSN: 10772626. DOI: 10.1109/TVCG.2011.279.

[Lew90] James R. Lewis. "Psychometric evaluation of an after-scenario questionnaire for computer usability studies". In: *ACM SIGCHI Bulletin* 23.1 (1990), pp. 78–81. ISSN: 0736-6906. DOI: 10.1145/122672.122692.

[Lew93] James R. Lewis. *IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use*. Tech. rep. Boca Raton, FL: IBM, 1993.

[Luo+18] Yuyu Luo et al. "Deepeye: towards automatic data visualization". In: *Proceedings - IEEE 34th International Conference on Data Engineering, ICDE 2018* (2018), pp. 101–112. DOI: 10.1109/ICDE.2018.00019.

[Mac86] Jock Mackinlay. "Automating the design of graphical presentations of relational information". In: *ACM Transactions on Graphics* 5.2 (1986), pp. 110–141. ISSN: 07300301. DOI: 10.1145/22949.22950.

[MB08] Leonardo de Moura and Nikolaj Bjørner. "Z3: An Efficient SMT Solver". In: *Tools and Algorithms for the Construction and Analysis of Systems*. Ed. by C R Ramakrishnan and Jakob Rehof. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 337–340. ISBN: 978-3-540-78800-3. DOI: 10.1007/978-3-540-78800-3_24.

[Min69] Charles Joseph Minard. *Carte Figurative*. 1869. URL: https://en.wikipedia.org/wiki/File:Minard.png (visited on 03/19/2019).

[Mor+19] Dominik Moritz et al. "Formalizing visualization design knowledge as constraints: Actionable and extensible models in Draco". In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2019), pp. 438–448. ISSN: 19410506. DOI: 10.1109/TVCG.2018.2865240.

[NM90] Jakob Nielsen and Rolf Molich. "Heuristic Evaluation of user interfaces". In: *CHI '90 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* April (1990), pp. 249–256. ISSN: 1942-597X. DOI: 10.1145/97243.97281.

[RM90] Steven F. Roth and Joe Mattis. "Data characterization for intelligent graphics presentation". In: *Proceedings of the SIGCHI conference on Human factors in computing systems Empowering people - CHI '90*. 1990, pp. 193–200. DOI: 10.1145/97243.97273.

[Rot+94] Steven F. Roth et al. "Interactive graphic design using automatic presentation knowledge". In: *Proceedings of the SIGCHI conference on Human factors in computing systems celebrating interdependence - CHI '94* April (1994), pp. 112–117. DOI: 10.1145/191666.191719.

[Rot13] Robert E Roth. "An empirically-derived taxonomy of interaction primitives for interactive cartography and geovisualization". In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (2013), pp. 2356–2365. ISSN: 10772626. DOI: 10.1109/TVCG.2013.130.

[Sat+17] Arvind Satyanarayan et al. "Vega-Lite: A Grammar of Interactive Graphics". In: *IEEE Transactions on Visualization and Computer Graphics* 23.1 (2017), pp. 341–350. ISSN: 10772626. DOI: 10.1109/TVCG.2016.2599030.

[Sch12] Philipp Schaer. "Better than Their Reputation? On the Reliability of Relevance Assessments with Students". In: *Information Access Evaluation. Multilinguality, Multimodality, and Visual Analytics*. Ed. by Tiziana Catarci et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 124–135. ISBN: 978-3-642-33247-0.

[SD09] Jeff Sauro and Joseph S. Dumas. "Comparison of three one-question, post-task usability questionnaires". In: *Proceedings of the 27th international conference on Human factors in computing systems - CHI 09*. 2009, p. 1599. ISBN: 9781605582467. DOI: 10.1145/1518701.1518946.

[SED18] Bahador Saket, Alex Endert, and Cagatay Demiralp. "Task-Based Effectiveness of Basic Visualizations". In: *IEEE Transactions on Visualization and Computer Graphics* 14.8 (2018). ISSN: 10772626. DOI: 10.1109/TVCG.2018.2829750. arXiv: 1709.08546.

