Institute for Visualization and Interactive Systems

University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Bachelorarbeit

Evaluating Linking and Brushing Techniques in Aggregated Views

Patrick Dreesmann

Course of Study:	Data Science
Examiner:	Dr. Steffen Koch
Supervisor:	Samuel Beck, M.Sc., Franziska Huth, M.Sc.
Commenced:	November 9, 2022
Completed:	May 9, 2023

Abstract

Mobile devices are used more and more in the context of information visualisation. They differ from common desktop devices in many ways, especially the display sizes and ways of interaction are anything but similar. These differences bring a lot of advantages with them, but also some challenges that need to be overcome, that were not there before. The way visualisations are designed needs to adapt. Those challenges arise especially for visualisation systems that need a lot of screen space.

Multiple coordinated views are such visualisation systems because they consist of at least two visualisations and have to allow interactions with those visualisations. This bachelor's thesis aims to conquer those hardships by proposing an approach for adapting multiple coordinated views for the use on mobile devices. The suggested strategy includes the use of an interactive *main visualisation* with a *linked* micro-visualisation. The user of the visualisation system can interact with the *main visualisation* by *brushing* data points which are then highlighted via *linking* in the micro-visualisation. The application of a micro-visualisation makes it possible to use most of the screen space for the main visualisation which is important for the performance of the *brushing and linking* interactions. Additionally, the micro-visualisation shows an aggregated view of the data to save even more screen space. In the context of this work, three prototypes are developed that realise the described approach.

To evaluate the approach, one online user study per prototype was conducted to first test the general understandability of the *linking* and then investigate the influence of the height of the micro-visualisation on the accuracy and completion time for tasks completed with the help of the prototypes. The results show that the approach is understandable for the users. Furthermore, it is discovered that the completion time does not depend on the height of the micro-visualisation. Finally, the studies revealed, that the accuracy increases with the height of the micro-visualisation but the effect decreases once a certain height is reached. Ideally, a compromise is made to keep the accuracy as high as possible and the height as small as possible.

Kurzfassung

Mobile Endgeräte werden immer häufiger im Kontext der Informationsvisualisierung eingesetzt. Sie unterscheiden sich in vielerlei Hinsicht von gängigen Desktop-Geräten, insbesondere die Display-Größen und Interaktionsmöglichkeiten sind alles andere als ähnlich. Diese Unterschiede bringen viele Vorteile mit sich, aber auch einige Herausforderungen, die überwunden werden müssen und die zuvor nicht vorhanden waren. Die Art und Weise, wie Visualisierungen gestaltet werden, muss angepasst werden. Diese Herausforderungen treten insbesondere bei Visualisierungssystemen auf, die viel Bildschirmplatz benötigen.

Multiple koordinierte Ansichten sind solche Visualisierungssysteme, weil sie aus mindestens zwei Visualisierungen bestehen und Interaktionen mit diesen Visualisierungen ermöglichen müssen. Diese Bachelorarbeit zielt darauf ab, diese Schwierigkeiten zu überwinden, indem sie einen Ansatz zur Anpassung von multiplen koordinierten Ansichten für die Verwendung auf mobilen Geräten vorschlägt. Die vorgeschlagene Strategie umfasst die Verwendung einer interaktiven Hauptvisualisierung mit einer verlinkten Mikro-Visualisierung. Die Nutzer*Innen der Visualisierung kann mit der Hauptvisualisierung interagieren, indem Datenpunkte per*Brushing* ausgewählt werden, die dann über *Linking* in der Mikro-Visualisierung hervorgehoben werden. Die Verwendung einer Mikro-Visualisierung ermöglicht es, den größten Teil des Bildschirms für die Hauptvisualisierung zu verwenden, was für die Performanz der *Brushing und Linking* Interaktion wichtig ist. Zusätzlich zeigt die Mikro-Visualisierung eine aggregierte Ansicht der Daten, um weitere Bildschirmfläche zu sparen. Im Rahmen dieser Arbeit wurden drei Prototypen entwickelt, die den beschriebenen Ansatz realisieren.

Um den Ansatz zu bewerten, wurde pro Prototyp eine Online-Benutzerstudie durchgeführt, um zunächst die allgemeine Verständlichkeit zu testen und dann den Einfluss der Höhe der Mikro-Visualisierung auf die Genauigkeit und Bearbeitungszeit für mit Hilfe der Prototypen durchgeführte Aufgaben zu untersuchen. Die Ergebnisse zeigen, dass der Ansatz für die Benutzer verständlich ist. Darüber hinaus wird entdeckt, dass die Bearbeitungszeit nicht von der Höhe der Mikro-Visualisierung abhängt. Schließlich zeigen die Benuterstudien, dass die Genauigkeit mit der Höhe der Mikro-Visualisierung zunimmt, der Effekt jedoch nicht mehr so stark ist, sobald eine bestimmte Höhe erreicht ist. Idealerweise wird ein Kompromiss gefunden, um die Genauigkeit so hoch wie möglich und die Höhe so klein wie möglich zu halten.

Contents

List of Figures

List of Tables

List of Listings

1 Introduction

This chapter will shortly motivate the bachelor's thesis and formulate its goals. Additionally a structure of the work is given.

1.1 Motivation

Multiple views [Rob07] are a common approach in information visualisation in order to depict different variables of the same dataset at the same time by showing multiple visualisations in one layout. This technique can help to get deeper insights into the data and can also uncover hidden connections and correlations. The effect can even be enhanced by coordinating these views using the interaction technique *brushing and linking* [BMMS91]. With *brushing and linking*, the user "brushes" data points inside of one visualisation. This means the user interacts with the visualisation and selects a subset of the displayed data. The same subset of data is then highlighted in the other visualisation; the visualisations are "linked".

Another trend that emerged and grew quickly in the last couple of years is the usage of mobile devices, e.g., smartphones and tablets, for information visualisation [BNW+16]. To display such multiple views on a mobile device is challenging because the available screen space is limited. However, there are use cases in which the user might not want to switch to a desktop device with a bigger screen but instead use multiple views on a mobile device. For example, data gathered by fitness trackers is already being sent to the mobile phone. Additionally, desktop devices are usually not available during or directly after a workout. Those fitness trackers usually record multiple kinds of data, e.g., the heart rate, the pulse, GPS data of the covered route, burned calories, blood oxygen level, running speed, workout time and others. Those are not only many different variables but also different data types, i.e., temporal, geospatial and quantitative data. In this particular scenario, it would be beneficial to utilise multiple views on a mobile device to visually analyse the data.

A possible approach to tackle the challenge of scarce screen space is the usage of micro-visualisations. Since they are small in size, they are ideal for mobile devices and small displays in general. This work proposes to use one brushable *main visualisation* next to one or more *linked* micro-visualisations. The main visualisation is displayed in a larger size in order to keep the performance of the *brushing* interaction as high as possible. The micro-visualisation takes up the remaining screen space. One possible approach is to save screen space by showing aggregations of the dataset in micro-visualisations next to a main visualisation.

1.2 Goals

The goal of this thesis is to design and implement different prototypes for *brushing and linking* using micro-visualisations on mobile devices. These prototypes are subsequently evaluated in a user study. The prototypes differ in the used visualisation techniques, both for the main visualisations and the micro-visualisations. The choice is based on the visualised data.

The main visualisations are a timeline for temporal data, a choropleth map for geospatial data and a scatterplot for quantitative data. The chosen micro-visualisations are a histogram for quantitative data and a bar chart for nominal/ordinal data. The *brushing* for the scatterplot is a straight-forward rectangular brush. The approach for the timeline is similar, but the brush is only allowed on the temporal variable. The brush inside the choropleth map is realised by a simple clicking/tapping selection. The brushed data points are highlighted in the *main visualisation* as well as in the micro-visualisation by the same colour. This is very intuitive because this way the data points are perceived as two different groups: brushed and unbrushed.

The prototypes are afterwards evaluated in an online user study. First of all the general understandability of the implemented visualisations and the *brushing and linking* design is tested. The focus of the study however lies on the effect that the micro-visualisation height has on the completion time and accuracy when using those visualisations.

1.3 Thesis Structure

The bachelor's thesis follows the succeeding structure:

Chapter 2 - Fundamentals The second chapter presents the technical foundation needed for understanding and following this thesis. This foundation includes an introduction to visualisation techniques, micro-visualisations, *Gestalt Laws*, *brushing and linking* and interaction techniques.

Chapter 3 - Related Work Chapter 3 presents and summarises related work. It focuses on similar dashboards and visualisation systems as well as *brushing and linking* on mobile devices.

Chapter 4 - Conceptual Design of Prototypes In this chapter, the general design choices that were made for the prototypes are described and explained. The purpose is to give the reader a general idea of the nature of the prototypes.

Chapter 5 - Prototype Implementation The fifth chapter gives a detailed explanation of the prototype implementation. First, the used project structure, frameworks and libraries are introduced briefly. Afterwards, all aspects of the implementation are described in detail.

Chapter 6 - Evaluation Chapter 6 is about evaluating the prototypes implemented in Chapter 5. The evaluation is split into three studies, one for each prototype. The goal is to test the general understandability and the effect of the height of the micro-visualisation on the accuracy and completion time.

Chapter 7 - Conclusion The last chapter wraps up this bachelor's thesis by summarising this work's results and the gained insights. Lastly, some potential future work is suggested.

2 Fundamentals

This chapter gives a technical background of the terms and concepts that are important for the bachelor's thesis. In the beginning, Section 2.1 introduces the *Gestalt Laws*, a general concept in perception theory. Section 2.2 introduces the concept of interaction in information visualisation and especially *brushing and linking* interactions. Thereafter, Section 2.3 introduces a visualisation technique for geospatial data called *choropleth map*. Section 2.4 explains the term *micro-visualisation*. Afterwards, Section 2.5 describes the important aspects to look out for when creating visualisations for mobile devices.

2.1 Gestalt Laws

The *Gestalt Laws* of perception by Wertheimer [Wer23] describe how the human brain perceptually organises visual information into meaningful patterns and structures. The original six principles are shown in Figure 2.1 and explained in the following paragraphs.

Law of Proximity The Law of Proximity states, that objects that are close to each other are perceived as a group. For example, the points in the scatterplot shown in Figure 2.1a are perceived as two groups because they consist of two clusters that are far apart.

Law of Similarity The Law of similarity declares, that objects that are similar in appearance are perceived as a group. The points in Figure 2.1b are again perceived as two groups but in this scenario, the reason is the two different colours. One group is made up of the blue points and one group is composed of the orange points.

Law of Good Gestalt The Law of Good Gestalt articulates, that humans tend to perceive objects or elements as a unified, complete, and organised whole, rather than a collection of individual parts or pieces. This is demonstrated in Figure 2.1c. The left picture shows the original object as seen by the viewer: Two straight lines that have the same colour but cross each other. The picture in the middle shows how a typical observer perceives those two lines, i.e., as two straight lines as stated above. Usually, those lines will not be perceived as the two right angles demonstrated in the right picture.

Law of Good Continuation The law of good continuation specifies, that objects that are aligned or appear to be connected are perceived as a group. The dashed line in Figure 2.1d is perceived as such and not as entirely separate lines. The pattern is quite obvious.

Law of Closure The law of closure expresses, that the brain tends to complete figures that are not fully visible. The typical viewer understands that the object in Figure 2.1e is a circle that is not fully closed.

Law of Common Fate The Law of Common Fate says, that objects that move together are perceived as a group. Figure 2.1f shows six steps of an animation to illustrate this law. The user perceives two groups of two points each, one group is rotating clockwise and the other group is rotating counter-clockwise. Overall, the *Gestalt laws* provide valuable insights into how the human brain processes visual information and offer a useful framework for designing visually appealing and user-friendly interfaces and graphics.

2.2 Interaction Techniques

Yi et al. define interaction techniques as “the features that provide users with the ability to directly or indirectly manipulate and interpret representations” [YKSJ07]. They additionally reviewed a lot of interactive visualisation systems and came up with the seven categories of interaction techniques [YKSJ07] that are briefly described in the following paragraphs. Per category, one example is provided in Figure 2.2.

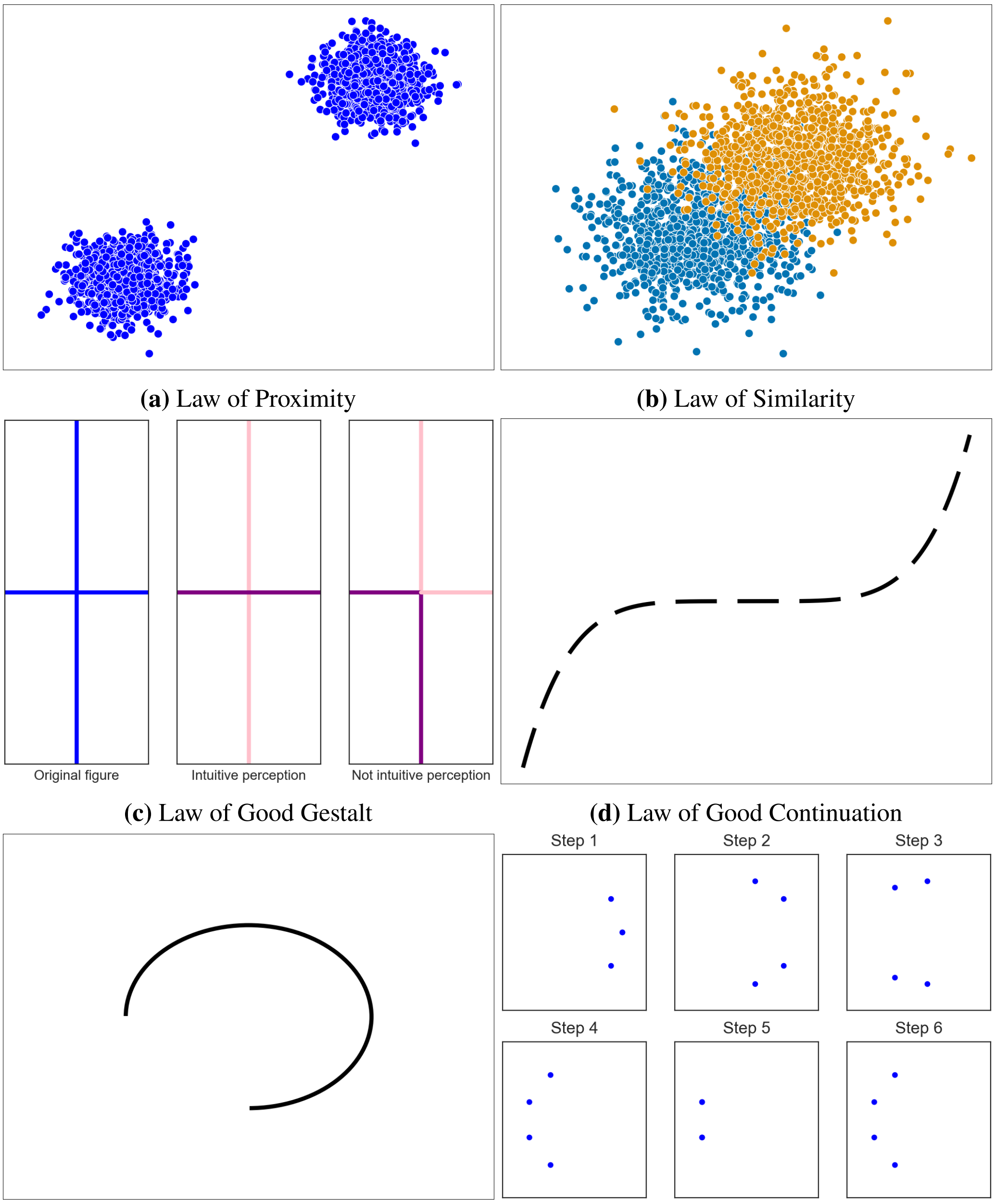
Select Select interactions allow users to highlight data points. This highlighting should help the user keep an eye on those specific data points, also under the condition of a changing visualisation. The example in Figure 2.2a shows one bar chart before and after sorting. The user selected one bar before sorting. This selection is illustrated by changing the bar’s colour. This way the user can easily find the selected bar after sorting.

