HUMAN-DATA INTERACTION IN LARGE AND HIGH-DIMENSIONAL DATA

by

Saman Amirpour Amraii

B.Sc. in Software Engineering, Amirkabir University of Technology, 2006

M.Sc. in Artificial Intelligence and Robotics, University of Tehran, 2009

M.Sc. in Intelligent Systems Program, University of Pittsburgh, 2013

Submitted to the Graduate Faculty of

the Kenneth P. Dietrich School of Arts and Sciences in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2017

UNIVERSITY OF PITTSBURGH

KENNETH P. DIETRICH SCHOOL OF ARTS AND SCIENCES

This dissertation was presented

by

Saman Amirpour Amraii

It was defended on

November 27th 2017

and approved by

Dr. Michael Lewis, School of Computing and Information, University of Pittsburgh

Dr. Christian Schunn, Department of Psychology, University of Pittsburgh

Dr. Yu-Ru Lin, Department of Informatics and Networked Systems, University of Pittsburgh

Randy Sargent, M.Sc., CREATE Lab, Carnegie Mellon University

Dissertation Advisors: Dr. Michael Lewis, School of Computing and Information, University of Pittsburgh,

Dr. Illah Nourbakhsh, Robotics Institute, Carnegie Mellon University

HUMAN-DATA INTERACTION IN LARGE AND HIGH-DIMENSIONAL DATA

Saman Amirpour Amraii, PhD

University of Pittsburgh, 2017

Human-Data Interaction (HDI) is an emerging field which studies how humans make sense of large and complex data. Visual analytics tools are a central component of this sensemaking process. However, the growth of big data has affected their performance, resulting in latency in interactivity or long query-response times, both of which degrade one's ability to do knowledge discovery. To address these challenges, a new paradigm of data exploration has appeared in which a rapid but inaccurate result is followed by a succession of gradually more accurate answers. As the primary objective of this thesis, we investigated how this incremental latency affects the quantity and quality of knowledge discovery in an HDI system. We have developed a big data visualization tool and studied 40 participants in a think-aloud experiment, using this tool to explore a large and high-dimensional data. Our findings indicate that although incremental latency reduces the rate of discovery generation, it does not affect one's chance of making a discovery per each generated visualization, and it does not affect the correctness of those discoveries. However, in the presence of latency, utilizing contextual layers such as a map result in fewer mistakes while exploring higher-dimensional visualizations lead to more incorrect discoveries. As the secondary objective, we investigated what strategies improved a subject's performance. Our observations suggest that successful participants explore the data methodically, by first examining simple and familiar concepts and then gradually adding complexity to the visualizations, until they build a correct mental model of the inner workings of the tool. With this model, they generate several discovery patterns, each acting as a blueprint for forming new insights. Ultimately, some participants

combined their discovery patterns to create multifaceted data-driven stories. Based on these observations, we propose design guidelines for developing HDI platforms for large and high-dimensional data.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

I am indebted to many mentors and friends who made this work possible through their generosity and support. I would like to thank my thesis advisors, Prof. Illah Nourbakhsh and Prof. Michael Lewis, who guided me in every step of this path. Their wealth of experience and depth of knowledge were imperative in the formation and development of this dissertation. I would also like to thank Randy Sargent who was an invaluable source of wisdom in the design and implementation of the visual analytics system. I would also like to thank the members of the Intelligent Systems Program at University of Pittsburgh and my colleagues at CREATE Lab who enthusiastically assisted me in testing the software and provided feedback on designing the experiments. I would also like to thank the Google corporation for an unrestricted gift to the Carnegie Mellon CREATE Lab that, in part, supported this work.

I would like to dedicate this work to my mother, Nadereh, who taught me how to dream, to my father, Sasan, who taught me how to think, to my sister, Sara, who taught me how to smile, and to the love of my life, Ziba, who taught me how to never give up.

1.0 INTRODUCTION

Making sense of data is the core of scientific discovery. It is then no surprise that we have become obsessed with collecting and analyzing an ever-increasing amount of data. However, the sheer size and complexity of such datasets have made us reach the limits of our data analytics tools. Sometimes, asking the simplest questions may take hours or days of computation. An even more daunting task is to grasp the complex internal relationships of a high-dimensional dataset with thousands of variables. Such challenges have motivated researchers to call for the development of *Human-Data Interaction* (HDI) as a new field of study, with the goal of answering how humans make sense of data, especially when it becomes large and complex.

One of the earliest accounts of HDI is presented in Elmqvist's work on embodied human-data interaction [26]. He defines HDI as *"the human manipulation, analysis, and sensemaking of large, unstructured, and complex datasets."* In a more recent effort, Mortier et al. [54] discuss HDI as an emerging field and define it as the interaction between humans and the analysis of large datasets. Therefore, there is both a focus on the data itself and also on the analytics and the algorithmic part. HDI pursues two sets of questions where it first explores how to design an HDI system that would facilitate knowledge discovery, and it then discusses how to build a system that can provide these desirable design guidelines in the context of working with large and complex datasets. The design aspect of the HDI is tightly coupled with the human's cognitive abilities. Here, HDI addresses questions such as how can we make a dataset presentable to humans, how can people navigate in that space and find what they are looking for, and eventually how can they draw correct conclusions based on the patterns they observe. Overall, HDI researchers investigate how humans build a mental model of the data and how they utilize that model to form new hypotheses and eventually

make discoveries.

Humans are inherently skillful in discerning visual patterns. Furthermore, eyes provide the highest bandwidth of information to the brain. As a result, visual analytics tools are a fundamental part of most HDI systems. Nevertheless, unique cognitive characteristics of the human vision system (discussed in Chapter 2), mandate a set of stringent requirements for an HDI system to be effective. At the core of such cognitive characteristics is our limited working memory. Not only we seek highly interactive tools to understand a data better, but we also need low latency between the time we start asking a question till when we see a pattern on the screen, representing the answer. Achieving such design requirements is challenging, specifically when one is dealing with large and complex data. Consequently, the second aspect of the HDI research focuses on designing and implementing software architectures that on the one hand, are capable of providing an interactive and fluid experience and on the other hand, are scalable to large data sizes. One example of such a system that has been influential on this thesis is TimeMachine [67][1].

TimeMachine is a web-based tool for exploring large videos. It has been utilized to ingest satellite imagery of a terapixel scale, and then convert them into a hierarchy of smaller videos, each representing the data at a particular location and a particular zoom level. TimeMachine's interface then allows a user to explore these videos, seamlessly moving from one to another, without the user noticing that her view is composed of multiple videos stitched together and replaced on-demand based on zooming in and out actions. Such an experience allows the user to explore any high-resolution video interactively and with no latency. Also, the client software only has access to a small portion of the data at any time, while the server stores the raw data and streams the required videos over the net.

Several experts have used TimeMachine to explore a wide array of datasets from cosmology and planetary science, to environmental protection. These collaborations further illuminated the vital role of keeping humans in the loop of data exploration and knowledge discovery, as human expertise coupled with the right computational tools leads to knowledge acquisition abilities that are hard if not impossible to achieve by humans or machines alone.

[1] A version of the TimeMachine, featuring a time-lapse of satellite imagery of the entire planet Earth over the past 30 years, is accessible online at https://earthengine.google.com/timelapse/.

For example, blindly applying mathematical models to find a pattern in the data hardly yields any desirable results. Instead, experts usually first visualize the data to get a feel of it and then they apply the relevant models based on their experience. Also, the contextual knowledge of the expert is often critical in drawing the right conclusions, as the data under study may not be sufficient for deriving the intended discoveries. Furthermore, mathematical models can sometimes act like a black box, hiding the reasons behind a conclusion. Again, human expertise in conjunction with powerful data exploration tools can act as a bridge between the patterns one can observe and the real cause behind such patterns. Overall, these benefits further acknowledge the importance of keeping the humans in the loop and hence motivate designing better HDI tools to expand our knowledge discovery capabilities.

Inspired by the effectiveness of TimeMachine in knowledge discovery, this thesis started as an effort to expand the capabilities of TimeMachine beyond raster data and allow researchers to visualize and explore vector datasets in a similar manner. An initial outcome of this effort was Explorable Visual Analytics (EVA), a web-based tool for visualizing large and high-dimensional numeric data, which is described in detail in Chapter 3. Several researchers used EVA in their respective fields to explore datasets such as demographics [66], twitter feed, and product reviews. These collaborations demonstrated the effectiveness of HDI tools for knowledge discovery but more importantly revealed a lack of accessible tools for exploring big data. Consequently, a repeated request from many experts was to increase the capabilities of EVA to accommodate even larger datasets. However, achieving this goal was challenging as supporting larger datasets while keeping the tool interactive and responsive required a redesign of how data was ingested and presented to the user. Another issue observed during the testing of EVA was the inconsistency of the performance of experts with the tool. Some experts were utilizing the tool more fluidly than others and hence were able to gain a better outcome in the quality and quantity of the discoveries they made. Therefore, a second question was formed to find out why some users were more successful in using the tool and if it would be possible to redesign EVA in such a way that would facilitate knowledge discovery for a wider audience. As a result, the objectives of this thesis were shaped to find a suitable architecture for implementing a scalable HDI tool and proposing design guidelines required for supporting knowledge discovery with such a tool.

Latency is the core problem of building a scalable visual analytics tool. As human cognition demands sub-second interaction response time and only tolerates a few seconds of query-response delay, all proposed solutions focus on reducing latency while expanding the data ingestion capacity of the system. Section 2.5 provides an overview of the three classes of approaches proposed for achieving this goal. The first class of solutions reduces latency by leveraging more powerful hardware such as Graphical Processing Units or by increasing the utilization of the available hardware, for example by precomputing partial results even before the user sends any queries. These *instantenous* solutions aim to provide a rapid and accurate result, although to the cost of using more expensive hardware. Also, to precompute their answers, they often put limitations on what a user can ask from the system. The second class of solutions is championed by researchers such as Fekete et al. [28], where they reduce latency by lowering the accuracy of their results. These *approximate* methods do not need expensive hardware and can provide interactive data exploration on big data. Nevertheless, inaccurate results are only acceptable for the exploration phase, and after analysts find an area of interest, they often require accurate results to make informed decisions. As a result, a third class of *gradual or incremental* methods have emerged that reduce latency by initially providing an inaccurate and fast answer, but then gradually improve the accuracy over time [30]. These systems do not require expensive hardware and do not put limitations on the questions one can ask of the data. More importantly, they can scale very well with large data sizes. However, such systems are still in their infancy, and general purpose gradual data analytics tool are not available for public use.