[Shn96] Ben Shneiderman. "The eyes have it: a task by data type taxonomy for information visualizations". In: *Proceedings 1996 IEEE Symposium on Visual Languages* (1996), pp. 336–343. ISSN: 1049-2615. DOI: 10.1109/VL.1996.545307.

[Sid+16] Tarique Siddiqui et al. "Effortless Data Exploration with zenvisage: An Expressive and Interactive Visual Analytics System". In: *Proceedings of the VLDB Endowment* 10.4 (2016), pp. 457–468. ISSN: 21508097. DOI: 10.14778/3025111.3025126. arXiv: 1604.03583.

[Ste46] S.S. S Stevens. "On the Theory of Scales of Measurement". In: *Science* 103.2684 (1946), pp. 677–680. ISSN: 0036-8075. DOI: 10.1126/science.103.2684.677.

[STH02] Chris Stolte, Diane Tang, and Pat Hanrahan. "Polaris: a system for query, analysis, and visualization of multidimensional relational databases". In: *IEEE Transactions on Visualization and Computer Graphics* 8.1 (2002), pp. 1–14. ISSN: 10772626. DOI: 10.1109/2945.981851.

[TT06] Donna P Tedesco and Thomas S Tullis. "A Comparison of Methods for Eliciting Post-Task Subjective Ratings in Usability Testing". In: *Usability professionals association annual conference (UPA)*. 2006, pp. 1–9.

[Tuf01] Edward Rolf Tufte. *The visual display of quantitative information*. 2nd ed. Cheshire: Graphics Press, 2001, p. 197. ISBN: 9780961392147.

[Van05] Jarke J. Van Wijk. "The value of visualization". In: *Proceedings of the IEEE Visualization Conference*. Vol. 00. 2005, p. 11. ISBN: 0780394623. DOI: 10.1109/VIS.2005.102.

[Var+15] Manasi Vartak et al. "SeeDB : Efficient Data-Driven Visualization Recommendations to Support Visual Analytics". In: *Proceedings of the VLDB Endowment* 8.13 (2015), pp. 2182–2193. ISSN: 21508097. DOI: 10.14778/2831360.2831371. arXiv: 15334406.

[Var+17] Manasi Vartak et al. "Towards Visualization Recommendation Systems". In: *ACM SIGMOD Record* 45.4 (2017), pp. 34–39. ISSN: 01635808. DOI: 10.1145/3092931.3092937.

[Voo09] Ellen M Voorhees. "Topic Set Size Redux". In: *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '09. New York, NY, USA: ACM, 2009, pp. 806–807. ISBN: 978-1-60558-483-6. DOI: 10.1145/1571941.1572138.

[Wic10] Hadley Wickham. "A Layered Grammar of Graphics". In: *Journal of Computational and Graphical Statistics* 19.1 (2010), pp. 3–28. ISSN: 1061-8600. DOI: 10.1198/jcgs.2009.07098.

[Wil05] Leland Wilkinson. *The Grammar of Graphics (Statistics and Computing)*. 2nd. Berlin, Heidelberg: Springer-Verlag, 2005. ISBN: 0-387-24544-8.

[WL90] Stephen Wehrend and Clayton Lewis. "A Problem-oriented Classification of Visualization Techniques". In: *Proceedings of the 1st Conference on Visualization '90*. VIS '90. Los Alamitos, CA, USA: IEEE Computer Society Press, 1990, pp. 139–143. ISBN: 0-8186-2083-8.

[Won+16a] Kanit Wongsuphasawat et al. “Towards a general-purpose query language for visualization recommendation”. In: *Proceedings of the Workshop on Human-In-the-Loop Data Analytics - HILDA '16* (2016), pp. 1–6. DOI: 10.1145/2939502.2939506.

[Won+16b] Kanit Wongsuphasawat et al. “Voyager: Exploratory Analysis via Faceted Browsing of Visualization Recommendations”. In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (2016), pp. 649–658. ISSN: 10772626. DOI: 10.1109/TVCG.2015.2467191.

[Won+17] Kanit Wongsuphasawat et al. “Voyager 2”. In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17*. 2017, pp. 2648–2659. ISBN: 9781450346559. DOI: 10.1145/3025453.3025768.