Explore Explore interactions allow users to view a different subset of the underlying dataset. The most common example of this kind of interaction is panning. Figure 2.2b shows a map before and after a panning interaction.

Reconfigure Reconfigure describes a category of interaction techniques that let the user change the placement of data points, e.g., by sorting data points by size or colour as demonstrated in Figure 2.2c.

Encode Encode interactions enable the user to change the mapping of the data variables to the visual variables, e.g., change the colour or size of data items. Figure 2.2d depicts an encode interaction. The colours in a bar chart are changed based on the height of the bars. Bars with a value smaller than five turned blue and the other bars turned red.

Abstract / Elaborate Those kinds of interaction techniques allow the user to change the level of detail for the visualisation, either increasing or decreasing the number of details visible. Good examples of this category are tooltips and zooming. To illustrate this Figure 2.2e shows a map before and after zooming out.



(a) Law of Proximity

(b) Law of Similarity

(c) Law of Good Gestalt

(d) Law of Good Continuation

(e) Law of Closure

(f) Law of Common Fate

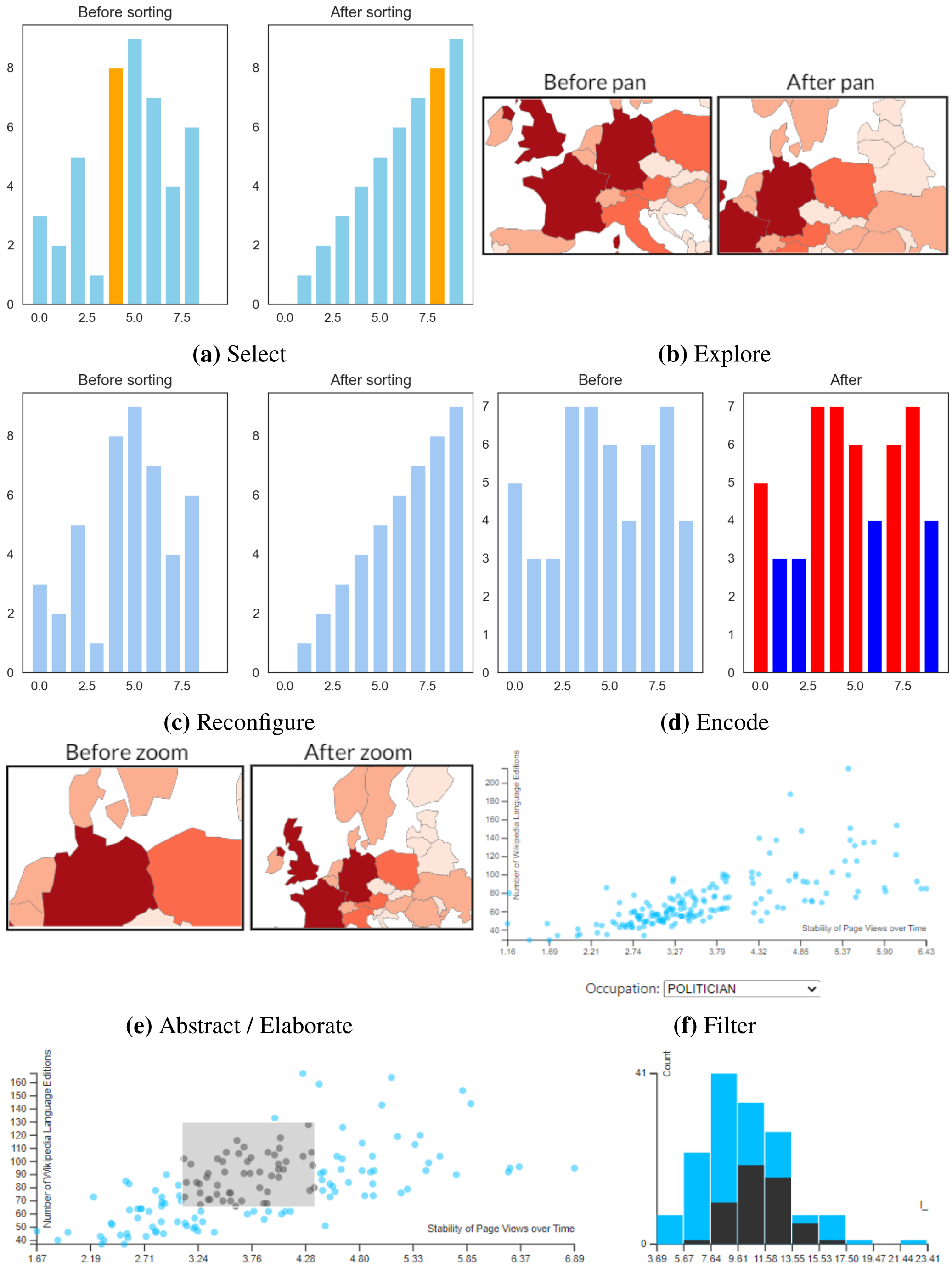
Figure 2.1: Examples for the original Gestalt Laws by Wertheimer [Wer23]

Filter Filter interaction techniques permit the user to show data points based on conditions, e.g., only show people that follow a certain profession or are born within a certain time period. As an example, Figure 2.2f shows a scatterplot in which the user has the possibility to filter for the occupation variable via a drop-down menu.

Connect Connect interactions are used to highlight relationships between visual items. The best example for this are *brushing and linking* interactions which are explained below in Subsection 2.2.1. Figure 2.2g demonstrates such an interaction.

2.2.1 Brushing and Linking

The interaction technique *brushing and linking* [BMMS91], which is at the centre of this bachelor's thesis, clearly belongs to the *Connect* category. It is a powerful interaction technique, that involves the *linking* of multiple visualisations, such that the selection of one data point in one visualisation is *linked* to the selection of corresponding data points in other visualisations. The selected data points are highlighted in all *linked* visualisations, allowing users to see the relationships between different parts of the data. One of the key benefits of *brushing and linking* is that it allows users to explore and analyse data in multiple ways. For example, users can select a data point in one visualisation and see how it is related to other data points in different visualisations. They can also compare the selected data points across different visualisations to see how they vary with respect to different variables or dimensions. Another benefit of *brushing and linking* is that it supports the human cognitive process of pattern recognition. By highlighting the selected data points in multiple visualisations, *brushing and linking* makes it easier for users to see patterns and relationships in the data that might otherwise be difficult to detect.



(a) Select

(b) Explore

(c) Reconfigure

(d) Encode

Occupation: POLITICIAN

(e) Abstract / Elaborate

(f) Filter

(g) Connect

Figure 2.2: Examples for the interaction technique categories by Yi et al. [YKSJ07]

2.3 Choropleth Map

A *choropleth map* is a type of information visualisation that is used to represent geographical data. The data is displayed on a map, with regions shaded or patterned based on the value of a particular attribute [ES92]. The goal of a *choropleth map* is to provide a high-level view of the geographical distribution of data, making it easier to understand patterns, trends, and relationships in the data. In a *choropleth map*, geographical regions are usually divided into predefined areas, such as states, counties, or postal codes. The data attribute of interest is then assigned to each area, and the regions are shaded or patterned according to the data value. A colour scale is typically used to represent the data values, with different colours representing different ranges of data values. *choropleth maps* are particularly useful for displaying data that has a geographical component, such as population density, income, or political affiliation. The maps provide a clear visual representation of the data and can help users to identify areas of high and low data values, as well as trends and patterns in the data. An example of a choropleth map is illustrated in Figure 2.3.

2.4 Micro-Visualisations

There is no clear definition for the term *micro-visualisation*, moreover, there are even multiple terms that describe the same or very similar concepts. Those concepts evolved and changed over time until today. This chapter aims at finding the understanding or definition of the concept of *micro-visualisations* that best suits this bachelor's thesis.

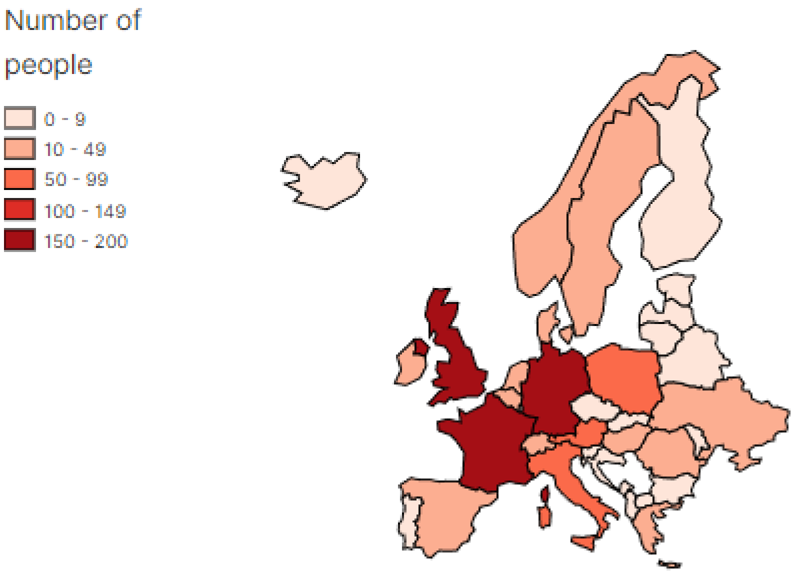
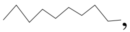
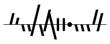
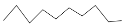


Figure 2.3: Choropleth map of Europe showing the number of famous people born in each country

It all started with Edward Tufte's introduction of *Sparklines* , which he defines as "small intense, simple, word-sized graphic[s] with typographic resolution" [Tuf06]. A lot of research in this direction concentrates on reforming Tufte's concept. Many different adaptions of the *Sparkline* exist.
Brandes et al. [BNRS13] created the *Gestaltline*  which refer to a specific type of micro-visualisation that is based on the Gestalt Laws. In the context of micro-visualisation, Gestaltlines aim to leverage these laws to create simple, meaningful representations of data that are easily interpreted by the human brain.
The definition of the term *Sparkline* as introduced by Tufte is generally restricted to refer to word-sized line charts  or bar charts . However, Beck and Weiskopf's [BW17] interpretation of the term *word-sized graphics* encompasses a broader definition, including all forms of word-sized visualisations. These word-sized graphics can be integrated into various media platforms such as text and tables.

Goffin et al. [GBWI17] take this one step further and define the term *word-scale graphics*, by this they ease the limitations of the size of the visualisation. They specifically state that a *word-scale graphic* is "larger than the size of a letter, but smaller than a sentence or paragraph" [GBWI17].

Great parts of the research concerning micro-visualisations concentrated on the placement of the graphic, which is irrelevant to the context of this work and therefore excluded. Since we do not want to embed our visualisations into text, the size of our graphics does not need to be word-oriented, only small enough to fit on the screen of a mobile device next to a larger main visualisation. Additionally, the micro-visualisation is not supposed to display the complete data set, but an aggregation of it. This is done in order to significantly decrease the used screen space to leave as much space as possible for the interactive main visualisation. Because of this reasoning, the thesis uses *aggregated views*, micro-visualisations that show an aggregation of the data.

2.5 Visualisation Design for Mobile Devices

When designing a visualisation for the use on mobile devices, other factors come into play which are not at all or not as important as when designing a visualisation for desktop devices. In the bounds of this work, only smartphones and larger devices are considered. The developed prototypes are not usable on smartwatches or similar small devices. Those mentioned mobile devices come in many forms and shapes and sizes, ranging from smartphones to tablets to laptop PCs. Although the intended use case is a mobile device, the usage on a desktop PC should also be possible. Out of the above reasons, the visualisation design and the design of the prototypes needs to be responsive, i.e., the visualisations need to dynamically adapt to the size of the screen and the interaction possibilities that the used device offers [AS17]. Horak et al. [HAB+21] provide detailed guidelines on the responsive design of visualisations.

The second aspect that is important when designing visualisations for mobile devices is the use of interactions. The developer needs to keep in mind that touch interactions behave differently than keyboard and mouse interactions. Also, there is no possibility of using tooltips to provide additional information since there is no mouse. Brehmer et al. [BLST21] go over those differences and explain touch interactions thoroughly.

3 Related Work

This chapter presents related work and discusses how the contributions of this thesis expand on it. Section 3.1 talks about visualisation dashboards and systems. Thereafter, Section 3.2 concentrates on related work about *brushing and linking* on mobile devices.

3.1 Dashboards and Visualisation Systems

This section is about related work concerning dashboards and visualisation systems that are similar to the prototypes developed in the context of this bachelor's thesis. A lot of research has been done in this field, and for this reason, it has to be narrowed down further. This section only includes dashboards and systems that have at least one focus point in common with the prototypes developed by this work. Hence, only systems are introduced, that include micro-visualisations, mobile devices or *brushing and linking* interactions.

Tschinkel et al. [THHS16] created a recommendation dashboard for faceted search, which is an entire research topic on its own. The dashboard consists of one main visualisation and multiple micro-visualisations, one for each filter. Only one main visualisation is displayed at a time but they can be switched. The micro-visualisations representing the filters only appear after the *brushing* interaction and only show the brushed data points, not all of them. This makes the comparison between those groups impossible but it makes sense in the use case of faceted search. Additionally, the dashboard allows filtering the main visualisation using the brushes. Again, this makes sense because their work is tailored for faceted search. The purpose of the micro-visualisations is to save the filtering interactions made with the main visualisation, not to gain new insights into the data. Therefore, the micro-visualisations do not include an aggregation step. Their work's goal was to create a new faceted search experience. At that time the recommender systems were purely text-based, the addition of visualisations was a novelty. In follow-up research Tschinkel [Tsc20] focused on creating new micro-visualisations for three main visualisations (temporal, categorical and spatial). An evaluation was done to compare those micro-visualisations to the textual representations of faceted search filters mentioned before. Furthermore, they adapted their recommending dashboard for the use on mobile devices. They carried out a small heuristic evaluation of the adapted dashboard but they solely focused on the use case of faceted search. The micro-visualisations themselves and especially *brushing and linking* were not the focus of their research.

Another system was presented by Langner et al. [LHD18]. The focus of their work lies on the distribution of their dashboard across multiple mobile devices. The dashboard can be rearranged by rearranging the used devices because they use spatially-aware displays. The goal is to allow easy collaboration. *Brushing and linking* interactions are also possible across the devices but there is no research done in that direction. No micro-visualisations are used. Also, no evaluation of the developed prototype is conducted.