To choose the right architecture for scaling EVA, we have focused on gradual and instantaneous methods. Considering the lower cost of deployment and maintenance of gradual systems and also their superior scalability, they seem to be an appropriate choice for any big data HDI system. However, it is not evident whether they provide the same data exploration experience and can help their users reach the same level of knowledge discovery attainable in instantaneous systems. Therefore, before investing in designing and building such gradual systems, we should first investigate how the latency introduced in gradual systems affects the quality and quantity of knowledge discovery. Consequently, the primary objective of this thesis is to answer *how the introduction of incremental latency in a Human-Data Interac-*

tion system operating on large and high-dimensional data affects the quantity and quality of knowledge discovery performed by its users?

The central challenge in designing an HDI system for big data is the vastness of the search space. At any moment, only a small portion of the data can be fully visualized. Moreover, data looks different depending on the scale one looks at it, and some patterns only make sense in particular zoom levels. The enormity of the search space adds extra complexity on how one can navigate such a space and find an area of interest. Additionally, large datasets are often multi-dimensional as well which means they have an astronomical number of possible projections, each leading to a different dissection of the data. Considering these complexities, the design of an HDI system can dramatically affect the performance of its users. During the initial field tests of EVA, users demonstrated a wide range of mastery over how they navigated the space of data and ultimately how many discoveries they had. This diverse behavior prompted us to investigate what aspects of participants' behavior lead to a better utilization of the tool and if insights learned from this analysis can pave the way for building more effective big data HDI systems. Consequently, the secondary objective of this thesis is to answer *how the data exploration strategies utilized by high performer users—those who generate the highest number of correct discoveries—differ from the ones exercised by low performers—those with the highest number of incorrect discoveries—and as a result, what design guidelines can be incorporated into a Human-Data Interaction system to facilitate knowledge discovery?*

To answer the questions posed in this thesis, a user study was performed where 40 participants utilized EVA to explore a large and high-dimensional data on demographics. Participants were encouraged to make as many discoveries as they can via visualizing and interacting with this data. Chapter 3 provides a detailed description of the experiment. The general design of this experiment was inspired by the guidelines proposed by Isenberg et al. [40] on evaluating visual analytics systems. Their work is based on a systematic review of hundreds of visualization papers over the course of a decade to assess what techniques are commonly used to measure the effectiveness of a visual analytics tool. The majority of these techniques can be categorized into two classes of *user experience* and *user performance* evaluation metrics. User experience metrics consist of interviews with the users on their

subjective opinions about the tool and the experiment while user performance metrics include objective measurements of the effects of a particular feature of the visualization tool on the performance of the user. As such, we chose a think-aloud protocol where participants spoke their thought process loudly and in particular mentioned whenever they had made a discovery. Their speech and interaction with the tool were recorded for further analysis and marking of the discovery events. Also, all user interactions with the tool were recorded to provide the required data for further objective assessments. Participants also filled out two questionnaires, one before and one after the experiment to collect their familiarity with the data and visual analytics tool and also to investigate how their exposure to the tool has affected their beliefs about visual analytics tools. The independent variable of this study was latency. Hence, half of the participants worked with an altered version of EVA which simulated an incremental visualization system and the other half were kept as the control group and worked with the original version of EVA which does not impose any delays on query-response times. The dependent variables of this study are quantity and quality of discoveries, in which the latter measure consists of the correctness, depth, breadth, and the uniqueness of discoveries. Chapters 4 presents the results of the experiment and discusses the observed differences between the two groups. Furthermore, we select four participants with the highest and lowest performance measures from each group to investigate what strategies were utilized by each one and how those strategies affected their chance of making a discovery. In Chapter 5, we interpret these findings, discuss how incremental HDI systems affect the process of knowledge discovery, and propose suggestions on the applicability of these emerging data analytics methods. Additionally, we will examine the insights learned from our qualitative user assessments and provide guidelines for designing HDI tools dedicated to exploring large and complex data.

2.0 RELATED WORKS

This chapter is organized as follows. Section 2.1 discusses the definitions and architectures proposed on the topic of *visual analytics*. In Section 2.2, we look at the body of research on the characteristics of the human visual system. Understanding these characteristics is critical for designing effective visual analytics systems. In Section 2.3, we investigate studies on the topic of *knowledge discovery*, in particular, the role of visualization in making sense of data. Also, as the main focus of this thesis is on knowledge discovery in high-dimensional and large data, we look at the proposed approaches for visual exploration of such challenging datasets in Sections 2.4 and 2.5. Finally, Section 2.6 discusses the emerging field of *Human-Data Interaction* (HDI), where researchers investigate how people interact and make sense of information in a world where data is ubiquitous and is becoming more complex every day.

2.1 VISUAL ANALYTICS

Casner [17] argues that visualization is a tool for facilitating information-processing tasks. It helps in two ways, first, by simplifying the search process where the user finds patterns in the visual space, and second, by assisting in the computation where the user performs simple computations with her eye. Ware [84] expands on these benefits and suggests that visualization can also help in coping with various scales of the data, making it easier for the user to understand both large and small datasets. Also, it helps in forming and testing new hypotheses, effectively empowering the user to ask questions that she did not have before visualizing the data.

Keim et al. [41] discuss the topic of visual analytics from the perspective of information

overload. They argue that information is getting increasingly more complex and hence the goal of visual analytics is to simplify the data and extract meaningful knowledge from it. In this quest, humans and machines are combining their unique capabilities to get effective results. They then define visual analytics as:

> Visual analytics combines automated analysis techniques with interactive visualizations for an effective understanding, reasoning and decision making on the basis of very large and complex data sets.

In a follow-up book, Keim et al. [42] expand on this definition and argue that in visual analytics, experts pursue knowledge discovery by leveraging their contextual knowledge, and benefiting from powerful information processing tools such as databases, all while representing and manipulating the data in an interactive visual space.

Before the user can benefit from visual analytic systems, the data must be processed and prepared. The visualization pipeline consists of three basic elements: (1) gathering and cleaning the data, (2) processing it through actions such as scaling, filtering, mapping, ... and finally (3) visualizing it where we can interact with the data in the visual space [84, 51].

Shneiderman [72] introduced the mantra of visual analytics workflow: "overview first, zoom and filter, then details-on-demand". The user first creates a visual representation of the data, then interacts with it through actions such as zooming and filtering, and if needed, the system then provides more details. For example, when the user zooms in on a part of a map, the resolution improves. Additionally, Schneiderman suggests other interaction mechanisms such as relation, history, and extraction. Relation provides links among items (such as arrows). History provides support for undo and replay, therefore facilitating progressive refinements of the visualization through trial and error. Extraction acts similar to filtering by selecting a subset of visualization and is mainly used in sharing the visualization as it limits the scope of exploration, letting the new user familiarize herself with the visualization space.

Researchers have tried to provide a more mathematical foundation for the visualization process. This can, in particular, be helpful in automatizing the visualization pipeline in which a computer can easily suggest and create correct and meaningful visual representations of the data. One of the seminal works in this area is presented by Wilkinson [87, 86]. He introduces

a grammar for defining a visualization pipeline. This grammar works in a seven-dimensional space. It starts with raw data. The first step, *Variables*, turns the raw data into a set of variables. The next three steps, *Algebra*, *Scales*, and *Statistics*, analyzes this set of variables and turn them into new variables ready for visualization. The final three steps—*Geometry*, *Coordinates*, and *Aesthetics*—work in the visual space and turn the variables into the final visual representation. This pipeline has been used extensively in several current visualization tools.

Many advanced visual analytics systems have been designed and adopted over the years. Ganeshapillai et al. [32] introduce a powerful example of such a system. Their visual analytics tool has three main components. On the backend, a database ingests and serves the data. Simple analytics such as filtering and aggregations are performed by the database layer as well. On the frontend, they use D3 [14]—a widely adopted visualization library—to visualize the data in a web-browser. Many of the visual analytics systems follow a similar architecture of connecting a frontend visualization layer to a data management backend, yet Ganeshapillai et al. introduce an additional layer for machine learning (ML) as well. This layer is in charge of processing trends, similarities, outliers, and other pattern recognition tasks. Such analytical capabilities are invaluable in making sense of large and complex datasets. Nevertheless, traditional databases are not designed to perform ML analytics, and hence few visual analytic systems are offering such features. The computationally heavy nature of these calculations has made them a challenging value proposition, especially given the high interactivities demanded by visual analytic systems.

2.2 HUMAN VISUAL SYSTEM

Human vision and perception have evolved to recognize a particular class of patterns. Any visual analytics system should be aware of these characteristics and obey the corresponding design guidelines to be useful for human utilization. There is an extensive body of research on how people make sense of visualizations and what features are more appropriate for presenting information. One of the seminal works on this topic is a paper by Cleveland

and McGill [23]. They provide a taxonomy for distinct visual features and demonstrate people perceive these features with different levels of accuracy. A recent book by Cairo [15] lists these features and their order (from highest accuracy to lowest) as length, direction, angle, area, volume, curvature, shading, and color saturation. Mackinlay [51] builds on these findings and suggests that a visualization system should assign more essential aspects of the data to visual features with the highest level of perceived accuracy. For example, if we intend to show the difference between two values, we should use shapes with different lengths rather than different colors. Casner [17] further extends on these guidelines and suggests that an effective visualization system should also consider the task at hand. Some visual features may be more appropriate for a particular kind of task. Herman et al. [36] add predictability as another important feature of a usable visual analytics system. They argue that if a user slightly changes the data or the mapping function from the data to the visual space, the visualization should still look similar and should not change drastically. This feature is essential in creating and preserving a mental model of the data and its representation in the visual space.

According to Ware [84], there are nine visual dimensions which we can use in a visualization. These separate visual channels are position, color, shape, motion, and visibility span (blinking). By assigning different data values to these visual dimensions, we can visualize up to nine dimensions at the same time. Nevertheless, our cognitive abilities are limited in the number of resolvable steps we can distinguish in each visual dimension. He mentions we can usually perceive four bits of information in each visual dimension and therefore our visual space has $4^8 = 65,536$ distinguishable states (considering eight visual dimensions: three for the position, three for motion, and two for color and shape). Even then, because perceiving a complex visual object with multiple visual dimensions is not an easy task for our brain, we end up with a far more limited number of effective distinguishable visual representations. Other researchers pose similar limits on the number of resolvable steps in the visual dimensions. For example, Fry [31] argues our brain can only process up to five colors in an image. Our ability to perceive motion may be even more limited. Although it can be a beneficial visual representation, we are more adept in perceiving relative motion rather than absolute moving values [84]. Also, our brain is hard-wired to follow moving objects, therefore if a

visualization consists of moving patterns, the user may not pay attention to other visual aspects of it [15]. We are also limited in our ability to comprehend and remember a complex pattern [84]. This is because we can only process a limited amount of information. For example, Cairo [15] suggests using abstract pictures for visualizing how something works, as realistic pictures convey more information to the brain and distract the audience from the intended message.

A visual query is the core of visual analytics. We form such queries by creating a mapping function which transforms abstract numeric data into some observable object in the visual space. Although visual queries are easy to understand, they carry much information. Because our working memory capacity is insufficient, Ware [84] argues that we cannot keep many visualizations in our mind. Hence, to decrease the processing load of our working memory, we should be able to switch between different visualizations rapidly. This *responsiveness* is a fundamental feature of an effective HDI system. The user should be able to generate a new question, visualize it quickly, assess the validity of her hypothesis and then go to the next question. Any lag in this pipeline and the user may lose her train of thought. Ware also suggests other techniques that can improve our limited memory capabilities such as having side-by-side windows for comparing two different visualizations or having an overview window for showing the general outline of the visualization while the user is exploring a smaller section of the picture.

There has been extensive research on the way visual searches work. On the most fundamental level, our eyes are more sensitive to particular visual features. These basic features are called *pre-attentive* features because our brain processes them unconsciously [31]. Recognizing pre-attentive features happens quickly—less than 10ms vs. the usual 40ms for other visual features. They include variations in shade, patterns such as size and orientation, the proximity between objects, similarity or connectedness, continuity such as smooth vs. angular lines, and closures [15]. One layer above these basic features, we have *Gestalt laws* [84]. These rules dictate which visual representations would look like a meaningful pattern. They include proximity, similarity, connectedness, continuity, symmetry, closure, relative size, ... For example, if we draw a contour around some points, it is very natural to perceive them as belonging to the same group. To find patterns hidden in the data, we need

to map them into visualizations which obey these laws.

Ruchikachorn and Mueller [65] present an excellent example of utilizing such fundamental features in a visual analytics tool. Their goal is to expand a user's visual vocabulary, essentially teaching users new visual representations of the data without any instructions or manual. They achieve this via learning-by-analogy where the system starts with a familiar visualization such as a line chart, and then gradually morphs that into a more complex (and new) chart such as a spiral chart. By starting from a simpler visual representation, they leverage innate pattern recognition capabilities of their participants. Then, by gradually transforming the visualization into something more complicated, they utilize the Gestalt laws to expand a user's visual vocabulary all without needing any external interventions.

Cairo [15] provides another set of high level approaches we utilize to identify objects: (1) feature-based recognition like recognizing a face by recognizing pre-attentive features similar to eyes, mouth, ..., (2) component-based recognition like observing the components of class and then identifying the object, and (3) configuration-based recognition where we pay attention to the way the components are arranged and connected in order to recognize the object. Having these visual patterns does not guarantee that the user can see them. Often, she needs to explore a visualization from different perspectives until she can see those patterns. This is why navigation is another important aspect of visual search. Ware [84] provides a comparative list of different navigation methods and argues that a flying camera is one of the easiest ways of navigating through a visual space. Also, the author emphasizes that navigation should be fast to facilitate efficient thinking: *"When simple pattern finding is needed, the importance of having a fast, highly interactive interface cannot be emphasized enough. If a navigation technique is slow, then the cognitive costs can be much greater than just the amount of time lost, because an entire train of thought can become disrupted by the loss of the contents of both visual and nonvisual working memories."* For example, the visualization system should provide at least two frames per second or in a 3-dimensional point cloud visualization, Shneiderman [74] argues we need a visual feedback within at least 100 milliseconds to feel we are in control of the data. As we will see in Section 2.5, many of the challenges we face in building visual analytics systems for big data are to keep up with this high interactivity requirement.

2.3 KNOWLEDGE DISCOVERY

Waters [85] provides a detailed overview of the role of exploration in knowledge discovery. He mentions that traditionally, scientific experiments are guided by theory. This *theory-driven* approach indicates that we use theoretical knowledge to come up with new hypotheses, and then we design experiments to test them. However, this view has been challenged by the emergence of big data and its pivotal role in knowledge discovery. In the new perspective, some scientific endeavors are not necessarily based on concrete theories. Here, instead of having theory-driven experiments, we have *theory-informed* ones. According to Waters, *"Sometimes, experimentation is exploratory in nature."* Hence, exploratory experimentation is a new ingredient of scientific discovery. In particular, exploration becomes important when the theory is not developed enough to articulate the correct set of questions to ask. In such cases, exploration becomes a necessary step in forming the initial hypotheses which would later on guide experiment designs and eventually lead to discoveries.

Considering the exploratory nature of knowledge discovery, visual analytic tools can play an important role in facilitating this process. Shneiderman and Plaisant [75] explain this role. First, users start with a hypothesis. They then generate a visualization and look for patterns or outliers. Then according to these partial results, they refine their hypotheses and repeat the process. Sometimes, users may find unexpected results which may change their theories, lead to new paths of inquiry, and eventually help them to make a discovery. Essentially, visualization tools make the data more tangible. They provide a space to form a hypothesis via a visual query and then test it by a visual search. Fry [31] explains this process in more detail and proposes seven steps for making knowledge discovery with visual systems. First, we should acquire the data. Then we have to parse it and make it machine-readable. This is then followed by filtering in which we select a subset of the data that is relevant to our work. We then mine for useful information which usually means some mathematical transformation. The results are then represented in an initial visualization. Finally, we refine the results, interact with the visualization, and improve it by redoing the previous steps until we extract or discover the desired knowledge. All of these steps are transforming raw data into some visual patterns that are easy to perceive. Consequently, a powerful visual

analytics system should be tailored according to human perception limitations and be aware of our inherent strategies for making sense of data.

Klahr and Dunbar are two of the pioneers of understanding how humans discover new insights. In a series of seminal works [45, 77] they propose that knowledge discovery is a search in the two dimensional space of *hypothesis space* and *experiment space*. How we search this space shapes the process of hypothesis generation, how we design our experiments, and how we eventually evaluate our hypotheses. The way we search our hypothesis space is shaped by our prior knowledge (contextual knowledge) and the results from our experiments. Also, our search in the experiment space is shaped by our current hypotheses (to test them and evaluate our hypotheses) or by our need to generate necessary information for forming new hypotheses. They also suggest that people utilize two main strategies for searching this two-dimensional space:

1. **Bottom-Up Approach:** They call these participants *experimenters* as they are data-driven. They start with a hypothesis and then perform several experiments to rule out all other possible hypotheses. As they are highly dependent on the experiments, they may not reach certain parts of the hypothesis space.
2. **Top-Down Approach:** They call these participants *theorists* as they are hypothesis-driven. They only perform an experiment after they have formed a hypothesis. They then perform the experiments to test those hypotheses, and if they see contradicting evidence, they change their previous hypothesis a little bit and then perform a new experiment. Theorists often need fewer experiments to reach the correct conclusion. Klahr and Dunbar believe that hypothesis-driven subjects are more successful than data-driven ones.

In a follow-up work, Schunn and Klahr [71] extend this model and argue that scientists navigate in a four-dimensional space to extract meaning from their observations. The first dimension in this paradigm is called *data representation*. This is where an abstract representation of data is being formed from a set of features. The second dimension is *hypothesis space*. Here, the scientist generates new assumptions on the possible causal relationships. Then she moves to the third dimension of *experimental space* to test those hypotheses. It

should be noted that the experiments themselves live in an experimental framework that defines the boundaries of valid experiments and expectable outcomes. Therefore, the fourth dimension is *experimental paradigm space* where the scientist can choose a completely different class of experiments for her task.

Chang et al. [19] differentiate between two main types of insights. The first category is the *spontaneous insight* which is mostly studied by cognitive scientists. The other type is *knowledge-building* insight which is mostly investigated by visual analytics researchers. These two are inherently different, and they happen via different processes as well:

1. **Knowledge-Building Insight (KI):** This is usually the type of insight discussed in information visualization community. It is a gradual process upon which the scientist step by step solves the problem. The general process includes four steps: provide an overview, adjust, detect patterns, and finally match the mental model. This insight is more like a unit of knowledge than being an *aha* moment as it is in the spontaneous insight category. Brain scans show that KI is related to areas of the brain that represent tightly related semantic networks.
2. **Spontaneous Insight (SI):** This is usually when a scientist experiences an *aha* or *eureka* moment. It often happens after people have worked on the usual solutions and have exhausted the knowledge-building insight and methods. It is interesting that the usual ways of problem-solving often prohibit SI. SI best happens in a relaxed state. Also, it is still not obvious how SI happens. Some say it happens when we try to solve a problem, then we fail, then we look at it from a different perspective until we finally solve it. Interestingly, people often do not know how they reached SI and what steps lead them to the final discovery. Brain scans show that SI is related to areas of the brain that represent less clearly relevant information and weak semantic networks.