The last system discussed in this section by Sadana and Stasko [SS16] is a dashboard explicitly developed for the iPad. In order to create their app, the authors analysed existing desktop applications and tried to map the functionalities to the use on tablets. No micro-visualisations are included in the app, all visualisations have the same size and are connected via *brushing and linking*. The focus of their work was completely on the creation of dashboards for mobile devices and the challenges that come with that task.

All of the aforementioned systems are somewhat similar to the prototypes that are developed in the context of this thesis but the research presented in the work is focused in other directions. Some systems lack the use of micro-visualisations [LHD18; SS16] and none of them apply *aggregated views*. The recommending dashboard is completely focused on the use case of faceted search, while the work by Sadana and Stasko [SS16] concentrates on the challenges of mobile development. Langner et al. [LHD18] fixate on the collaboration and distribution across devices. In contrast to the work presented above, we focus on *brushing and linking* interactions with aggregated views on mobile devices.

3.2 *Brushing and Linking* on Mobile Devices

This section covers related work about *brushing and linking* interactions on mobile devices. There has not been done a lot of research on this specific topic.

As mentioned in Section 3.1, there is some related work about the development of dashboards and visualisation systems that support *brushing and linking* interaction on mobile devices. Tschinkel [Tsc20] implements those interactions by simply showing a micro-visualisation after the *brushing* interaction that only depicts the brushed data points. They briefly explain their interaction design. However, the main aspect of their work is the creation of micro-visualisations and the evaluation of those.

Langner et al. [LSBD21] developed a visualisation framework that combines mobile devices and head-mounted devices for augmented reality. The framework allows cross-device *brushing and linking* but the entire work is limited to the combination of mobile and AR devices. No additional research in the direction of *brushing and linking* is provided. The work by Flemisch [Fle18] follows a similar approach, but only for mobile devices. Their contribution is the technical realisation of *brushing and linking* interactions across multiple devices. They do not look at the visualisation aspect of the *brushing and linking* interactions.

Finally, Sadana and Stasko [SS16] did some research in the direction of *brushing* interactions on mobile devices with touchscreens. They concentrated on the technical implementation of brushing interactions via touch. They did not focus on the *linking* part of that interaction design. They conclude that the *brushing* should only be implemented via interactions with one finger.

As stated before, research in the direction of *brushing and linking* interactions on mobile devices is scarce. Some work only mentions the possibility of those interactions [LHD18; LSBD21; Tsc20] but focuses on other challengers. Other work looks into the technical realisation of *linking* across devices [Fle18] or the technical realisation of *brushing* via touch interactions [SS16]. Our focus differs from the aforementioned work in that we concentrate on the development and evaluation of a *linking* approach for aggregated views on mobile devices.

4 Conceptual Design of Prototypes

The focus of this chapter is on the conceptual design of the prototypes that were created within the scope of this bachelor's thesis. In the following paragraphs, the choices and thoughts that went into those prototypes are explained. First, a description of the overall layout is given. Afterwards, the choice of visualisations for the prototypes is elucidated. At last, the *brushing and linking* design is presented. Details concerning the implementation of those prototypes are given in Chapter 5.

Overall Layout The main goal of this work is to evaluate *brushing and linking* for *aggregated views*. In this specific scenario, the usage of more than one main visualisation and one *aggregated view* does not bring any additional value, it only reduces the available screen space per visualisation. Because of that, a prototype is made up of one main visualisation and one *aggregated view*.

Since the prototypes are created for mobile devices, a column layout is the most sensible approach. Mobile devices are typically higher than wide. Thus, a column layout leads to smaller aspect ratios than a row layout. Furthermore, the column layout allows the alignment of the horizontal axes of the main visualisation and the *aggregated view*. This makes the interpretation of the *brushing and linking* more intuitive. The *aggregated view* is placed above the main visualisation in all prototypes. A draft of the overall layout for a prototype is provided in Figure 4.1.

Visualisations Another important decision is the choice of visualisations for the prototypes. Multiple factors were taken into consideration here. On the one hand, the prototypes are supposed to be simple and intuitive. Hence, common and easy-to-use visualisations are preferable. On the other hand, the generalisability of the findings of this work increases when the types of the displayed variables are diverse. Therefore, the choice for the three prototypes is the following: Prototype one consists of a scatterplot and a histogram. Prototype two is made up of a timeline and a histogram. Prototype three includes a choropleth map and a bar chart. This collection of visualisations covers most variable types: quantitative data, temporal data, spatial data and nominal data. The composition of the three prototypes can also be found in Table 4.1.

prototype	main visualisation	aggregated view
1	scatterplot	histogram
2	timeline	histogram
3	choropleth map	bar chart

Table 4.1: Main visualisation and *aggregated view* for the prototypes

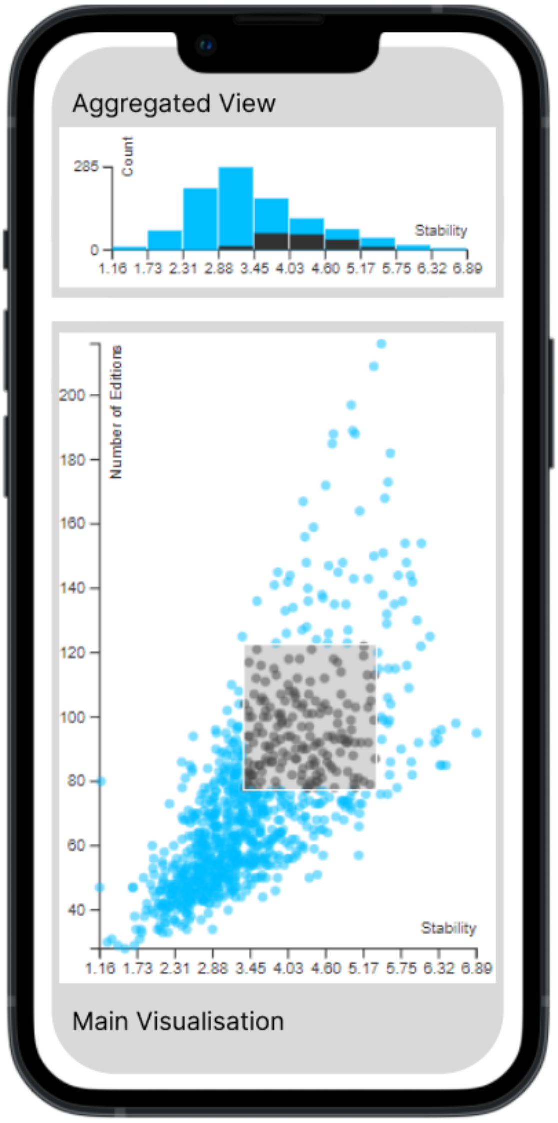


Figure 4.1: Overall layout of a prototype consisting of a *brushed* scatterplot as the main visualisation and a *linked* histogram as the aggregated view

***Brushing and Linking* Design** The last part of the conceptual design is the *brushing and linking* itself. First of all, the *brushing* interaction is limited to the main visualisation because it is quite clear how this direction works. To *link* the *brushing* interactions from the aggregated view to the main visualisation is not that obvious because not all mappings are unambiguous due to the aggregation step in between. For example *brushing* half a bar in a histogram is not unambiguously mappable to a selection inside of a timeline or scatterplot. Further research is needed to investigate which possible solution is most intuitive and works best.

Now the actual *brushing* of data points in the main visualisation and *linking* to data points in the *aggregated view* has to be designed. Within the context of this thesis, this is done by adding a second histogram/bar chart in the *aggregated view* on top of the first one. The second chart only shows the *brushed* data points. This way it uses length as the visual variable which is very accurate for quantitative data [Mac86]. Additionally, the two charts share the same baseline which makes comparisons very easy. Moreover, the *brushed* data points and the *brushed aggregated view* as well as the *unbrushed* data points and the *unbrushed aggregated view* share the same colour. This makes the *linking* between the two visualisations most intuitive because the human mind will automatically perceive two groups (*brushed* and *unbrushed*), as stated by the *Gestalt Law of Similarity* [Wer23]. An example of that approach is illustrated in the draft in Figure 4.1. Some data points in the scatterplot are *brushed* and highlighted in the colour anthracite. The proportions of the bars in the histogram, that belong to the *brushed* data points are also highlighted in the colour anthracite.

The *brushing* interaction is realised through simple mouse / single-finger touch interactions. Both possibilities are implemented to keep the responsiveness alive. For the scatterplot a straight-forward rectangular brush is used: The user can select a rectangle by clicking/tapping and dragging, which means the user can brush on both variables. For the timeline, the brush is limited to the temporal variable, i.e., the user can select a certain time span by clicking/tapping and dragging along the time axis. Finally, for the choropleth map, a simple selection by clicking / tapping is used. A country is selected/unselected by clicking/tapping on it.

5 Prototype Implementation

The three prototypes created for this bachelor's thesis are embedded in a Single-Page Application (SPA) created with the Vue.js [You14] framework in a Node.js application. All visualisations used within the prototypes are implemented using the JavaScript framework D3.js [BOH11] and wrapped in a Vue component. Each prototype is placed inside its own view which is made accessible using Vue Router [YM14]. All data is managed and shared between the views and components with the help of Pinia [Mor19] data stores. It is always a good idea to test visualisation systems with different datasets. In this case, multiple datasets are generated from the *Pantheon* [YRH+16] 2.0[1] dataset artificially by adding the possibility to filter for the profession attribute of the people. A detailed description of the dataset is provided in Section 5.2. This is done through a component that is shared by all views. The component only includes a simple drop-down menu that lets the user choose between all professions that contain more data points than a certain threshold, e.g. more than 50 people. The application is build using Vite [You19], the build tool created for the Vue ecosystem.

This chapter is split up into three sections. Section 5.1 is about the general layout of the prototypes. Thereafter, Section 5.2 introduces the used data set and additionally covers the data management throughout the application. Section 5.3 explains the general structure of a visualisation component whereas Section 5.4 goes into the details specific to one or more components.

5.1 General Layout

This section is about the general layout choices concerning the prototype's application. As described in the chapter's introduction, all prototypes are placed inside of their own view. All views follow the same layout choices, only the shown visualisations differ. At the top of the screen, you can find the University of Stuttgart logo and the nav bar to navigate between the different views, i.e., the different prototypes. Below that the user sees the actual prototype. A prototype consists of a main visualisation and an *aggregated view* in a column layout. The main visualisation takes up a bigger proportion of the available screen space and the *aggregated view* is placed above the main visualisation. At last, the drop-down menu for filtering the data for profession is placed at the bottom of the screen. An example of the scatterplot view can be found in Figure 5.1. Within the bounds of this bachelor's thesis, three prototypes were created. The composition of the three prototypes can be found in Table 4.1. The reasons for those design choices are laid out in Chapter 4.

[1] https://storage.googleapis.com/pantheon-public-data/person_2020_update.csv.bz2

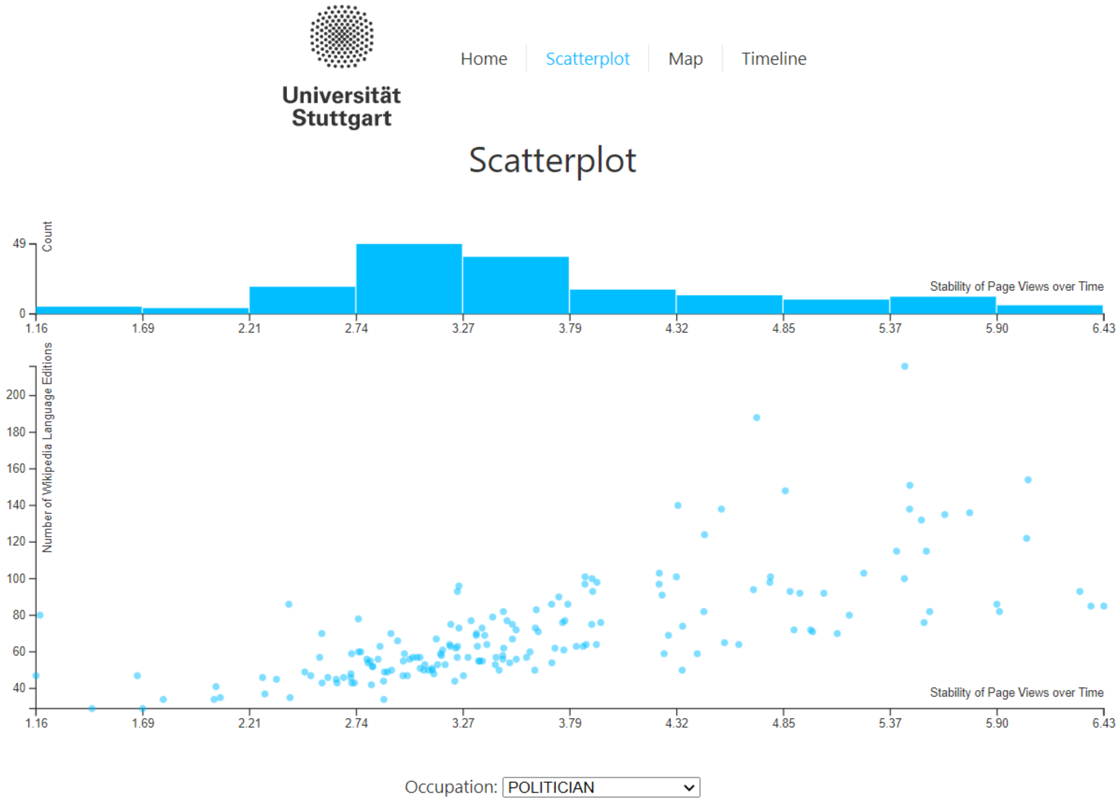


Figure 5.1: View of the scatterplot prototype

5.2 Data Set and Data Management

This section first presents the *Pantheon* [YRH+16] data set that is used in all visualisations in version 2.0 as well as the data for creating the map. Afterwards, the data management for the prototypes is explained.

Pantheon 2.0 Data Set Because of the chosen visualisations a data set is needed that includes quantitative, temporal and spatial data. This applies to the *Pantheon 2.0* [YRH+16] data set, which is why it is used for all three prototypes. The *Pantheon 2.0* [YRH+16] data set contains biographical data about people that have Wikipedia articles written about them in more than 15 different languages. Among other things, for all people, the birth and death date (temporal aspect), the birth - and death place (spatial aspect) and the number of different languages (quantitative aspect) are included. The exact composition of the data set is outlined on the website.
For the map, additional data was needed. The data to draw the countries' outlines was taken from a GitHub repository [Bos19]. The data to map a country to its region was taken from another GitHub repository [Dun20].

Data Management It is necessary to read data from files and to make the read data accessible to all components. Furthermore, some of the data needs to be shared between components in a way that makes the reactive rendering of the visualisations possible. That goal is achieved in the

Listing 5.1 Example HTML code for a visualisation component

```
<template>
    <div id="visualisation-type"></div>
</template>
```

Listing 5.2 Example code for the state object

```
const state = reactive({
    svg: undefined,
    padding: {top: 20, right: 20, bottom: 25, left: 50},
    // other variables shared by the whole component
});
```

context of this work with the help of Pinia data stores. A data store is very similar to a component but it serves a different purpose. To create a data store you declare the variables to be stored and implement functions that allow the manipulation of the stored data. The same data store can be accessed and observed by all components. This makes it possible for one component to push data to the store and another component to react immediately to that push, e.g. by rerendering a visualisation with new data. The filtering and *brushing and linking* are both handled this way. The data is initially read using native functions of d3.

5.3 Structure of a Visualisation Component

In this section, the general structure of a single visualisation component is covered. The focus lies on the elements that all or almost all visualisation components have in common. The more specific details follow in Section 5.4.
All visualisation components are so-called *Single-File Components* which means one file contains all JavaScript, HTML and CSS specific to that component. The HTML for the components is fairly short since it consists of one empty div which is manipulated in the JavaScript code to render the visualisation. Listing 5.1 shows an example of the HTML of a visualisation component.

Setup Function All JavaScript Code handling the rendering of the visualisation, reactivity and responsiveness is written inside the <*script setup*> function which is executed every time an instance of the specific component is created. At the beginning of the setup function, a reactive state object is created that contains all the variables that are shared between the functions of the component. An example of that can be seen in Listing 5.2. The function that is responsible for rendering the actual visualisation is also placed inside the setup function. It attaches a new group element to the component's svg and then draws the visualisation inside of that group, purely using d3. Whenever the visualisation needs to be rerendered, the group will simply be removed from the DOM and the rendering function will be called again. A general example of the rendering function is provided in Listing 5.4. Furthermore, Vue's *onMounted* function is used to execute some code whenever the component's DOM tree is generated and added to the parent's DOM tree, which in this case is

Listing 5.3 Example code for the onMounted function

```
onMounted(() => {
    dataStore.setTitle('Title');
    state.svg = d3.select('#visualisation-type').append('svg');
    resizeObserver.observe(document.getElementById('visualisation-type'));
});
```

Listing 5.4 Example code for the visualisation function

```
const drawVisualisation = (data) => {
    const g = state.svg.append("g").attr("id", "container-group");
    // d3 code for rendering visualisation
}
```

the view's DOM tree. An example for the *onMounted* implementation is accessible in Listing 5.3. First, the title of the webpage is set. After that, the svg element for the visualisation is created and appended to the component's div. At last, the resizeObserver is set up to observe the size of the div. Details can be found later in this section.

Responsiveness The prototypes are mostly meant to be used on mobile devices, but the usage on desktop or laptop pcs should also be possible. Because of that, the design of the web app needs to be responsive, as explained in Section 2.5. This application implements responsiveness by rerendering all visualisations visible in the active view. Each prototype's view wraps the used visualisation components in another div element whose height and width are depending on the viewport. The available space is then divided up between the components via CSS. This way the size of the components' div indirectly depends on the screen size. Now the components need to react whenever the size changes. The "reacting" is done through a native JavaScript interface called *ResizeObserver* [Fou98] which listens to changes on the border-box of a DOM element. Example code for this can be observed in Listing 5.6. Whenever the size of the surrounding div element changes, the component's svg's dimensions are adapted as well. Additionally, the svg group containing the actual visualisation is removed from the DOM and the rendering function is called. Basically, the visualisation is drawn with the new dimensions whenever the screen size changes.

Listing 5.5 Example code for the ResizeObserver

```
const resizeObserver = new ResizeObserver((entries) => {
    const entry = entries[0];
    const {width, height} = entry.contentRect;
    state.svg.attr("width", width).attr("height", height);
    d3.select("#visualisation-type #container-group").remove();
    drawVisualisation(filteredData);
});
```

Listing 5.6 Example code for the data listener

```
watch(
    (dataStore.filteredData),
    (newFilteredData) => {
        d3.select('#visualisation-type #container-group').remove();
        drawVisualisation(newFilteredData);
    }
);
```

Additionally, all the interactions are implemented in such a way, that they can be done using touch screens or mouse and keyboard. All the points mentioned above make the prototypes usable on mobile devices with touch screens and on computers that have a mouse and a keyboard.

Reactivity between components Some data needs to be shared between the components. For one, there is the data that is supposed to be visualised. Then there is the brushed data. The goal is that whenever the user chooses another profession, which leads to a change of data, all visible visualisations are rerendered automatically, without a page reload. The same goes for *brushing* the data. The brush in the main visualisation should immediately be *linked* in the *aggregated view*. This concept is called reactivity and can be achieved by using Vue's *watch* function. An example implementation is presented in Listing 5.6. In this instance, the *watch* function *watches* for changes to the data inside the data store. As soon as a change is detected, the visualisation is rerendered with the new data. The same idea is used to handle the reactivity of the brushes. Details for that are supplied in Section 5.4.

5.4 Visualisations

This section covers all implementation details that are more or less specific to one or more visualisation components. Those are mostly the axes and scales as well as the *brushing and linking* semantics. The scatterplot component starts off, followed by the choropleth map and the timeline. All of these components are used as the main visualisation in one of the prototypes. The section ends with the histogram and the bar chart. Both merely serve as *aggregated views* in at least one of the prototypes.

Scatterplot The axes for the scatterplot are simple. Both have a linear scale that ranges from the minimum element to the maximum element present in the dataset for the depicted attribute. The data points themselves are shown as svg circle elements. To reduce the effects of overplotting, the opacity value for the circles is set below 1.
The *brushing* semantics for the scatterplot are also easy to implement when using d3. The user can select a rectangle by clicking or touching and dragging. All data points inside that rectangle will be marked as brushed by changing their colour. At the same time, the brushed proportions are marked in the corresponding histogram in the same colour. These actions are handled by two different functions, one is called for every *brushing* event, and the other one is only called at the

Listing 5.7 Example code for the brush function of the scatterplot component

```
const brushed = (brushEvent) => {
    const brushDimensions = brushEvent.selection;
    if (brushDimensions === null) {
        dataStore.setBrushedData([]); // brush is cleared
    } else {
        const leftBoundary = brushDimensions[0][0];
        const rightBoundary = brushDimensions[1][0];
        const upperBoundary = brushDimensions[0][1];
        const lowerBoundary = brushDimensions[1][1];
        dataStore.setBrushedData(
            dataStore.filteredData.filter((datapoint)=> {
                const xCoordinate = xScale(datapoint[selectedXattribute]);
                const yCoordinate = yScale(datapoint[selectedYattribute]);
                return (
                    xCoordinate >= leftBoundary &&
                    xCoordinate <= rightBoundary &&
                    yCoordinate >= upperBoundary &&
                    yCoordinate <= lowerBoundary
                );
            });
        );
    }
};
```

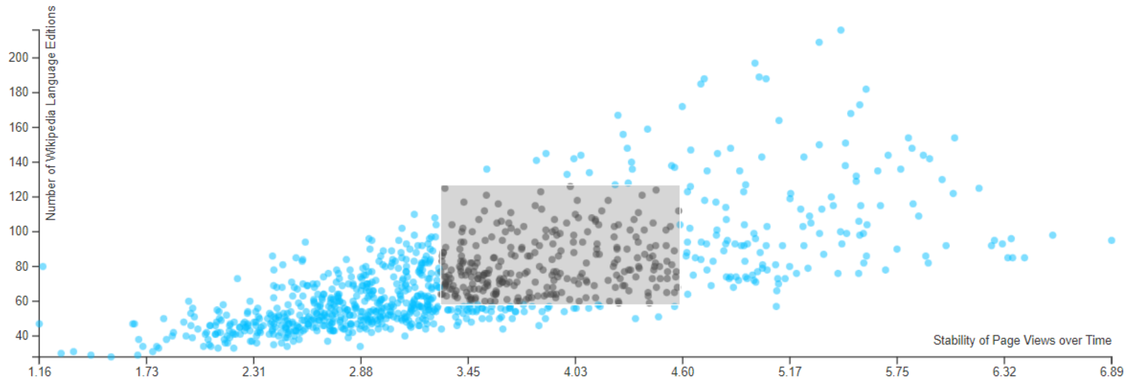


Figure 5.2: Example of a brushed scatterplot

end, out of performance reasons.

The first function is illustrated in Listing 5.7. It simply checks for all data points whether or not they are placed inside of the brush rectangle. All data points that are inside are pushed to another variable of the data store so they can be accessed by all components, e.g. the corresponding aggregated View. The second function can be found in Listing 5.8. It simply changes the colour of all brushed data points to anthracite. An example of a brushed scatterplot is displayed in Figure 5.2.

Listing 5.8 Example code for the brush end function of the scatterplot component

```
const brushEnd = () => {
    d3.select('#scatter-plot #container-group')
        .selectAll('circle')
        .data(dataStore.filteredData)
        .attr('fill', (datapoint) =>
            dataStore.brushedData.includes(datapoint)
                ? 'var(--colour-data-brushed)'
                : 'var(--colour-data-not-brushed)'
        );
};
```

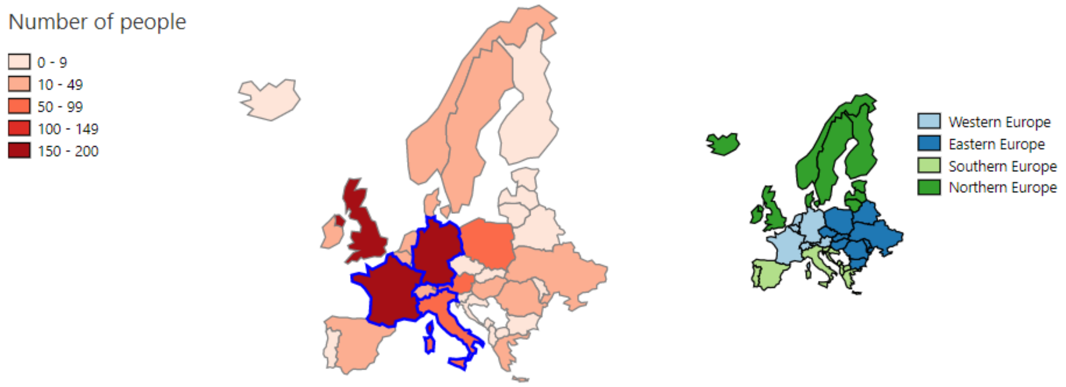


Figure 5.3: Example of a brushed choropleth map

Choropleth Map The choropleth map utilises the Mercator projection and only displays European countries. SVG paths are generated from a GeoJSON file, and a threshold colour scale is used to display the number of people born in each country. The colour scale has five shades of red, with bins closer to zero being smaller due to the high number of data points in that range. A colour legend is included to indicate the range of the colour scale, and another legend shows the four regions of Europe.

brushing is achieved through clicking or touching a country, which highlights the selected countries by changing their stroke colour to blue. Finally, the map filters data from the Pantheon dataset based on the selected birth countries and pushes the filtered data to the brushed data in the data store as can be seen in Listing 5.9. An example of a brushed choropleth map is shown in Figure 5.3.

Timeline The timeline needs two very different axes for the two dimensions. One axis is supposed to show a time variable, the other one is intended to portray the different people. The horizontal axis is chosen for the time, i.e., the life span. That is why this axis has a linear scale that ranges from the earliest birth date to the latest death date. The data points themselves are illustrated as rectangles that reach from the birth date to the death date. Each rectangle is placed in its own row. The *brushing* semantics are really similar to the scatter plot, only that the brush is limited to the time dimension. The user can choose a time span by clicking/touching and dragging. The selected data points are pushed to the data store and coloured in anthracite. The logic that decides

Listing 5.9 Example code for the filtering for selected countries

```
const filterDataForSelectedCountries = () => {
    dataStore.setBrushedData(
        filteredData.filter((person) => {
            selectedCountries.includes(person.birthCountry);
        });
    );
};
```

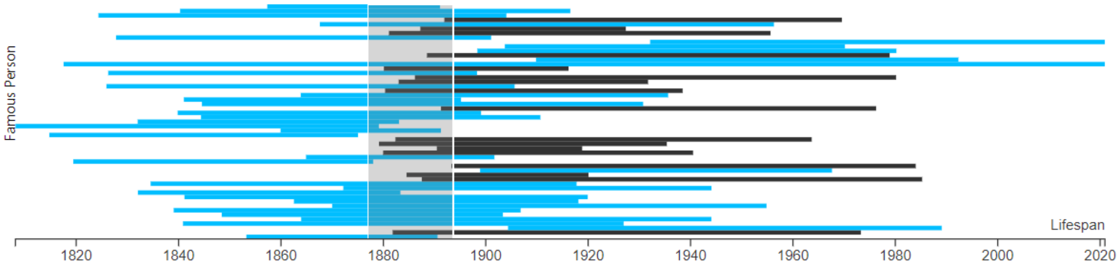


Figure 5.4: Example of a brushed timeline

which data points count as brushed can differ and should be chosen with the underlying task in mind. Sometimes it makes sense to count everyone as brushed that lived in the picked time span, sometimes it is more sensible to only count the people that were born or died in the time period. An example of a brushed timeline can be found in Figure 5.4.

Histogram The histogram is the chosen *aggregated view* for all variables of continuous kind. The x-axis is calibrated with a linear scale that ranges from the minimum value to the maximum value of the depicted attribute. The bin thresholds are calculated based on those min and max values and the desired number of bins. d3 does provide its own function for that purpose, but it is not very reliable when it comes to the number of bins. As a result, the binning semantics are implemented within the scope of this bachelor's thesis using a self-developed approach as illustrated in Listing 5.10. The bins themselves are then determined using d3 and the calculated thresholds. The y-axis is also calibrated using a linear scale, this time it ranges from 0 to the maximum count present in the bins. Since the histogram component is always used as an *aggregated view* rather than a main visualisation, no *brushing* semantics are needed. Instead, the linking needs to be implemented, i.e. the representation of the brushed data points from the main visualisation. This is done by highlighting the corresponding proportions in the same colour as the brushed data points in the main visualisation,

Listing 5.10 Example implementation for calculating bin thresholds

```
const thresholdArray = [...Array(numberOfBins)].map(
    (item, i) => (maxValue - minValue) / numberOfBins * i + minValue;
);
```

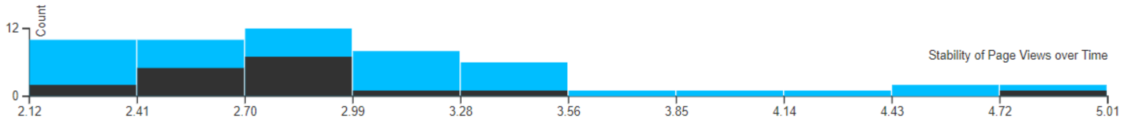


Figure 5.5: Example of a brushed histogram

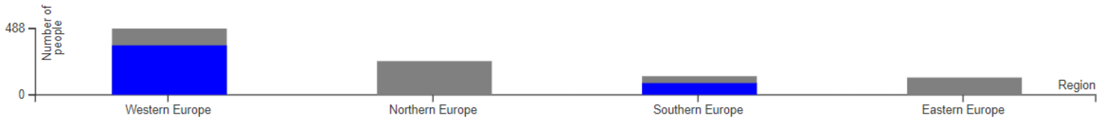


Figure 5.6: Example of a brushed bar chart

as described in Chapter 4. In practice, a new histogram in anthracite is drawn on top of the other one, but the bin heights are based on the brushed data and not all data. To make this reactive, another listener is necessary that observes the brushed data in the data store. Whenever those change, the brushed histogram is rerendered. An example of a brushed histogram can be found in Figure 5.5.