Silberschatz and Tuzhilin [76] introduce the concept of interestingness as a prerequisite for the insight generation process. They suggest two subjective measures as for why a user may find a pattern interesting:

1. **Unexpectedness:** Where a pattern is considered interesting if it surprises the user.

2. **Actionability:** Where a pattern is considered interesting if the user acts on it, leading to meaningful decisions.

They believe actionable insights are more interesting to users although they mention that there is a large intersection between actionable and unexpected knowledge, and hence they are a good approximation of each other.

One of the early works in understanding how knowledge discovery happens is performed by Qin and Simon [62]. This work is especially important as it compares successful participants with unsuccessful ones and then considering the strategies used by these participants, they provide guidelines for an effective knowledge discovery process. They have based their work on an experiment where participants are given a set of numbers and are asked to find a mathematical formula describing those numbers. Interestingly, these are precisely the numbers used by Kepler to derive his famous law of planetary motions. However, as the participants do not know this, their knowledge discovery process is purely data-driven and deprived of any context. As Qin and Simon explain, the motivation behind this experimental design is that in scientific discovery, *"regularities of data came first; causes had to be shaped to fit them."* This is a challenging experiment, as participants often do not know where the goal is, or how close they are to the solution, even sometimes getting very close to the answer and then missing it.

In general, participants mostly used simple functions to find patterns in the data, and almost all of them used visualizations such as diagrams and scatter plots to look at the data first. Successful participants followed three main steps: (1) first, they allocated some time to understand the numbers better, for example, by using some visualizations, (2) then they did a breadth search where they applied several classes of different functions to find some potential candidates, and (3) finally they performed a depth search where they focused on the most promising function types and looked for the parameters that would make those functions fit the data best. Successful participants relied on feedback loops. They used data to improve upon their hypotheses. They also used abstraction to simplify their search. For example, one of them changed data values to integral numbers to speed up testing various formulas. On the other hand, unsuccessful participants had no systematic approach in searching their hypothesis space, had poor feedback loops where they were not able to obtain information

from their tests, and frequently repeated the same series of tasks, without any apparent goal. In summary, successful participants in knowledge discovery utilized a systematic approach to search for a solution, continually tested their hypotheses and gained feedback from it, and used that feedback to form new questions.

The design of a data exploration tool can deeply affect the knowledge discovery process. Joolingen and Jong [82] propose three steps for the hypothesis generation process:

1. **Variable Identification:** Where the user identifies the variables that she can use in her model.
2. **Variable Selection:** Which acts as a high-level filtering phase where the user selects the variables that might have some relation together.
3. **Relationship Identification:** Where the user defines the general relation that is hypothesized to hold between the selected variables.

In a series of experiments, they measure the effects of imposing a structure on participants' hypothesis generation process, on their overall performance. Their findings suggest that a structured tool can facilitate hypothesis generation; however, it also hinders participants from finding specific relationships between the variables. Hence, an efficient data exploration tool should seek a balance between the degrees of freedom it provides to its users and the scaffolds it enforces on their hypothesis generation process.

A recent work by Guo et al. [35] looks at the topic of knowledge discovery in visual analytics tools. Their goal is to discern which actions lead to insights. To do so, they have designed an insight-based user study using an information visualization application. During the experiment, they collect user interaction logs and videos of participants. Then, they transform those raw level actions (such as user keystrokes and mouse movements) into high-level actions and analyze which combinations of those high-level actions lead to a discovery. The authors argue that this approach makes it easier to find the underlying patterns resulting in a discovery and also helps them to generalize their findings for other visualization tools as well.

They propose seven high-level actions:

1. **Select:** Where the user marks something as interesting.

2. **Explore:** Where the user looks for something else.
3. **Elaborate:** Where the user looks for more details.
4. **Reconfigure:** Where the user creates a different arrangement of variables.
5. **Filter:** Where the user wants to see something conditionally.
6. **Connect:** Where the user looks for related items.
7. **Retrieve:** Where the user wants to see matches to a query.

After coding participants' recorded videos, they measure the most common sequences of actions. They find four prevailing patterns:

1. **Orienting:** reconfigure → explore → elaborate
2. **Locating:** retrieve → elaborate → elaborate, elaborate → retrieve → elaborate
3. **Sampling:** explore → elaborate → elaborate → elaborate, explore → elaborate → elaborate
4. **Elaborating:** elaborate → elaborate → elaborate

Among these four patterns, they find that *Sampling* results in the highest number of insight events. This confirms previous research that suggested discoveries happen when participants start with a broad search, and then investigate a small set of variables in more depth. Guo et al. further suggest that a productive visual analytics system should make it easy for the users to distinguish between sampled entities and those not examined yet, essentially helping the user to separate what they have already explored from new information.

Recently, some researchers are designing visual analytics tools to facilitate computer-assisted knowledge discovery. For example, Wongsuphasawat et al. [89] have implemented a data analytics tool where the system utilizes statistical analysis to suggest appropriate visualizations, leading to a better insight generation performance. In another work, Lehmann and Theisel [47] have designed a system which automatically applies dimensionality reduction algorithms to complex data. The outcome is a selected number of visualizations with fewer overlapping features, effectively shrinking the search space for the participant and speeding up the knowledge discovery process.

Although knowledge discovery is an essential part of science, its ultimate benefit is achieved when we can share that knowledge with others. Knowledge dissemination in science

has been traditionally achieved through papers, lectures, ... However, with the increasing role data plays in scientific discoveries, the traditional means of knowledge communication are not sufficient anymore. In a famous lecture, Jim Gray [37] argues that science started as an empirical endeavor, then through works of people such as Newton it turned into a theoretical practice, then with the emergence of computers it heavily benefited from computational simulations, and now with the rise of big data it is entering the fourth paradigm of data exploration. He then discusses the importance of sharing data. New ways of standardizing and connecting data can immensely benefit scientific discoveries. We can argue that it even forms a foundation for not only shareable data but shareable knowledge. An example of this idea is shown by Kidd et al. [43] where authors investigate visualization and data exploration tools in the field of immunology. They emphasize that there is a tremendous need for new data-centric tools that can be used without specialized training. This, in turn, would allow collaboration among experts. Knowledge dissemination from experts to non-experts is also an essential part of science. Kosara and Mackinlay [46] talk about this aspect through the lens of storytelling. They argue that people can easily remember stories and this can push them to act upon what they learn from data.

Despite the importance of knowledge dissemination, current visual analytics systems have not still fully developed their capabilities for supporting this aspect of the scientific endeavor. Isenberg et al. [40] provide a thorough survey of hundreds of visualization papers over the course of the previous decade, yet there are only a handful of efforts dedicated to collaborative knowledge discovery and none on knowledge dissemination. One of the few tools that has specifically been developed for helping non-experts understand data-backed facts is Gapminder [64]. Rosling et al. argue that although many facts about our world are now available on the Internet for free, few people study and benefit from them. This is because most of this data is presented in *hard to read* numerical table format. Gapminder addresses this issue by presenting information in an interactive and easy to understand visual space. The software has been designed for non-data-expert but technologically savvy people who are familiar with other interactive software such as computer games. Gapminder has been a very successful tool in presenting multidimensional data such as United Nations' information on wealth and health of the nations. Gapmider's success is partly due to its

ease of use and understandable visualizations. It also provides tours similar to Microsoft PowerPoint slides. This makes it a valuable tool for storytelling in which the data expert can accompany his visualizations with other modes of communications such as text and audio to provide context and background knowledge for the data. Another example in knowledge dissemination is presented by Sargent et al. [67] in GigaPan and TimeMachine projects. GigaPan visualizes large pictures while TimeMachine presents ultra high-resolution videos. The first design goal of these tools is to facilitate knowledge discovery. Because they store very large images and videos and represent more details upon zooming on a desired location, they are great tools for finding patterns and new knowledge, especially when the expert is not sure what aspect of the data plays a part in the final discovery. Hence, through saving and loading all the details of a large visual dataset, the scientist can search for the relevant patterns via interacting (e.g., zoom, pan) with the imagery data. For example, Nichols et al. [57] take high-resolution photographs to monitor rangelands and then uses GigaPan to search for relevant information in resource management. In another project, Feng et al. [29] use TimeMachine to explore a massive cosmological simulation where each frame exceeds gigapixel resolution. Besides their capabilities in data exploration and discovery, GigaPan and TimeMachine are designed to support tours. For example, a TimeMachine tour is a spatiotemporal trajectory through the space of the data. In this way, the expert can create a tour that zooms and pans to a series of desired locations and times in a high-resolution video. These tours can be accompanied by annotations too. It is then possible to create a meaningful story which is directly linked to the actual data. Also, the user can switch from the tour mode to exploration mode and engage directly with the data whenever she desires. This approach to knowledge dissemination is quite powerful. It first builds a contextual knowledge around the data through annotations and text. Then, the user learns about interesting facts about a particular dataset by watching some tours. These tours help the user to build a mental model of the data, understand how to explore it, and learn what the possible patterns that can be found in it are. Therefore, tours serve as a starting point for turning the users from a data consumer to an information producer, which happens when the user exits from the tours and starts interacting with the dataset directly.