Bar Chart The bar chart is used as the *aggregated view*, whenever the histogram is not applicable, i.e. the data attribute is nominal. Here the x-axis has a nominal scale. No bins or thresholds need to be calculated, but the counts per attribute value still need to be determined. The y-axis is calibrated with a linear scale ranging from 0 to the maximum count, exactly as in the histogram. The linking semantics are also implemented in the same way.

6 Evaluation

The next logical step after implementing the prototypes as described in Chapter 5 is the evaluation of said prototypes in a user study. Since these prototypes are implemented for the usage on mobile devices, screen space is rare. The interaction with the main visualisation works better, the larger the visualisation is. That is why one wants to use as much screen space as possible for the main visualisation, which does not leave a lot of space for the *aggregated view*. Consequently, it needs to be investigated how the size of the *aggregated view* affects the accuracy and completion time when working with the brushed and *linked* visualisations. Additionally, this evaluation investigates whether or not the implemented *linking* is understood by the users.

6.1 Study Design

This section describes the design of the conducted studies. In the beginning, the research questions are introduced. Afterwards, the general methodology used for the studies is described. Subsequently, the stimuli are briefly described, followed by the data analysis. The section is completed with a short overview of the participants.

6.1.1 Research Questions

The above considerations lead to the research questions and hypotheses presented in Table 6.1 that are the basis of the conducted studies. For each aspect we want to investigate, one research question is presented in the table. “Research Question X” is abbreviated with *RQX*. *RQ1* covers the understandability of the linking approach, *RQ2* is about the accuracy and at last *RQ3* examines the completion time. Per research question, one hypothesis, abbreviated with H_X, and one corresponding null hypothesis, abbreviated with H_{X_0} is provided.

6.1.2 Methodology

To scrutinise the aforementioned research questions, a total of three user studies in the form of online questionnaires via LimeSurvey were conducted: One study for each prototype. For all studies, the same methodology was used. The study design was within-subject. Mean completion time and accuracy were measured for all participants per condition. For each task and stimulus, the participants had a choice between four answer options of which only one was correct. During a trial, the participants saw one stimulus after the other. Five different heights of the *aggregated view* were tested, three stimuli per group. The groups were presented in randomised order, as well as the stimuli within these groups to prevent learning effects. Beforehand, two example stimuli and an explanation of the visualisations were given. There was no time limit but the users were asked to

RQ1	Is the implemented *linking* understandable to the user?
H_1	The number of correct answers is higher than by chance.
H_{1_0}	The proportion of correct answers is not higher than 25%
RQ2	Does the height of the *aggregated view* affect the accuracy with which users read the visualisation?
H_2	The accuracy with which users understand the visualisation depends on the height of the *aggregated view*.
H_{2_0}	There is no significant difference between the accuracy distributions of the groups.
RQ3	Does the height of the *aggregated view* affect the time it takes for users to understand the visualisation?
H_3	The completion time for users reading the visualisation depends on the height of the *aggregated view*.
H_{3_0}	There is no significant difference between the mean completion times of the groups.

Table 6.1: Research questions and hypotheses

answer as accurately as possible. To assure that the sizes were the same for all participants, they were asked in advance to zoom in or out with the browser until a reference rectangle had about the size of a credit card.

6.1.3 Stimuli

For each study, five groups of stimuli were used. The groups differed in the height of the *aggregated view*. For the timeline and scatterplot, the heights of 40px, 60px, 80px, 120px and 160 px were used. For the choropleth map, the heights of 20px, 40px, 60px, 80px and 100px were used. For each height, a total of three stimuli were presented.

6.1.4 Data Analysis

To answer the formulated research questions, we applied a spectrum of statistical tests to the data collected in the study. The choice of which test to apply for RQ2 and RQ3 is based on the book by Motulsky [Mot14]. For both questions, five groups of a within-subject study need to be compared. If the data is normally distributed, the repeated-measures ANOVA [Gir92] test is appropriate. The sphericity as a precondition for the ANOVA test is shown via Mauchly's [Mau40] test. If the data is not normally distributed, the choice is a Friedman test. To test for normality, the Shapiro-Wilk [SW65] test is used. If those tests find any significant effect, a post-hoc test needs to be run. This is done to find the groups, that have a significant difference between them. For the ANOVA test, pairwise t-tests [CGHH91] with Holm [Hol79] corrections are the recommended post-hoc tests. The Holm correction is done in order to control the family-wise error rate [Hol79], i.e., it is used to reduce the probability of false positives during multiple pair-wise tests. The suitable post-hoc test for the Friedman test is the Wilcoxon [RN11] signed-rank test, also with Holm corrections.

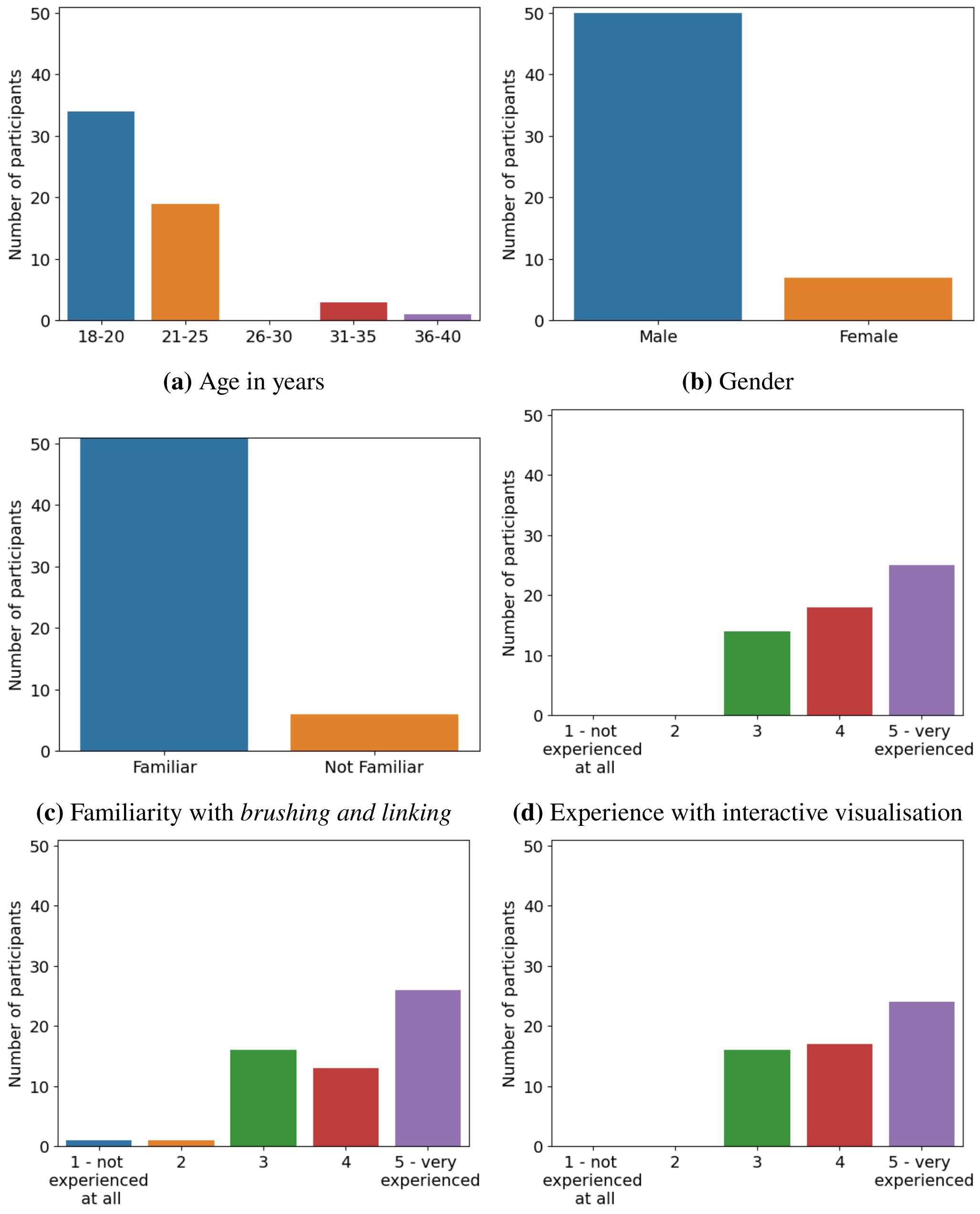
For RQ1, a simple one-sided binomial test [Wag05] with a hypothesised *success rate* of 0.25 can be run. The correct answer option is counted as *success* and all other options as *not success*. This way we artificially create a dichotomous variable. All other preconditions are also met since the trials are independent and the number of trials is fixed and known.

For all tests, an α-value of 0.05 is used.

6.1.5 Participants

For the studies, members of the VIS Institute as well as students from the faculty of Computer Science were recruited via e-mail. In total, 70 people participated. Of those 70 trials, 57 were complete. Participants were not compensated for completing the study. An overview of the participant's demographics and prior experience with the concepts used in the study is presented in the following paragraph and can also be found in Figure 6.1.

For the scatterplot, there were 20 completed trials, for the choropleth map there were 18 completed trials and for the timeline, 19 participants submitted a complete survey. 87% of the participants were male and 13% were female, as can be seen in Figure 6.1c. 60% were between 18 and 20 years old, and 33% of all participants were aged between 21 and 25. The rest was between 26 and 40 years old like Figure 6.1a depicts. 90% of all participants reported that they were familiar with the term *brushing and linking* in the context of information visualisation, cf. Figure 6.1b. 68% of the participants stated that they were pretty experienced with the used main visualisation, while 72% declared that they were quite experienced with the used aggregated view. At last, 75% claimed to have more than a bit of experience with interactive visualisations. Those numbers were gathered with the use of Likert scales ranging from one (not experienced at all) to five (very experienced). The exact answer distribution is displayed in Figures 6.1e, 6.1f and 6.1d.



(a) Age in years

(b) Gender

(c) Familiarity with *brushing and linking*

(d) Experience with interactive visualisation

(e) Experience with the main visualisation

(f) Experience with aggregated view

Figure 6.1: Overview of demographics and prior experience of participants

6.2 Results

As described before, three studies were conducted, one for each prototype. The results of the data analysis for the three research questions are presented in this section for each prototype individually, starting off with the scatterplot/histogram, followed by the timeline/histogram and completing with the choropleth map/bar chart.

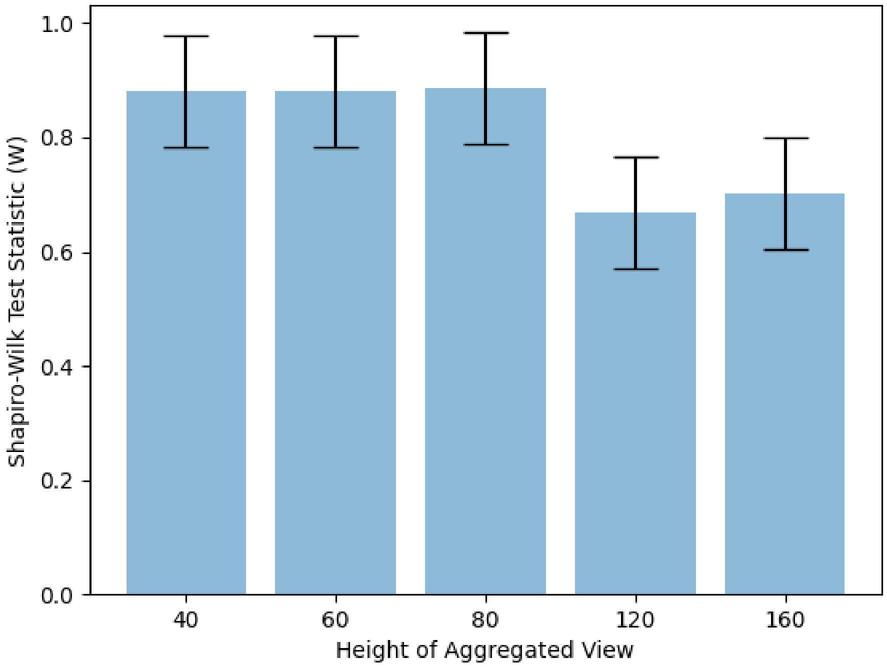


Figure 6.2: Results of Shapiro-Wilk test for the accuracy of the scatterplot/histogram prototype

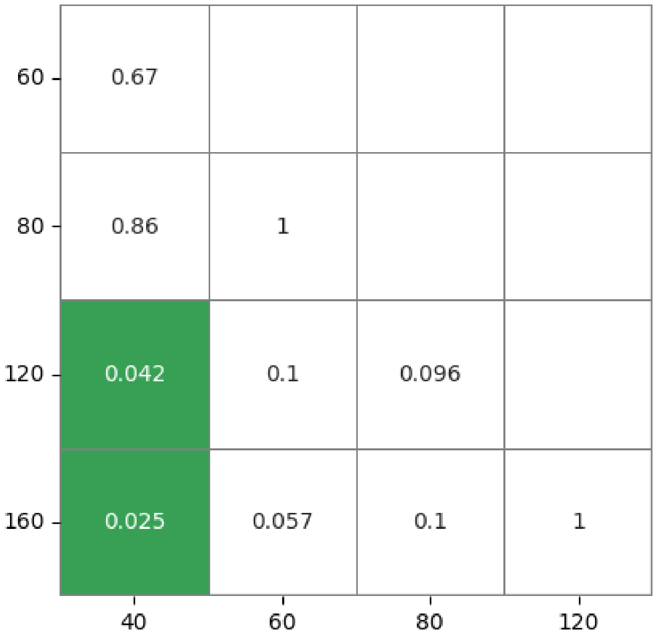
6.2.1 Scatterplot / Histogram

RQ1 The binomial test reveals that the number of correct answers is significantly higher than by chance. Out of 300 answers, 197 are correct, which results in a p-value of 0.0. $H1_0$ is rejected.

RQ2 The results of the Shapiro-Wilk tests for the five groups are shown in Figure 6.2. The error bars show the 95% confidence intervals. None of them overlap with 1.0, especially the groups with histogram heights of 120px and 160px are far away. Consequently, the normality cannot be assumed and the Friedman test is applied to compare the group distributions.

The Friedman test reveals a significant difference between the accuracy distributions of the groups ($Q = 20.52$, $p = 0.000395$). $H2_0$ is rejected.

The post-hoc test using Wilcoxon signed-rank tests with Holm corrections shows significant differences between the groups 40px / 120px (*corrected* $p = 0.042$) and between the groups 40px / 160px (*corrected* $p = 0.025$), as shown in Figure 6.3. The boxplots depicted in Figure 6.4 underline those findings. The accuracy values for the 60px and 80px groups are similar and a little higher than the values of the 40px group. The accuracy values of the 120px and 160px groups are very similar and considerably higher than the values of the 40px group.



	40	60	80	120
60	0.67			
80	0.86	1		
120	0.042	0.1	0.096	
160	0.025	0.057	0.1	1

Figure 6.3: Results of the post-hoc Wilcoxon signed-rank tests for the accuracy of the scatterplot/histogram prototype, tests with significant differences highlighted in green

RQ3 The results of the Shapiro-Wilk tests for the five groups are shown in Figure 6.5. The error bars show the 95% confidence intervals. They all overlap with the value 1.0. Consequently, the normality can be assumed and the repeated-measures ANOVA test is applied to compare the group means. Mauchly's test (W = 0.71, p = 0.76) shows the sphericity of the data which is the last requirement for the ANOVA test.