Recently, researchers are paying more attention to the importance of knowledge dissemi-

nation in visual analytic tools. For example, Chen et al. [20] present a method for suggesting viewpoints in a 3-dimensional modeling environment based on the recorded motions of the modeler. They further argue that providing an appropriate view path can be beneficial to viewers and hence a worthwhile research path. However, due to our limited working memory, creating a viewing path is not sufficient in communicating an acquired mental model from the data, especially if the visual space is very large and high-dimensional. Therefore, future knowledge dissemination tools should also have guidance mechanisms to help the user remember her location in the data space. For example, part of the system presented by Ingram et al. [38] provides both global and local guidance. The user can then better find her way in the data and also navigate between interesting aspects of it. In conclusion, it is worth mentioning that tours can have their pitfalls that a system designer should be aware of. Kosara and Mackinlay [46] support the potentials of storytelling but then warn us that interactive storytelling can distract the focus of the user from the actual story. It is, therefore, an open question how to design a knowledge dissemination system that incorporates interaction in a beneficial manner which actively engages the user and improves her understanding of data while not itself becoming a distraction.

2.4 EXPLORING HIGH-DIMENSIONAL DATA

High-dimensional datasets are inherently hard to visualize (e.g., a 4-dimensional cube). By using color, blinking, and motion, we can visualize up to eight dimensions. However, creating a useful mental model from this visualization is not an easy task. Also, when we face with more complex datasets with hundreds of dimensions, direct visualizations are not feasible anymore. As a result, most high-dimensional visualization systems focus on dimension reduction algorithms or interactive techniques to make sense of complex datasets.

There are two general approaches to visualizing high-dimensional data. In *automated methods*, a predefined mathematical formula reduces data dimensions to a manageable 2 or 3-dimensional space. Many commonly used algorithms such as Multi-Dimensional Scaling (MDS) and Principal Component Analysis (PCA) belong to this category. In *human-assisted*

methods [80], dimension reduction algorithms are guided by user feedback. These approaches often use a parametric dimension reduction algorithm and then rely on human input to fine-tune these parameters. For example, the user may change the orientation of the projection plane or may choose the most relevant diagrams among several proposed projections of the data. Considering our inability to perceive high-dimensional data, human-oriented methods rely on highly interactive mediums to compensate for this shortcoming.

In automated dimension reduction methods, a predefined algorithm transforms the original complex data into a low-dimensional representation of it. The user usually does not have any influence on the internal parameters and features of the projection algorithm. However, she can often interact with the final visualization. Automated complexity reduction methods are diverse and extensively used. A detailed source on the main algorithms used in this category is provided by Dzemyda et al. [25]. Innovations in this category are either new algorithms such as a new dimension reduction method presented by Carlsson [16] which is based on algebraic topology, or improving the efficiency of current algorithm such as a new method presented by Ingram et al. [39] for speeding up MDS using Graphical Processing Units (GPU).

Theus [80] provides a detailed introduction to common visualization-oriented tools and techniques used in visualizing high-dimensional data, such as mosaic plots, trellis displays and parallel coordinate plots. These tools are often used for visualizing datasets with moderate complexity where it is possible to plot all the two-by-two combinations of the data dimensions. However, visualization options for complex datasets with hundreds or thousands of dimensions are very limited [84]. Some approaches rely on user interaction to overcome this limitation. For example, a common practice is to draw a scatterplot for each pair of dimensions and then via the brushing technique, the user selects a small subset of points in one scatterplot and sees the same points highlighted in the other plots. Another example is presented by Elmqvist et al. [27] where starting with a similar matrix of scatterplots, the user can navigate from one plot to an adjacent one while the transition happens in a 3-dimensional space, providing a better spatial representation of those dimensions. Reshef et al. [63] propose another algorithm which still focuses on pairwise correlations between data dimensions but combines it with sophisticated nonlinear pattern recognition methods,

therefore increasing user's ability to understand the data.

Human-assisted dimension reduction usually starts with a projection algorithm that has some parametric values. The role of the human is to fiddle with these parameters until the final projection is more suited to her needs. This approach adds an extra layer of sophistication to the visualization system and extends its capabilities in generating meaningful projections of a complex data. It also provides the added value of engaging the operator in the visualization process. This can both increase the awareness of the analyst and also save valuable computational resources as the user can indicate which subset of the data should be projected first. One of the early examples of human-assisted methods in visualizing high-dimensional datasets is Grand Tour [80]. In a Grand Tour, the analyst can choose any arbitrary nonorthogonal projection of the data. This can reveal features that may remain hidden in the conventional orthogonal projections used in some other approaches such as parallel coordinate plots. Another early example of human-assisted methods is proposed by Olsen et al. [58]. Their system has been used to visualize documents in a multidimensional setting. Each dimension is represented as a point in the visualization plane and documents would attract or repel these points based on their similarity to each dimension. Also, by moving the feature points, the user can see how each document reacts which is helpful in clustering documents into similar groups in their complex environment.

To sum up, we can emphasize the role of interaction in human-assisted methods. Exploring complex datasets with many dimensions is an inherently challenging task. Therefore, the basic method used in all complex data visualizations is to reduce the complexity to a manageable number of dimensions. Allowing the analyst to manipulate the projection algorithm and its parameters is an essential step in helping the user to make sense of the intricate relations between different dimensions of the data.

Steering is one of the recently developed techniques in human-assisted approaches. Williams and Munzner [88] introduce a navigation mechanism in which the operator steers the system toward the desired subspace of original dataset. The projection algorithm is then focused on this area, avoiding unnecessary computations on the rest of the dataset. Also, by actively engaging the user in the process of complexity reduction, the operator builds a better mental model of the data. Ingram et al. [38] provide a different mechanism for engaging the user.

Their system provides a collection of different dimension reduction algorithms and provides tools for tuning their parameters. The analyst can combine these algorithms until she finds a desirable low-dimensional representation of the data. This approach is especially helpful when the user is not an expert in machine learning or dimension reduction techniques. The authors also extensively use the idea of navigation and landmarks. Different levels of global and local navigation improve the exploration ability of the visualization tool while landmarks help the user find interesting projections of the data. In a similar approach, Gratzl et al. [33] introduce a tool for exploring rank-based data where the projection algorithm is a simple weighted linear combination of dimensions, however, the user has more flexibility in selecting each weight and the overall combination rules. The tool is also highly interactive, making it easy to create new hypotheses and then testing them through a simple drag and drop process.

2.5 EXPLORING BIG DATA

Visualization research has been successful in transforming raw data into meaningful visual representations, however until recently, the size and complexity of the data have not been a major influencing factor in designing visual analytics systems. The emergence of big data has resulted in unforeseen challenges which in turn have caused a paradigm shift in the tools and technologies proposed by the visualization community. For example, a common problem in dealing with big data is the rapid loss of interactivity as data grows. In addition to the scalability issue, visualizing and understanding complex datasets with hundreds of dimensions is very challenging. These issues have opened new lines of research which often try to change the underlying visualization assumptions to overcome such limitations better.

Bethel et al. [9] define big data as one that cannot fit in its entirety in the available Random Access Memory (RAM). As all visualization software work with data that is already loaded in the memory, such large datasets impose challenges to the speed in which a visual analytics software can collect, analyze, and visualize data. Hence in the visual analytics domain, we can further limit the definition of big data to any large dataset that cannot be

processed in less than ten seconds (to avoid breaking user's thought process) or cannot be visualized in an interactive manner (i.e., a minimum of two frames per second). Most popular visual analytics tool cannot satisfy these requirements as an even medium-sized datasets with $20,000$ elements can cause their frame rates to drop below one frame per second [14].

Considering the exponential growth of data and the linear growth of some key hardware components such as GPU memory, Beyer et al. [11] call for rethinking the traditional visualization pipeline and pursuing new sets of algorithms and data structures designed to scale appropriately with the size of data. As visual analytics tools rely on database and data processing technologies, there are numerous efforts to speed up such systems and achieve fast query response times on large datasets, an example of which is Dremel data query engine, designed by Google [52]. However, Fekete [28] argues that even these approaches are not sufficient and will not be able to keep up with the speed of data growth. For example, he mentions that database systems assume that each query results in a small subset of the data, while in practice, visualization tools often require all the data. Therefore, Fekete also calls for developing new methods in approaching big data which are specifically designed with the limitations and requirements of visualization systems in mind.

Shneiderman proposes an early attempt in classifying big data visualization appraoches [73]. He argues there are three main categories of large-scale visualization. In *atomic visualizations*, each data element is preserved and represented by a pixel. Such visualizations often result in very dense visualizations where many points overlap each other. On the other hand, in *aggregate visualization* techniques, each pixel can represent several data elements, effectively reducing the complexity of the visual space. The third approach of *density plots* is similar to aggregate techniques. However, it considers spatial relationships between the underlying data elements to cluster them better and provide more visualization-aware aggregations. Shneiderman argues that aggregate methods reach a reasonable tradeoff between keeping the display complexity low and facilitating knowledge discovery.

The research on supporting big data in visual analytics can be classified into three main categories. The first approach revolves around the idea of optimizing the conventional visualization pipeline. These solutions utilize more powerful hardware components or smart preprocessing techniques to speed up the data processing pipeline. Consequently, their per-

formance is ultimately dependent on the size of the data. Section 2.5.1 discusses these *input-sensitive* methods which can be slow, but produce accurate results. In contrast, the second approach is based on the idea of being inaccurate but fast. These solutions argue that as the final output is always presented on a screen, providing more accuracy than the number of pixels available to the user is not necessary. Hence, such techniques provide answers that are not completely accurate but are good enough for the purposes of the data exploration, and more importantly, they have rapid query response times. These *output-sensitive* approaches are discussed in Section 2.5.2. The third approach aims to combine the desirable aspects of input-sensitive and output-sensitive methods. These emerging *incremental methods* provide fast results with the promise of eventual accuracy. As such, they start with a low resolution (or low accuracy) answer and then gradually improve it over time. It is then the responsibility of the user to decide when the accuracy of an answer has been good enough for her purposes. These approaches are discussed in Section 2.5.3.