The ANOVA test (*F(4, 76)* = 1.10, p = 0.36) reveals no significant difference between the group means. The boxplots depicted in Figure 6.6 underline those results as they also show no considerable differences between the groups' distributions. $H3_0$ cannot be rejected.

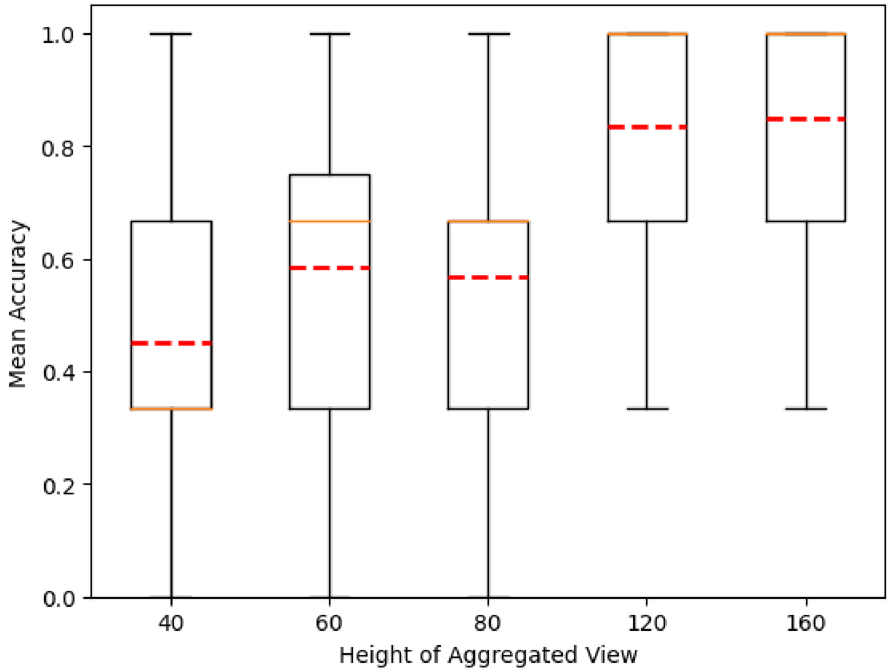


Figure 6.4: Boxplots of the accuracy distributions of the scatterplot/histogram prototype, median as orange line and mean as red dashed line

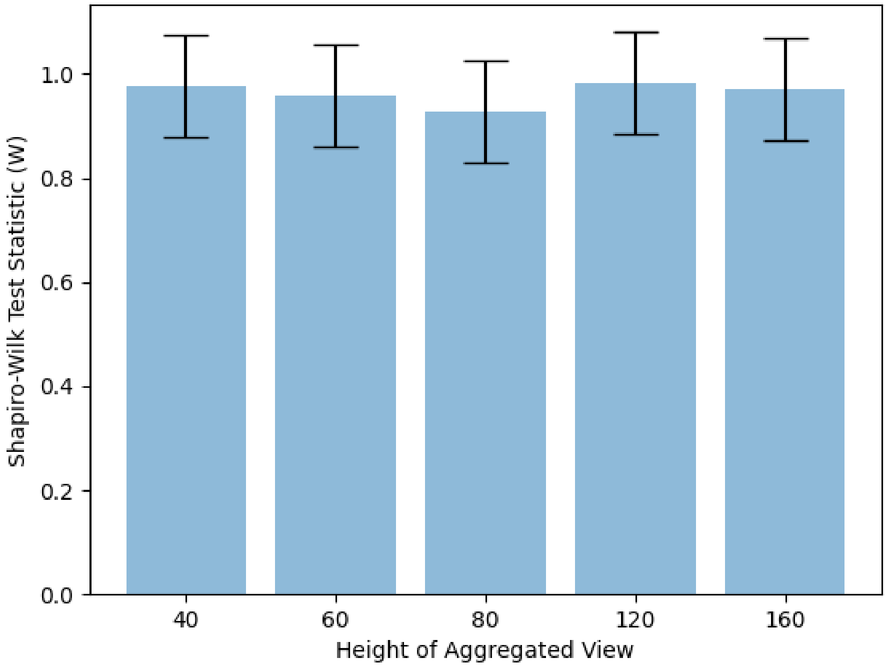


Figure 6.5: Results of Shapiro-Wilk test for the completion time of the scatterplot/histogram prototype

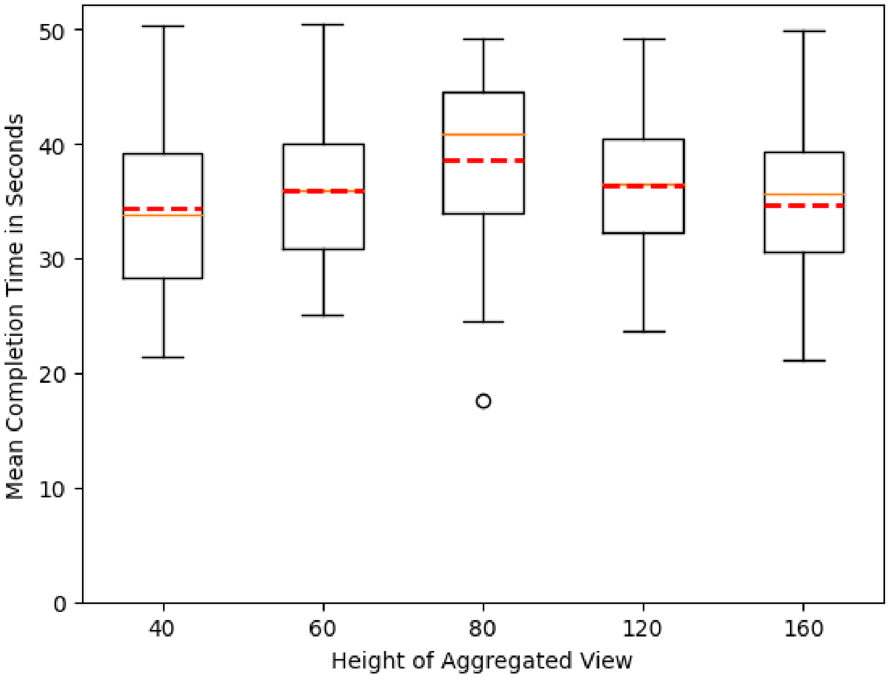


Figure 6.6: Boxplots of the completion time distributions of the scatterplot/histogram prototype, median as orange line and mean as red dashed line

6.2.2 Timeline / Histogram

RQ1 The binomial test reveals that the number of correct answers is significantly higher than by chance. Out of 285 answers, 149 are correct, which results in a p-value of 0.0. $H1_0$ is rejected.

RQ2 The results of the Shapiro-Wilk tests for the five groups are shown in Figure 6.7. The error bars show the 95% confidence intervals. None of them overlap with 1.0, especially the group with a histogram height of 160px is far away. Consequently, the normality cannot be assumed and the Friedman test is applied to compare the group distributions.

The Friedman test reveals a significant difference between the accuracy distributions of the groups ($Q = 28.05$, $p = 0.000012$). $H2_0$ is rejected.

The post-hoc test using Wilcoxon signed-rank tests with Holm corrections shows significant differences between the groups 40px / 120px (*corrected* $p = 0.044$), between the groups 40px / 160px (*corrected* $p = 0.006$), between the groups 60px / 120px (*corrected* $p = 0.029$), between the groups 60px / 160px (*corrected* $p = 0.023$) and between the groups 80px / 160px (*corrected* $p = 0.029$), as shown in Figure 6.8. The boxplots depicted in Figure 6.9 underline those findings. The accuracy values for the 120px and 160px groups are similar and considerably higher than the values of the 40px, 60px and 80px groups.

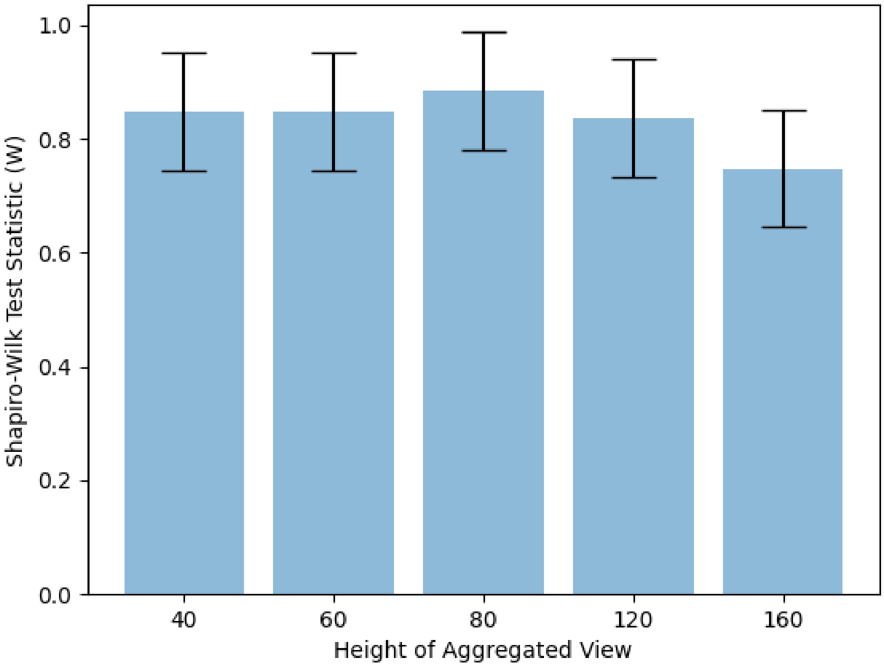
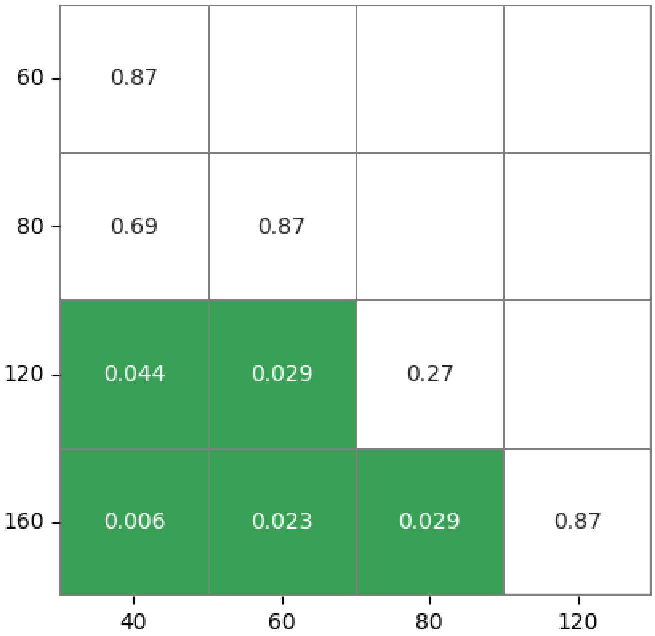


Figure 6.7: Results of Shapiro-Wilk test for the accuracy of the timeline/histogram prototype



	40	60	80	120
60	0.87			
80	0.69	0.87		
120	0.044	0.029	0.27	
160	0.006	0.023	0.029	0.87

Figure 6.8: Results of the post-hoc Wilcoxon signed-rank tests for the accuracy of the timeline/histogram prototype, tests with significant differences highlighted in green

RQ3 The results of the Shapiro-Wilk tests for the five groups are shown in Figure 6.10. The error bars show the 95% confidence intervals. They all overlap with the value 1.0. Consequently, the normality can be assumed and the repeated-measures ANOVA test is applied to compare the group means. Mauchly's test ($W = 0.48$, $p = 0.21$) shows the sphericity of the data which is the last requirement for the ANOVA test.

The ANOVA test ($F(4, 72) = 1.5$, $p = 0.21$) reveals no significant difference between the group means. The boxplots depicted in Figure 6.11 underline those results as they also show no considerable differences between the groups' distributions. $H3_0$ cannot be rejected.

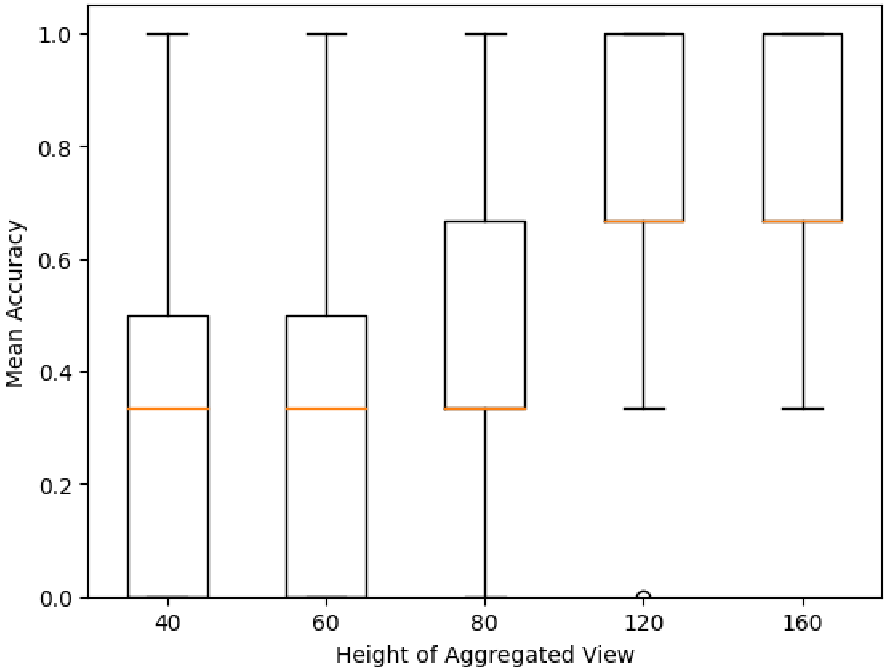


Figure 6.9: Boxplots of the accuracy distributions of the timeline/histogram prototype, median as orange line and mean as red dashed line

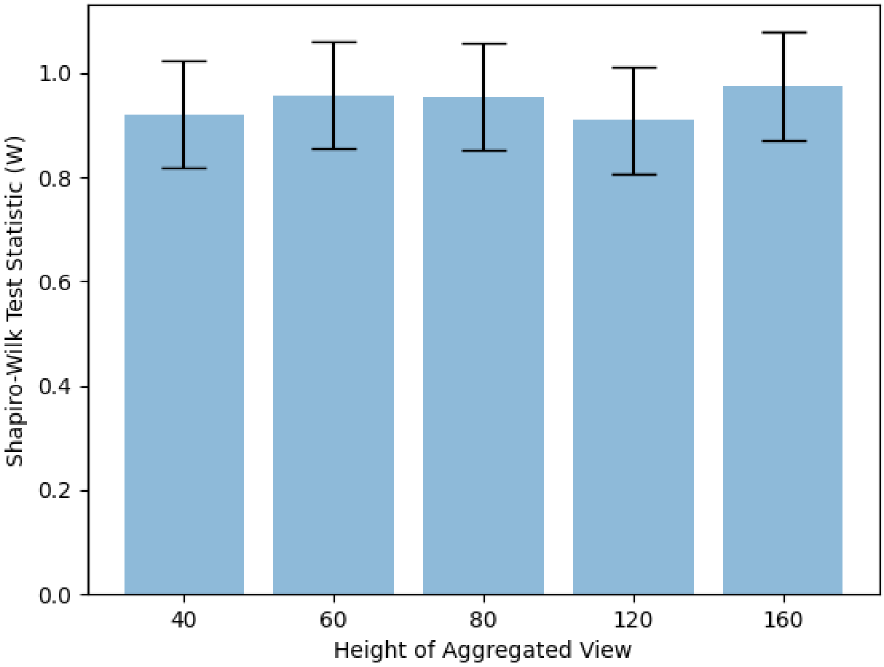


Figure 6.10: Results of Shapiro-Wilk test for the completion time of the timeline/histogram prototype