2.5.1 Optimization Methods

Scaling up is a common approach for improving the performance of visualization systems which is based on the idea of utilizing more powerful hardware to process the data. GPUs are specifically designed for rendering millions of visual objects and have seen an exponential growth in their processing power over the last few years. As such, they are a popular target for scale-up solutions. For example, Beyer et al. [11, 12] provide an extensive survey of visualization algorithms that benefit from GPUs and can reach interactive-level responsiveness in dealing with big data. The use of GPUs for pipeline optimization have even expanded to non-visualization domains such as databases where the massively parallel processing capabilities of GPUs have been utilized to improve the query response times of traditional database settings [7].

Another class of solutions is based on *scaling out*, meaning they use common and often cheap hardware, yet parallelize the processing on hundreds or thousands of such units to reach the desired performance level [24]. A famous example of such technologies is Dremel, an interactive data processing engine proposed by Melnik et al. from Google [52]. Dremel

moves the processing to the data where each computing node processes a small chunk of the data and then all the results are aggregated. Scale-out approaches are actively pursued by many enterprises as they do not require new and expensive hardware such as GPUs and instead utilize the common hardware already available in commodity computers. Nevertheless, these approaches are mainly used by large enterprises as the cost of running thousands of computers and the complexity of orchestrating them are prohibitively expensive for most researchers. As a result, some researchers have tried to improve the efficiency of their visualization algorithms to better scale with data. For example, when animating data over time, D3 only redraws visual objects that have changed, hence reusing its processing and improving its performance [14]. Unfortunately, such efficiency seeking solutions are not common and therefore researchers such as Fekete have encouraged the community to design new visual analytics systems which can repair their computations, meaning upon receiving a new query, they can reuse the relevant parts of their previous computation and in result reduce their overall computational needs [28].

Preprocessing is the third class of solutions proposed in optimization methods. The core idea of preprocessing is to predict the most frequent queries and then calculate the corresponding answers (or the building blocks of such answers) beforehand. Lins et al. [48] propose a visual analytics system which utilizes this idea to support large spatiotemporal data exploration. They assume that most queries involve some aggregation on the raw values. Hence, the system implements a data structure to store the corresponding aggregated answers for various queries in different zoom levels. Their system is then capable of visualizing billions of spatiotemporal data points with high levels of interactivity.

Preprocessing techniques require efficiently storing their results for fast retrieval upon user's request. Data Cube, initially proposed by Gray et al. [34], is one of the most influential data structures often used for storing preprocessing results. It was first introduced to generalize SQL's[1] aggregation functions and make them more suitable for data analysis tasks such as creating histograms or cross-tabulations. Data Cube is an N-dimensional generalization of simple aggregation functions and calculating a complete Data Cube requires the power set of all aggregation columns, although, in practice, only a subset of these functions are

[1]Structured Query Language, a language commonly used to query relational database systems.

calculated. Although Data Cubes were first introduced for database management systems, many have incorporated its ideas for visual analytics needs. For example, Stolte et al. [79] use Data Cubes to support multiscale visualizations, allowing the user to zoom along one or more dimensions of the data. Prevalence of new hardware architectures such as GPUs has also resulted in the rise of domain-specific preprocessing methods. For example, Schneider and Rautek [69] discuss a GPU-optimized data structure to support spatial indexing. After preprocessing these indices, they can use the GPU to visualize large-scale point cloud visualizations rapidly. A similar preprocessing method is multi-resolution hierarchies which are instrumental in visualizing datasets across multiple scales, allowing the user to zoom in and out rapidly on a large data [11].

To summarize, the core idea in optimization methods is to utilize data processing resources better. Some use faster hardware, some rely on cheaper hardware but use thousands of them at the same time, and some distribute the processing over time, preparing answers even before the user sends the first query. The algorithmic complexity of these methods is a function of input size, and hence they are input-sensitive approaches.

2.5.2 Approximation Methods

While data sizes are growing without any foreseeable limit, our cognition abilities are fairly limited. Fekete [28] points out that we probably can only perceive a few million features, if not less. As it is our cognitive abilities which is the actual bottleneck in understanding visualizations of large data, a new class of solutions are emerging which focus on the output instead of the input. These screen-aware (or output-sensitive [11]) tools use various data abstractions to reduce the size of the presented information and avoid analyzing portions of the data that are out of the scope of the screen. They then use interactive and exploratory mechanisms to help the user navigate through the visualization and understand the data better.

These data abstraction techniques are based on two assumptions. The first assumption is that people do not care for minute details in a big data visualization. A data analyst who looks at a visualization of millions of points is often only interested in the general shape

of the visualization; the exact location of a single pixel is usually not important to her. On the other hand, she would prefer to interact with this visualization fluidly to form a better mental model of the overall characteristics of the visualization. This leads to the second principal assumption which assumes people need to ask many questions and perform multiple iterations on their hypotheses before they can form the right questions. A data analyst seldom asks a single question. Instead, she forms many assumptions regarding the patterns in the data and refines those assumptions through consecutive visualizations until she can find the answers she is interested in. These two assumptions have led to a new mantra for visual analytics which Fekete nicely coins as "trading accuracy for speed" [28]. The methods based on this idea usually provide an abstract and often lossy representation of the data. However, they do it fast, therefore facilitating hypothesis generation while at the same time preserving the general outline and basic features of the visualization.

One class of solutions presented in this paradigm are called *on-demand processing* [11]. Their main idea is only to draw objects that are going to be visible. For example, if the visual representation of a data point is smaller than a pixel or outside of the scope of the screen, there is no need to process it. One of the most common techniques used in this class is semantic zoom [36]. In contrast to geometric zoom which redraws all pixels upon zooming, the semantic zoom provides more detail when zooming in and hides some of the detail when zooming out. This can reduce processing and communication load, and therefore it has been used extensively in visualization systems, such as online maps, ... An example of semantic zoom is provided by Perina et al. [61] where they represent large textual documents via a word cloud visualization. Then, based on the user's interest, the system provides more information for a particular document.

Semantic zooming is usually used in conjunction with multi-resolution data structures. The basic idea of a multi-resolution data structure is to pre-compute the visible data for each zoom level and then depending on the zoom level explored by the user, it only shows the relevant portion of the visualization. As an example, this technique has been used in GigaPan and TimeMachine [67] to present high-resolution images and videos in an interactive system which allows zooming on any desirable part of the video while keeping the communication and processing overload manageable. Another example of multi-resolution data structures

is presented by Liu et al. [50]. Their system, imMens, aggregates different variables of the data and pre-computes those values for several desirable zoom levels. This can then be used to interactively visualize multi-dimensional datasets with over billions of data points.

Several recent efforts are aiming at providing approximate data representations while abiding by reasonable statistical constraints. For example, Kim et al. [44] have implemented a system for visualizing approximate bar charts of big data. Using a small sample of the raw data, they provide bar charts with a correct ordering, meaning that the pairwise comparison between the size of bars is similar to an accurate bar chart of the same raw data. Nevertheless, as they use a small subset of the data, their processing is very cheap and fast. Another example is provided by Park et al. [60] where again a sampling algorithm is implemented to reduce the size of the raw data. However, the proposed algorithm is screen-aware, meaning that it oversamples from regions of the data that would look sparse on the screen. Consequently, the visualizations generated by this algorithm look very similar to an accurate visualization of the entirety of the data, while using orders of magnitude fewer samples.

Wang et al. [83] take a different approach to sampling. Instead of selecting a subset of the actual raw data elements, they calculate several statistical measures of the data and use those measures to regenerate a sample set depending on the user's query. For example, when their system ingests the raw data for the first time, it calculates the sums of all variables and their pairwise products. This is sufficient to, later on, calculate linear regressions between any sets of variables and hence draw scatterplots without actually traversing over the raw data again. As they mention, their approach is "replacing ideal, impracticably slow plots with rough, practically useful ones". Similar ideas have been proposed by Miranda et al. [53] and Pahins et al. [59]. Both systems are based on approximate data structures that collect simple statistics from the raw data and then make it possible to ask queries without actually querying the original data. Of course, this performance is achieved by forgoing accuracy in the results; a tradeoff that is often acceptable for visualization purposes.

The aforementioned approximate algorithms are all automated; however, it is not obvious whether the system will always choose the correct approximation of the data. This is why another class of solutions involve the user in the loop and ask her to provide feedback on what is important and what should be visualized. The most common type of these techniques is

query-based visualization [11]. Here, the user creates a query or search term and reduces the amount of data to a smaller subset which is then used for the final visualization. For example, Beyer et al. [10] present a query-based system for visualizing neurons in a terabyte scale dataset. The user selects regions and neurons of interest, and then the system presents neighboring neurons and their relationship in an interactive setting. Another technique used for finding the correct abstraction is steering. Here, the user guides the visualization system in a two-way mechanism—the system provides an initial visualization and then the user refines it by steering the system toward her regions of interest and then the process repeats. An early example of this approach is presented by Williams and Munzner [88]. Here, the system uses a dimension reduction algorithm to present a large and high-dimensional dataset but instead of keeping the users as passive observers, it actively engages them. The system gradually shows more points in the projected visual space, and the user selects her desirable regions. This allows the system to only focus on projecting data points in that region, therefore avoiding unnecessary calculations.

Approximate solutions are reshaping the conventional visualization paradigm. They put a priority on speed and responsiveness even if it results in reduced accuracy, presenting a subset of the data, or presenting an abstract and compressed version of it. These solutions are also often screen-aware, which means their computational complexity is usually dependent on the screen size rather than the data size. This makes them excellent candidates for emerging visual analytic tools that are capable of scaling with growing data sizes.

2.5.3 Incremental Methods

Approximate methods have shown significant performance improvements and have made it possible to visualize very large datasets. However, this speedup is accompanied by inaccurate results and often computationally heavy preprocessing requirements. To overcome these limitations, a new class of hybrid methods are emerging which combine the data processing benefits of input-sensitive methods with the visualization oriented philosophy of screen-aware approaches. Moreover, they make few assumptions about the queries a user may ask from the data and rely on simple preprocessing requirements. These new methods are similar to

approximate methods as they initially produce an inaccurate but fast response. However, they do not stop and continue processing the data, gradually improving the accuracy of their result over time. Given enough time, they provide exact results similar to traditional visual analytics system. As a result, these methods are called *incremental methods*.

One of the first examples of this class is proposed by Fisher et al. [30]. In an influential work, they show a visual analytics system that generates bar charts of a large dataset. The bar charts appear rapidly on the screen; however, they are accompanied by confidence interval marks, representing the statistical confidence of the system in the accuracy of its results. As the user waits, the system gradually traverses over more data points and improves the bar chart accuracy and also tightens the confidence intervals. A progress bar shows what percentage of the actual data has been processed at each moment. Surprisingly, most expert users only require 0.1% of the data to feel satisfied with the results. This effectively reduces the processing requirements of the system by a thousandfold, making it highly responsive and scalable. Occasionally, the user seeks an accurate answer for a question. In those cases, she waits until the result is finalized. Such occasions happen after the user has explored the data and has zeroed in on a specific question. In such cases, the user can afford to break from the interactive pace of the data exploration and wait for the answer, without disturbing her knowledge discovery process. In a follow-up paper by the same group, Barnett et al. [8] describe their approach as *progressive processing* and contrast it to traditional approaches such as Dremel. They mention in traditional approaches all the data must be first loaded into the memory, while in the progressive methods, data is processed gradually with only a small portion of it always occupying the memory.

In another study, Stolper et al. [78] make the case for incremental visualizations and argue that large datasets demand such a paradigm shift in visual analytics architectures. They also provide guidelines for progressive visualization systems where they should:

1. show meaningful partial results as the calculations proceed,
2. allow the user to steer the processing towards the subset of the data they are interested in,
3. update the visualizations without excessive changes to the visual space,
4. and make it easy for the users to understand what part of the visualization is new.

A recent system developed by Turkay et al. [81] implements these guidelines in a progressive visual analytics tool. They use visualization cues to help the users rapidly discern the current accuracy level of the visualization. For example, when showing a heatmap, they use large squares to show low accuracy results. As the system improves the accuracy of its answers by processing more data batches, they make the squares smaller, conveying to the user that new data has arrived and also its accuracy has improved. They also suggest that online algorithms are suitable for progressive visualizations. Online algorithms treat the data as a stream and generate their results incrementally. Schulz et al. [70] also study incremental methods and propose a model to decide when should a visual analytics system turn the data into chunks and consume it gradually, and when it should consume all the data and perform monolithic calculations.

Incremental methods are opening a new window into the big data exploration. By assuming that people are the primary consumers of the data and that they require several question and answer cycles before they reach a discovery, they have prioritized speed over accuracy. Nevertheless, these approaches have realized that data scientists eventually need to have precise answers and in those occasions, they are willing to wait. Incremental visualization in the context of big data is an emerging research field with researchers actively asking questions such as what are the architectures suitable for progressive visualization, what are the design guidelines for such systems, and how does gradual visualization affect knowledge discovery.

2.6 HUMAN-DATA INTERACTION

Data is playing a vital role in our daily lives. Until recently, only a few people had to understand and manipulate the data directly and turn it into insights. They use mediums such as simple charts and reports to convey that information to their audience. However, the growth in size and complexity of the data is making it difficult if not impossible to use similar communication mediums anymore. We are now witnessing a surge in popularity of novel and sophisticated data analytics tools designed for general public consumption. Consequently,

we need to learn how to collect data, interact with it, and make sense of it. To address these questions, Mortier et al. [55] are calling for developing *Human-Data Interaction* (HDI) as a new topic of study. As such, HDI investigates how agents (e.g., individuals, or even companies) are engaged in the use of data and how they make sense of it. Additionally, they emphasize HDI researchers should design systems that allow people control their data and protect their privacy. In our study, we have focused on the role of visual analytics in making sense of data, where we are investigating how size and complexity of data affect visual analytics tools and their architectures, and how we can design novel visual analytics techniques with the aim of improving knowledge discovery in big data.

Over the years, several studies have addressed the issue of knowledge discovery in visual analytics tools and provided important guidelines for designing similar systems. One of these core guidelines is interactivity. Fekete [28] argues in favor of having interactivity in all aspects of the visualization pipeline, from acquiring the data, to the analysis phase, to the final visualization. Building an interactive system in all aspects of visualization process is challenging, especially when data size and complexity reaches the bottlenecks of computation and memory. Also, interactivity is usually studied for the final step of the visualization itself, while having an interactive experience in previous steps such as analysis is less developed. Fekete suggests that new visualization systems should provide *non-episodic interaction* with the data. In this type of interaction, the user can continuously fiddle with the parameters of the query while the system instantly generates new visualizations. This means that when the system receives a new query from the user, it does not wait until it completes the previous query; instead, it adjusts its results to the new query mid computation. This interactive query building is essential in forming and improving our hypotheses about the data and as Fisher et al. [30] show in their case study, it can be highly beneficial to data analysts. Another example of non-episodic interaction is provided by Williams and Munzner [88] where a steering method facilitates navigation in a high-dimensional space and an optimized projection algorithm visualizes the large underlying data. Also, the user engages in every aspect of the visualization pipeline and can guide the processing to her point of interest.

Any HDI system working with big data have to deal with latency eventually. Latency can take various forms: it can be a long delay while the system is consuming the data for the

first time and preprocessing it, or it can be a long query-response time, or it can be a stuttery visualization. Hence, it is necessary to study how latency affects knowledge discovery, and how to design HDI systems that better mitigate its drawbacks. Liu and Heer [49] study the effects of having latency in interactive tasks, while the user is working with an exploratory visual analytics tool. In the experiment, they introduce a half a second delay per operation and then collect think-aloud and user interaction logs to assess users' performance. Their findings indicate that even such a small latency has adverse effects on participants' data exploration process, as it leads to a decrease in user activity, lower data coverage, lower rates of observation, generalization, and hypothesis generation. Also, they show that latency affects different operations to a variable degree. For example, operations such as zooming and panning are less sensitive to delay while other operations such as brushing and linking are more sensitive.

The rise of incremental methods (Section 2.5.3) has opened up a new way of dealing with latency, where instead of having the results appear after a delay, they show a continuous stream of ever improving answers to the user. One of the first studies that investigates the effects of progressive visualizations on knowledge discovery has been recently presented by Zgraggen et al. [90]. In their user study, they compare three cases of delay: (1) *blocking*, where the visualization is shown after a fixed amount of delay (either 6 or 12 seconds), (2) *instantaneous*, where the visualizations are shown without any delay, and (3) *progressive*, where they show an approximate visualization very quickly and then gradually improve it over 10 incremental updates (the total process takes either 6 or 12 seconds). They then tested random permutations of these settings on 24 participants and collected their think-aloud and user interaction logs. Participants are then compared based on the number of insights they generated per minute, the originality of their insights (which is the inverse of the number of times that same insight was reported by other participants), their visualization coverage (which is the percentage of visualizations generated by the participants of each group to the total number of visualizations possible in the tool), and their interaction metrics such as the number of brushing operations initiated per minute or the amount of mouse movement per second. As it can be expected, this study shows that blocking methods drastically reduce participants' performance. However, progressive and instantaneous methods show

similar insight discovery and data coverage rates. Although users have to wait for the same amount of time to get the final accurate answer in either progressive mode or blocking mode, in all measures the progressive method outperformed the blocking method. Considering these results, the authors encourage the use of incremental methods in developing HDI tools. Incremental methods require cheaper hardware (as opposed to optimization methods discusses in Section 2.5.1), and make fewer assumptions about the queries (as opposed to approximate methods discussed in Section 2.5.2) and yet users perform similarly as if there was no delay.

The rest of this thesis expands on these findings and investigates how incremental visualizations, in comparison to instantaneous methods, affect knowledge discovery specifically when the data under exploration is large, and the visualizations generated have millions of points, and the space of possible visualizations include hundreds of millions of projections of the data. Moreover, we investigate how successful participants generate insights and how we can learn from their strategies to design effective HDI tools.

3.0 METHOD

To answer the research questions proposed in this thesis, we first need to have a working interactive system capable of performing visual analytics on large and complex data. Over the years, many visual analytics tools have been developed and even commercialized, yet the ability to work with big data is still rare and often non-existent. As the literature review in Chapter 2 demonstrates, most tools either do not work with large datasets or even if they do, they use similar visual representations of the data whether the data is large or small. This means that details are lost and what users see is effectively a small subset of the original data. There are a few tools developed for working with big data, but to do so, they often introduce limiting factors on the types of visualizations they can produce. For example, they may only produce map-based visualizations. The lack of a general purpose and capable tool for working with big data led us to build an initial prototype of such a platform as one of the main (and necessary) contributions of this thesis. Section 3.1, describes this tool and how it can achieve the required high-level interactivity with big data. Section 3.2, shows several examples of using this tool for knowledge discovery with a real large and high-dimensional data. Finally, Section 3.3, describes the details of the user studies performed for this thesis, including how the participants use the tool mentioned above during an open-ended discovery session.

3.1 THE TECHNOLOGY: EXPLORABLE VISUAL ANALYTICS

Explorable Visual Analytics (EVA [6]) is a big data visualization system. It has been developed to address the challenges arising in visual analytics on large and complex datasets[1]. The main philosophy behind designing EVA is to improve hypothesis generation, both in quality and quantity. EVA provides natural mechanisms for navigating large and high dimensional data. It also helps the analyst to look at the multi-dimensional data from multiple perspectives, hence giving her a better chance of finding interesting phenomena in the data. In general, the interactive nature of EVA is critical in sense making and creating a mental model of the data. Also, EVA is designed to be fast with sub-second query response times. We hypothesize that it is important to minimize the time between generating a question and testing it. There is a critical period between when an analyst forms a question in her mind until she can see the relevant visualization to verify that hypothesis. If it takes too long (e.g., even more than 10 seconds), the analyst may lose her train of thought. This is mainly due to our limited working memory. EVA minimizes this delay period and therefore lets the analyst instantaneously test her new ideas. This is, in turn, helpful in generating more questions. EVA is also designed to provide very high-resolution visualizations to make the visualizations closer to the underlying large data. All of these aspects help the user to start with a relatively small set of assumptions, test them, generate new questions, refine them, and gradually build a better model of the data, which then results in finding new and meaningful patterns.