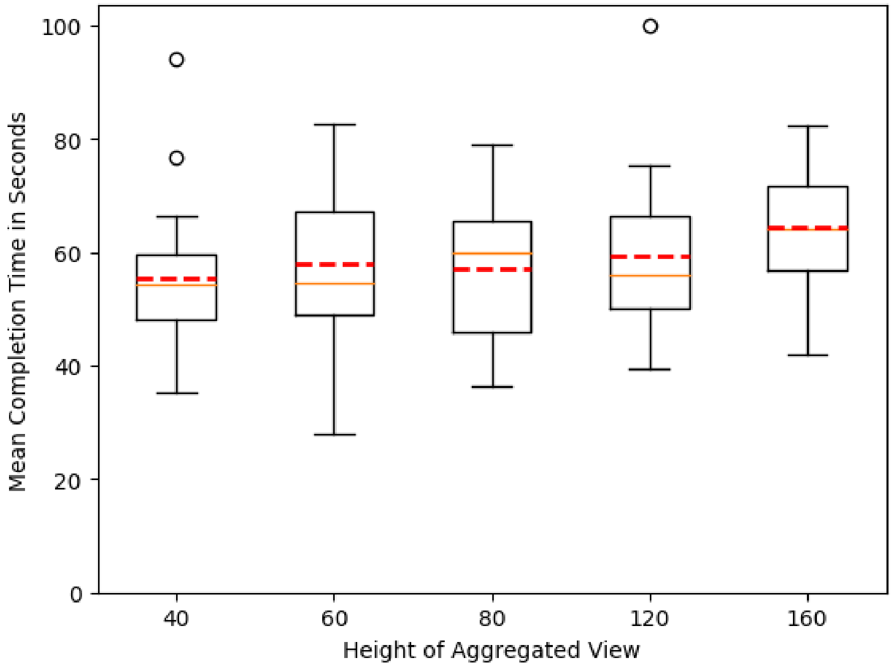


Figure 6.11: Boxplots of the completion time distributions of the timeline/histogram prototype, median as orange line and mean as red dashed line

6.2.3 Choropleth Map / Bar Chart

RQ1 The binomial test reveals that the number of correct answers is significantly higher than by chance. Out of 270 answers, 167 are correct, which results in a p-value of 0.0. $H1_0$ is rejected.

RQ2 The results of the Shapiro-Wilk tests for the five groups are shown in Figure 6.12. The error bars show the 95% confidence intervals. None of them overlap with 1.0, especially the groups with histogram heights of 80px and 100px are far away. Consequently, the normality cannot be assumed and the Friedman test is applied to compare the group distributions.

The Friedman test reveals a significant difference between the accuracy distributions of the groups ($Q = 21.66$, $p = 0.000234$). $H2_0$ is rejected.

The post-hoc test using Wilcoxon signed-rank tests with Holm corrections shows significant differences between the groups 20px / 80px (*corrected* $p = 0.018$), between the 20px / 100px groups (*corrected* $p = 0.034$) and between the groups 40px / 80px (*corrected* $p = 0.012$), as shown in Figure 6.13. The boxplots depicted in Figure 6.14 underline those findings. The accuracy values for the 60px, 80px and 100px groups are by far higher than the values of the 20px and 40px groups, with the 80px groups generally having the highest values. The accuracy values of the 100px group are only a little higher than the values of the 60px group.

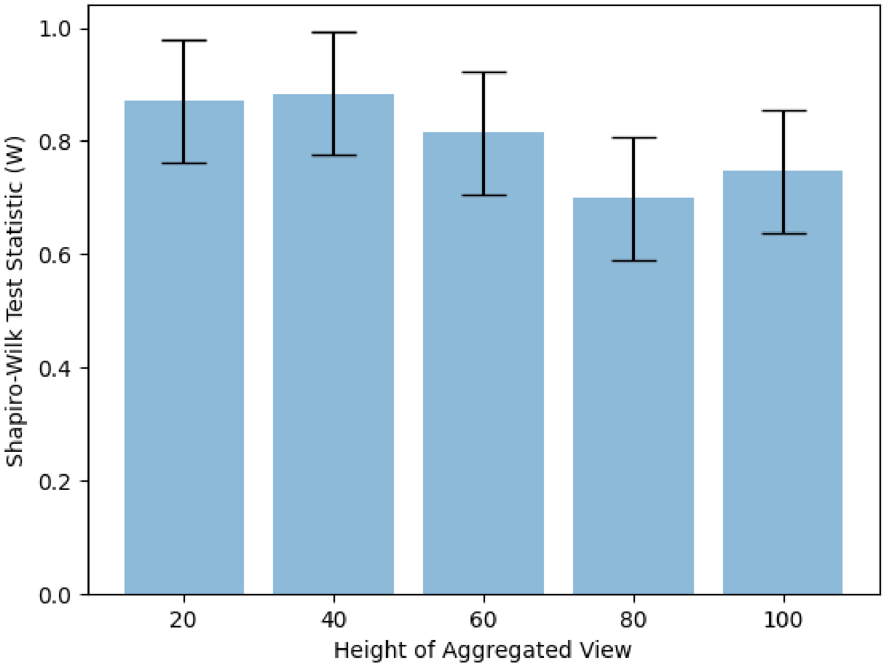
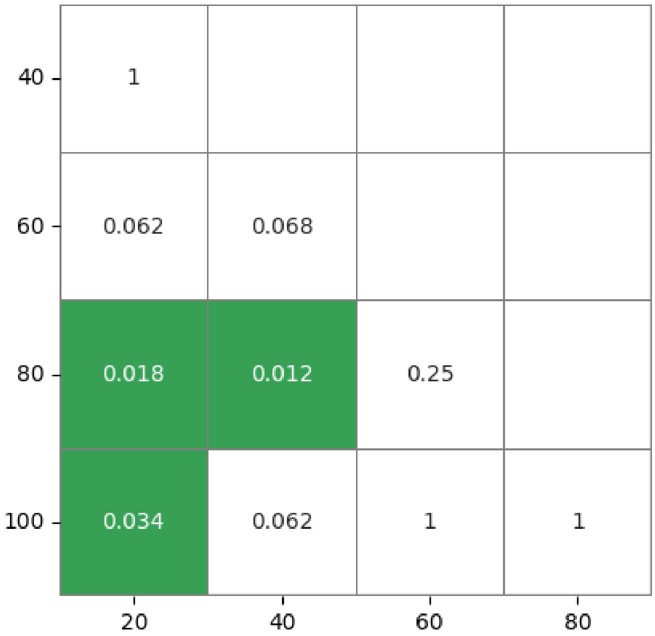


Figure 6.12: Results of Shapiro-Wilk test for the accuracy of the choropleth map/bar chart prototype



	20	40	60	80
40	1			
60	0.062	0.068		
80	0.018	0.012	0.25	
100	0.034	0.062	1	1

Figure 6.13: Results of the post-hoc Wilcoxon signed-rank tests for the accuracy of the choropleth map/bar chart prototype, tests with significant differences highlighted in green

RQ3 The results of the Shapiro-Wilk tests for the five groups are shown in Figure 6.15. The error bars show the 95% confidence intervals. They all overlap with the value 1.0. Consequently, the normality can be assumed and the repeated-measures ANOVA test is applied to compare the group means. Mauchly's test ($W = 0.48$, $p = 0.25$) shows the sphericity of the data which is the last requirement for the ANOVA test.

The ANOVA test ($F(4, 68) = 0.27$, $p = 0.9$) reveals no significant difference between the group means. The boxplots depicted in Figure 6.16 underline those results as they also show no considerable differences between the groups' distributions. $H3_0$ cannot be rejected.

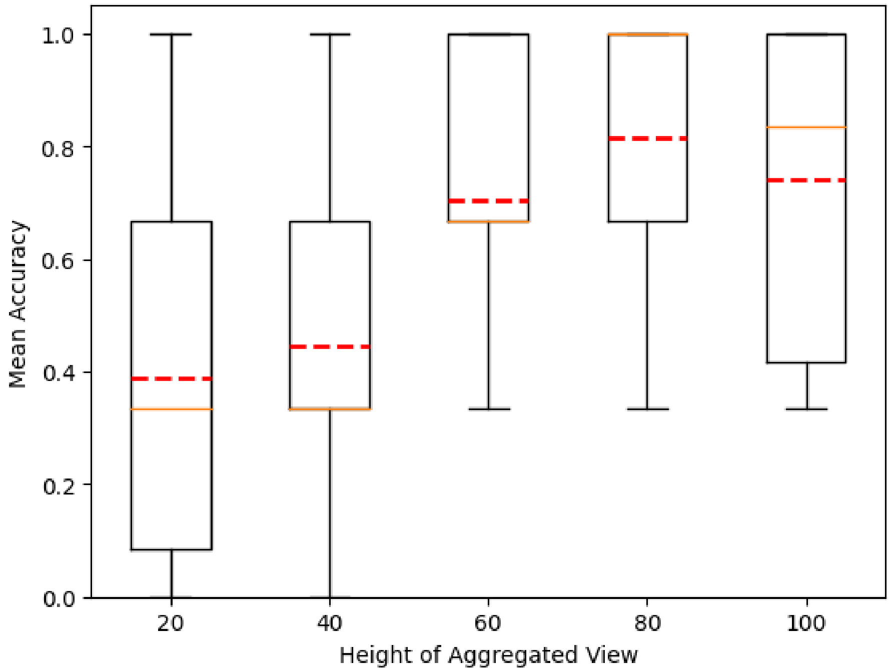


Figure 6.14: Boxplots of the accuracy distributions of the choropleth map/bar chart prototype, median as orange line and mean as red dashed line

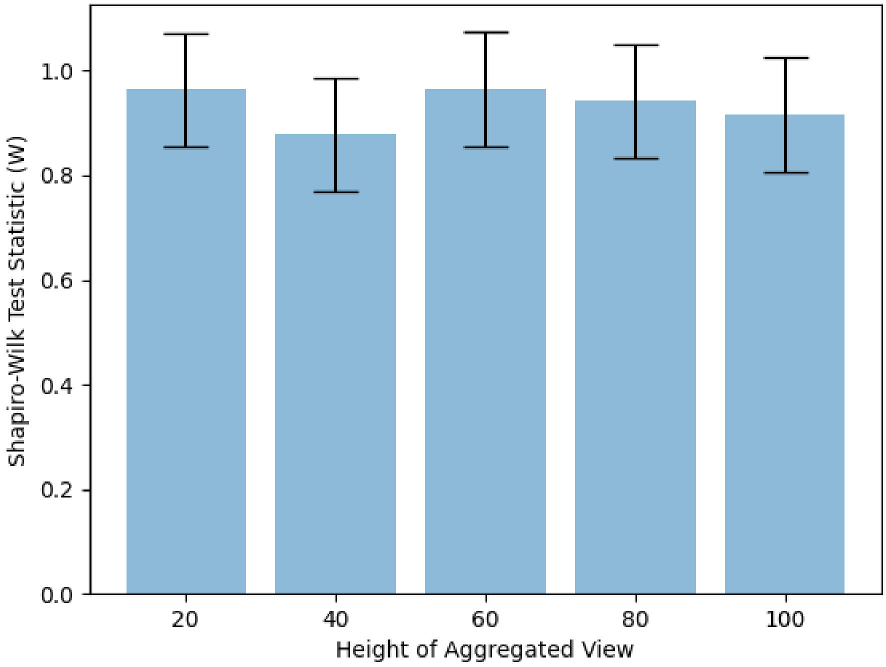


Figure 6.15: Results of Shapiro-Wilk test for the completion time of the choropleth map/bar chart prototype

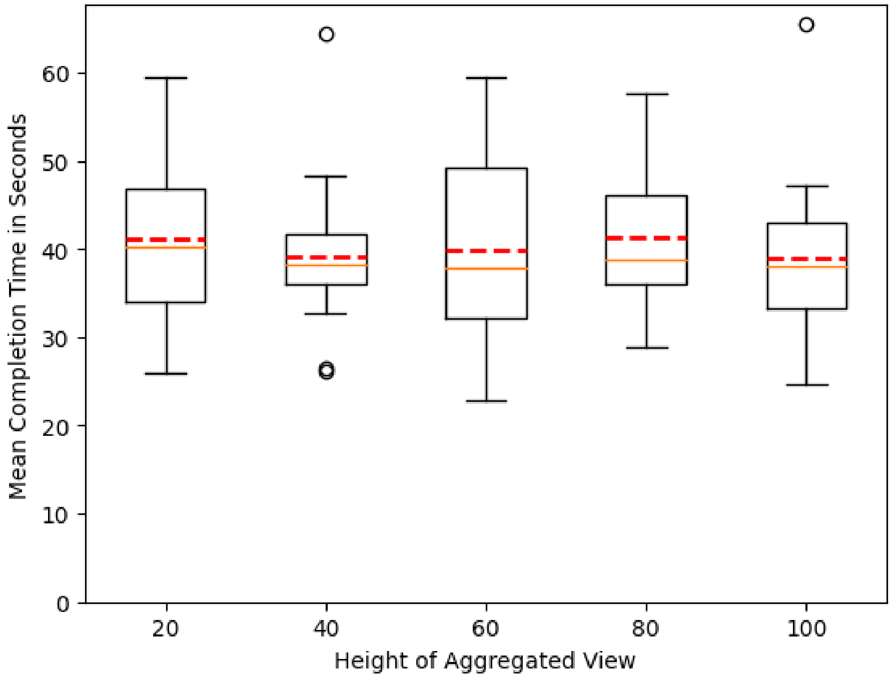


Figure 6.16: Boxplots of the completion time distributions of the choropleth map/bar chart prototype, median as orange line and mean as red dashed line

6.3 Discussion

This section discusses the findings of the conducted user studies described in the previous section.

Based on the studies it can be said for all prototypes that the implemented *linking* design is generally understandable. The amount of correct answers provided by the participants is much higher than if the participants were only guessing.

The studies also show that the time it takes to understand the *linking* does not depend on the height of the used *aggregated view*. The boxplots in Figures 6.16, 6.11, 6.6 show no significant differences between any of the groups' distributions.

Finally, it is revealed that the height of the used *aggregated view* has an influence on the accuracy with which a user reads the visualisation. Generally, the sizes on the larger end of the spectrum performed better. For the scatterplot/histogram prototype, the sizes 120px and 160px yielded the best results, as can be seen in Figures 6.3 and 6.4. For the timeline/histogram prototype, the sizes 120px and 160px also yielded the best results, as can be seen in Figures 6.8 and 6.11. At last, for the choropleth map/bar chart prototype, 60px, 80px and 100px yielded the best results, with 80px slightly outperforming 60px and 100px. Based on those results it is quite evident, that the height of the *aggregated view* does matter. It should not be too small, the larger heights generally performed better than the very small heights. But the difference in accuracy between the larger heights itself is not significant, so the benefit of increasing the height decreases at a certain point. The goal is to make the *aggregated view* as small as possible so the interaction with the *main visualisation* becomes easier. A trade-off needs to be made.

Another observation that can be made from the studies is, that the completion time for the timeline prototype was generally higher than for the other prototypes. Also, the accuracy for the timeline was a little lower. The exact values can be found in Table 6.2. However, they need to be interpreted with caution, since they are gathered from the different studies with different participants. These values can be explained with a few issues concerning the design of the timeline/histogram prototype. The way the brushed data points are highlighted within the timeline itself is not completely intuitive. Although only the people born within the brushed time period are counted as brushed, the lifespan of the whole person is highlighted by colour. This way, the highlighting possibly continues outside of the brushed timespan. Additionally, the x-axes of the timeline and the histogram are aligned spatially but not semantically. They do not have the same scales (both start with the earliest birth year, but the timeline stops at the latest death year while the histogram stops at the latest birth year). Improving those issues might enhance the performance for the prototype.

Threats to Validity Since the evaluation was conducted as an online questionnaire, there are multiple threats to validity.

prototype	mean accuracy	mean completion time
scatterplot	65.7%	36 seconds
timeline	52.3%	58.8 seconds
choropleth map	62%	40.1 seconds

Table 6.2: Mean accuracy and mean completion time per prototype

There are a few sources of internal threads to validity. For each study, the questions were created based on the same data set. Also, the alignment of the axes is not always given, especially if the *aggregated view* shows a variable not present in the *main visualisation*. This is actually the case most of the time since the *aggregated view* should give additional insides into the data. The evaluation did not test the actual interaction with the systems. The questions all consisted of static visualisations, the user did not have to interact with the main visualisation. Those interactions are a major part of the visualisations and should be tested as well, which could be done in a future lab study. The same applies to using mobile devices. Also, there is no way to guarantee that the participants did not change the zoom level during the trial.

There are also a few sources of external threads to validity which limit the generalisability of the study. The used visualisations are not representative of all visualisation types. Although lots of different data types and very common visualisations are covered, not everything is covered. Especially hierarchical and network data are not represented at all. The amount of used *aggregated views* could be extended as well. As described in Section 6.1, the participants all have an academic background and were quite familiar with the used concepts beforehand. This group is not representable for the general population but that is not a problem because the intended target population are visualisation researchers.

7 Conclusion

This section summarises the bachelor's thesis and describes the results of the evaluation. Furthermore, potential future work is proposed.

7.1 Summary

We presented an approach for the implementation of *brushing and linking* techniques in *aggregated views* on mobile devices. First of all, a conceptual design was drawn up, that proposes to highlight the *brushed* data points in both visualisations by colour. Three prototypes were implemented based on that conceptual design. The prototypes were programmed in a responsive manner, so that they can be used on desktop and mobile devices, especially the different screen sizes and interaction possibilities are taken into account here. One prototype shows quantitative data (scatterplot and histogram), one shows temporal data (timeline and histogram) and the last one shows geospatial data (choropleth map and bar chart). With the goal of evaluating those prototypes, three online user studies were conducted.

7.2 Evaluation Results

The results of the evaluation imply that the implemented *brushing and linking* technique is generally understandable. Additionally, some flaws of the timeline implementation are highlighted, which make the reading of the visualisation less intuitive and a bit confusing. Those flaws resulted in a slightly worse performance for that prototype.

On the one hand, the studies' results also indicate that the time it takes to read such a visualisation does not depend on the height of the used *aggregated view*. On the other hand, the height of the *aggregated view* does affect the accuracy with which a user can read such a visualisation. The studies show that increasing the height of the *aggregated view* enhances the performance, but the size of that effect decreases at a certain point. It is not sensible to make the *aggregated view* as large as possible since the screen space is generally scarce. The *main visualisation* should take up a considerably larger proportion since this makes the interaction with it easier. The *aggregated view* should be large enough to let a user read values from it with sufficient accuracy, but it should also be as small as possible.

7.3 Future Work

Finally, this section proposes some future work based on this bachelor's thesis. One possibility is to include multiple *aggregated views* in the prototypes that each show different variables. This can lead to additional insights but it needs to be tested as the user further splits their attention between additional views. Also, additional prototypes could be implemented that include visualisations for more data types. For example, a treemap for hierarchical data or a graph for network data would be sensible extensions of this thesis' work. Another possibility to alter the existing prototypes is to allow *brushing* interactions with the *aggregated view*. Those interactions are limited, however, because one needs to keep the interactions unambiguous. In particular, for the histogram, it should only be possible to brush complete bars.

Another point of entry for some future work is the evaluation. In our study, we showed static visualisations instead of letting the participants interact with them. This way the *brushing* interactions are never tested. This can best be done in a lab study.

Bibliography

[AS17] K. Andrews, A. Smrdel. “Responsive Data Visualisation.” In: *EuroVis (Posters)*. 2017, pp. 113–115 (cit. on p. 25).

[BLST21] M. Brehmer, B. Lee, J. Stasko, C. Tominski. “Interacting with visualization on mobile devices”. In: *Mobile Data Visualization* (2021), pp. 67–110 (cit. on p. 25).

[BMMS91] A. Buja, J. Mcdonald, J. Michalak, W. Stuetzle. “Interactive Data Visualization using Focusing and Linking”. In: *Visualization 91*. 2nd annual conf of the IEEE on Visualization (Visualization 91), San Diego, CA, Oct 22-25, 1991. IEEE, Tech Comm Comp Graphics; Assoc Comp Machinery. 1991, pp. 156–&. ISBN: 0-8186-2245-8 (cit. on pp. 15, 22).

[BNRS13] U. Brandes, B. Nick, B. Rockstroh, A. Steffen. “Gestaltlines”. In: *Computer Graphics Forum* 32.3pt2 (2013), pp. 171–180. DOI: https://doi.org/10.1111/cgf.12104. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12104. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12104 (cit. on p. 25).

[BNW+16] K. Blumenstein, C. Niederer, M. Wagner, G. Schmiedl, A. Rind, W. Aigner. “Evaluating Information Visualization on Mobile Devices: Gaps and Challenges in the Empirical Evaluation Design Space”. In: BELIV ’16. Baltimore, MD, USA: Association for Computing Machinery, 2016, pp. 125–132. ISBN: 9781450348188. DOI: 10.1145/2993901.2993906. URL: https://doi.org/10.1145/2993901.2993906 (cit. on p. 15).

[BOH11] M. Bostock, V. Ogievetsky, J. Heer. “D^3 Data-Driven Documents”. In: *IEEE Transactions on Visualization and Computer Graphics* 17.12 (2011), pp. 2301–2309. DOI: 10.1109/TVCG.2011.185 (cit. on p. 33).

[Bos19] M. Bostock. *World Atlas TopoJSON v2.0.2*. 2019. URL: https://github.com/topojson/world-atlas (cit. on p. 34).

[BW17] F. Beck, D. Weiskopf. “Word-Sized Graphics for Scientific Texts”. In: *IEEE Transactions on Visualization and Computer Graphics* 23.6 (June 2017). IEEE Pacific Visualization Symposium (IEEE PacificVis), Seoul Natl Univ, Seoul, South Korea, APR 18-21, 2017, pp. 1576–1587. ISSN: 1077-2626. DOI: 10.1109/TVCG.2017.2674958 (cit. on p. 25).

[CGHH91] K. Church, W. Gale, P. Hanks, D. Hindle. “Using statistics in lexical analysis”. In: *Lexical acquisition: exploiting on-line resources to build a lexicon*. Psychology Press, 1991, pp. 115–164 (cit. on p. 44).

[Dun20] L. Duncalfe. *ISO-3166 Country and Dependent Territories Lists with UN Regional Codes v9.0*. 2020. URL: https://github.com/lukes/ISO-3166-Countries-with-Regional-Codes (cit. on p. 34).

[ES92] S. Egbert, T. Slocum. “Exploremap - an Exploration System For Choropleth Maps”. In: *Annals of the Association of American Geographers* 82.2 (June 1992), pp. 275–288. ISSN: 0004-5608. DOI: 10.1111/j.1467-8306.1992.tb01908.x (cit. on p. 24).

[Fle18] T. Flemisch. “Effects of Different Display Form Factors on InfoVis Applications: Exploring Selection Management and Brushing and Linking for Mobile Cross-Device Interaction”. PhD thesis. Masterarbeit, Dresden, Technische Universität Dresden, 2018, 2018 (cit. on p. 28).

[Fou98] M. Foundation. *ResizeObserver*. 1998. URL: https://developer.mozilla.org/en-US/docs/Web/API/ResizeObserver (cit. on p. 36).

[GBWI17] P. Goffin, J. Boy, W. Willett, P. Isenberg. “An Exploratory Study of Word-Scale Graphics in Data-Rich Text Documents”. In: *IEEE Transactions on Visualization and Computer Graphics* 23.10 (2017), pp. 2275–2287. DOI: 10.1109/TVCG.2016.2618797 (cit. on p. 25).

[Gir92] E. R. Girden. *ANOVA: Repeated measures*. 84. sage, 1992 (cit. on p. 44).

[HAB+21] T. Horak, W. Aigner, M. Brehmer, A. Joshi, C. Tominski. “Responsive visualization design for mobile devices”. In: *Mobile Data Visualization* (2021), pp. 33–66 (cit. on p. 25).

[Hol79] S. Holm. “A simple sequentially rejective multiple test procedure”. In: *Scandinavian journal of statistics* (1979), pp. 65–70 (cit. on p. 44).

[LHD18] R. Langner, T. Horak, R. Dachselt. “VISTILES: Coordinating and Combining Co-located Mobile Devices for Visual Data Exploration”. In: *IEEE Transactions on Visualization and Computer Graphics* 24.1 (Jan. 2018). IEEE VIS Conference, Phoenix, AZ, OCT 01-06, 2017, pp. 626–636. ISSN: 1077-2626. DOI: 10.1109/TVCG.2017.2744019 (cit. on pp. 27, 28).

[LSBD21] R. Langner, M. Satkowski, W. Büschel, R. Dachselt. “MARVIS: Combining Mobile Devices and Augmented Reality for Visual Data Analysis”. In: CHI ’21. Yokohama, Japan: Association for Computing Machinery, 2021. ISBN: 9781450380966. DOI: 10.1145/3411764.3445593. URL: https://doi.org/10.1145/3411764.3445593 (cit. on p. 28).

[Mac86] J. Mackinlay. “Automating the Design of Graphical Presentations of Relational Information”. In: *ACM Trans. Graph.* 5.2 (Apr. 1986), pp. 110–141. ISSN: 0730-0301. DOI: 10.1145/22949.22950. URL: https://doi.org/10.1145/22949.22950 (cit. on p. 31).

[Mau40] J. W. Mauchly. “Significance test for sphericity of a normal n-variate distribution”. In: *The Annals of Mathematical Statistics* 11.2 (1940), pp. 204–209 (cit. on p. 44).

[Mor19] E. S. M. Morote. *Pinia The intuitive store for Vue.js*. 2019. URL: https://pinia.vuejs.org/ (cit. on p. 33).

[Mot14] H. Motulsky. *Intuitive biostatistics: a nonmathematical guide to statistical thinking*. Oxford University Press, USA, 2014 (cit. on p. 44).

[RN11] D. Rey, M. Neuhäuser. “Wilcoxon-signed-rank test”. In: *International encyclopedia of statistical science*. Springer, 2011, pp. 1658–1659 (cit. on p. 45).

[Rob07] J. C. Roberts. "State of the art: Coordinated & multiple views in exploratory visualization". In: *CMV 2007: fifth international Conference on Coordinated & Multiple Views in Exploratory Visualization, Proceedings*. Ed. by G. Andrienko, J. Roberts, C. Weaver. International Conference on Coordinated and Multiple Views in Exploratory Visualization 2007, Zurich, Switzerland, JUL 02, 2007. 2007, pp. 61–71. ISBN: 978-0-7695-2903-5. DOI: 10.1109/CMV.2007.20 (cit. on p. 15).

[SS16] R. Sadana, J. Stasko. "Designing Multiple Coordinated Visualizations for Tablets". In: *Computer Graphics Forum* 35.3 (June 2016). 18th Eurographics/IEEE VGTC Conference on Visualization, Groningen, Netherlands, JUN 06-10, 2016, pp. 261–270. ISSN: 0167-7055. DOI: 10.1111/cgf.12902 (cit. on p. 28).

[SW65] S. S. Shapiro, M. B. Wilk. "An analysis of variance test for normality (complete samples)". In: *Biometrika* 52.3/4 (1965), pp. 591–611 (cit. on p. 44).

[THHS16] G. Tschinkel, R. Hafner, P. Hasitschka, V. Sabol. "Using Micro-Visualisations to Support Faceted Filtering of Recommender Results". In: *2016 20th International Conference Information Visualisation (IV)*. 2016, pp. 318–323. DOI: 10.1109/IV.2016.77 (cit. on p. 27).

[Tsc20] G. Tschinkel. "A recommendation dashboard utilising micro -filter visualisations: An evaluation and investigation of applicability on mobile devices". In: *Graphical Models* 110 (July 2020). ISSN: 1524-0703. DOI: 10.1016/j.gmod.2020.101074 (cit. on pp. 27, 28).

[Tuf06] E. R. Tufte. *Beautiful evidence*. Graphis Pr, 2006 (cit. on p. 25).

[Wag05] M. M. Wagner-Menghin. "Binomial test". In: *Encyclopedia of statistics in behavioral science* (2005) (cit. on p. 45).

[Wer23] M. Wertheimer. "Laws of organization in perceptual forms. First published as Untersuchungen zur Lehre von der Gestalt II". In: *Psychologische Forschung* 4 (1923), pp. 301–350 (cit. on pp. 19, 21, 31).

[YKSJ07] J. S. Yi, Y. a. Kang, J. Stasko, J. Jacko. "Toward a Deeper Understanding of the Role of Interaction in Information Visualization". In: *IEEE Transactions on Visualization and Computer Graphics* 13.6 (2007), pp. 1224–1231. DOI: 10.1109/TVCG.2007.70515 (cit. on pp. 20, 23).

[YM14] E. You, E. S. M. Morote. *Vue Router The official Router for Vue.js*. 2014. URL: https://router.vuejs.org/ (cit. on p. 33).

[You14] E. You. *The Progressive JavaScript Framework*. 2014. URL: https://vuejs.org/ (cit. on p. 33).

[You19] E. You. *Vite Next Generation Frontend Tooling*. 2019. URL: https://vitejs.dev/ (cit. on p. 33).

[YRH+16] A. Z. Yu, S. Ronen, K. Hu, T. Lu, C. A. Hidalgo. "Pantheon 1.0, a manually verified dataset of globally famous biographies". In: *Scientific data* 3.1 (2016), pp. 1–16 (cit. on pp. 33, 34).

All links were last followed on May 6, 2023.

Declaration

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

---

place, date, signature