Inspired by the knowledge discovery framework proposed by Schunn and Klahr [71], EVA is composed of three major conceptual sections. In the *data representation* section, EVA provides a 5-dimensional visual space consisting of spatial coordinates (X, Y, Z), Color and Time. Each data point can be assigned to a point in this visual space. In the *hypothesis space*, the user can use a simple one-to-one mapping function from data space to visual space. It is also possible to scale data values to better fit them in the visual space. In the *experimental space*, EVA provides various tools for interacting with and manipulating the visualization to do a visual search and find interesting patterns. These mechanisms

[1]EVA's code repository is available at https://github.com/nebeleh/EVA.

include techniques such as zoom, pan, rotation, choosing a color palette, scaling, camera features, external visual aids such as Google Maps and also some textual helpers such as an information panel.

EVA is a web-based tool developed at Carnegie Mellon University's CREATE Lab [1]. It is a part of the Explorables [2] collaborative which consists of various projects aiming at interactive visual representations of large datasets. EVA is accessible on `http://eva.cmucreatelab.org`. Figure 1 shows a screenshot of EVA in a browser. Through the top panel, *Dimension Mapping*, the user can select a dataset and then assign each dimension of data to one of the five visual dimensions. The *Colors* panel sets up the palette. The *Camera* panel provides two types of camera, perspective and orthographic, plus several viewing direction. Some visual helpers such as Google Map can be accessed from the *Settings* panel. The *Parameters* panel is used for scaling visualization and point sizes. The *Information* panel provides some textual information about the data, such as its current scale or the number of visible points in the visualization. The bottom slider, *Time Slider*, is used when the user assigns a data dimension to the Time visual dimension. Through this slider, the user can navigate between different time frames.

EVA has a client-server architecture. The server has two sections. It has a web server for serving the data to the browser. It also has a data ingestion part for reading files in Comma-Separated Values format (CSV) and turning them into a binary format suitable for consumption by the client-side of EVA. As EVA intends to visualize large datasets, it is designed to load large chunks of data in the memory and directly manipulate them. Usually, datasets are sent over to the client using file formats that are easy to read both by the machine and by the human, such as JavaScript Object Notation (JSON) or text files. Although these files work well for small datasets, they introduce much unnecessary overhead for large datasets (e.g., repetitive metadata or suboptimal encoding formats such as ASCII). Also, when the client loads up these files in the memory, they turn it into Document Object Model (DOM) objects which often introduce another layer of overhead. Hence, loading up large datasets as objects is not feasible, specifically in the browsers. To avoid such problems, EVA works directly with binary formats ready to be copied into the client's RAM. Therefore, an important part of the job of the server is to read the raw data in the CSV format and

then convert it into a binary format. This binary file is then streamed over to the client whenever it requests it. The client then loads this file directly into the memory without any transformations and then works with the data in this low-level binary format. Although working with the data in this format is harder, the efficiency it provides justifies the costs.

The client part of EVA is written in Hypertext Markup Language (HTML) and JavaScript to make it usable by a normal browser. The benefits are that there is no need to install any software to use EVA and sharing the visualizations and stories over the web is easy. To support the detailed and high-resolution visualizations presented in EVA, we had to use technologies recently developed for web browsers which make it possible for them to use low-level resources on the client's computer. A central technology used in EVA is WebGL, using a library called Three.js [4]. WebGL allows direct communication with the Graphical Processing Unit (GPU) on the client and hence allows visualizations with a high level of details and interactivity. By incorporating the low-level memory binary format and use of WebGL, EVA can handle hundreds of millions of numbers and turn them into visualizations with under a second query response times and 60 frames per second refresh rates.

In comparison, current widely used web-based visualization technologies are using DOM objects (not binary formats) and are using the CPU to draw the visualizations (not the GPU). Both of these factors severely limit the number of data points they can handle. For example, D3 is one of the most widely used visualization libraries, but it can only handle up to 10 thousand numbers before the frame rate drops to 1 frame per second or lower [14].

The initial prototype of EVA developed for this thesis had several limitations. It could only load one binary file with 4GB of total data. Loading larger datasets would increase the memory allocation required which crashes 32-bit browsers (which is the most common type). Also, after EVA loaded a dataset, it cannot load a new dataset without deleting the old one, effectively allowing the user to only look at 4GB of the data at a time. These limitations have been resolved in the more recent versions of EVA.

As it is shown in Section 3.2, one of the main value offers of EVA is that it maximizes the amount of information presented to the user. Having millions of visual objects on the screen means that the visualizations are on par with the number of pixels available on the monitor. Such super detailed visualizations are essential for exploring big data and make

it possible to look at large datasets from various scales (zoom levels) and see the intricate details in the data without any compromise. As the human eye is excellent at picking these details up, we can then find discoveries that would have been almost impossible to achieve in any other way.

3.2 KNOWLEDGE DISCOVERY WITH EVA

This section demonstrates EVA in action. We will see how EVA can be used to make knowledge discovery. Specifically, we will see how the high resolution and high interactivity capabilities of EVA are essential in making sense of big data. Of course making any knowledge discovery is highly dependent on the right choice of the underlying dataset. Choosing the right dataset for EVA has been based on several factors. First, we wanted a dataset large enough to be beyond the processing capacity of common visualization tools, yet not too large to complicate the development of our first prototype. As current tools are usually limited to visualizing a few tens of thousands of objects, we set a limit of 5 million points for our dataset. The second factor in choosing a dataset is its complexity. A dataset with 4 dimensions can be visualized completely using spatial dimensions and color. On the other hand, manually selecting and navigating through hundreds or more dimensions is tedious and very complicated. Because of this, we limited the dimension cardinality to 100. It is also important to choose a meaningful dataset acquired from real world measurements. This can lead to relevant and useful knowledge discovery. Also, the analyst can benefit from her expertise in the contextual information accompanying that dataset. Finally, the data should have some meaningful representation in the spatial space; otherwise, a purely visual exploration may not be as beneficial.

Table 1: This table represents all dimensions of LEHD data which are also available in EVA. Some columns (like latitude and longitude) do not exist in the original raw data and are added later to enable geospatial visualizations. Most of the other columns are normalized to show the percentage of a specific job category in a census block rather than the actual number of jobs in that category.

Column	Title	Description
1	Census Block Code	Census blocks are the geographical boundaries similar to zip codes, although they often represent a smaller region such as a city block.
2	Latitude	Latitude of the center of the corresponding census block.
3	Longitude	Longitude of the center of the corresponding census block.
4	Total Jobs	Total number of jobs in the corresponding census block.
5	Age ≤ 29	Percentage of jobs for workers with age less than 29 years old.
6	Age $30 - 54$	Percentage of jobs for workers with age between 30 and 54 years old.
7	Age ≥ 55	Percentage of jobs for workers with age more than 55 years old.
8	Earn $< \$1K/m$	Percentage of jobs with earnings \$1250/month or less.

Continuation of Table 1

Column	Title	Description
9	Earn $\$1K/m$–$\$3K/m$	Percentage of jobs with earnings \$1251/month to \$3333/month.
10	Earn $> \$3K/m$	Percentage of jobs with earnings greater than \$3333/month.
11	Agriculture, Forestry, Fishing and Hunting	Percentage of jobs in this category.
12	Mining, Quarrying, and Oil and Gas Extraction	Percentage of jobs in this category.
13	Utilities	Percentage of jobs in this category.
14	Construction	Percentage of jobs in this category.
15	Manufacturing	Percentage of jobs in this category.
16	Wholesale Trade	Percentage of jobs in this category.
17	Retail Trade	Percentage of jobs in this category.
18	Transportation and Warehousing	Percentage of jobs in this category.
19	Information	Percentage of jobs in this category.
20	Finance and Insurance	Percentage of jobs in this category.
21	Real Estate and Rental and Leasing	Percentage of jobs in this category.

Continuation of Table 1

Column	Title	Description
22	Professional, Scientific, and Technical Services	Percentage of jobs in this category.
23	Management of Companies and Enterprises	Percentage of jobs in this category.
24	Administrative and Support and Waste Management and Remediation Services	Percentage of jobs in this category.
25	Educational Services	Percentage of jobs in this category.
26	Health Care and Social Assistance	Percentage of jobs in this category.
27	Arts, Entertainment, and Recreation	Percentage of jobs in this category.
28	Accommodation and Food Services	Percentage of jobs in this category.
29	Other Services [except Public Administration]	Percentage of jobs in this category.
30	Public Administration	Percentage of jobs in this category.
31	Race White	Percentage of jobs for workers with Race: White, Alone.

Continuation of Table 1

Column	Title	Description
32	Race African American	Percentage of jobs for workers with Race: Black or African American Alone.
33	Race American Indian or Alaska Native	Percentage of jobs for workers with Race: American Indian or Alaska Native Alone.
34	Race Asian	Percentage of jobs for workers with Race: Asian Alone.
35	Race Native Hawaiian	Percentage of jobs for workers with Race: Native Hawaiian or Other Pacific Islander Alone.
36	Race Two/More	Percentage of jobs for workers with Race: Two or More Race Groups.
37	Ethnicity not Hispanic	Percentage of jobs for workers with Ethnicity: Not Hispanic or Latino.
38	Ethnicity Hispanic	Percentage of jobs for workers with Ethnicity: Hispanic or Latino.
39	Education < High School	Percentage of jobs for workers with Educational Attainment: Less than high school.
40	Education High School	Percentage of jobs for workers with Educational Attainment: High school or equivalent, no college.

Continuation of Table 1

Column	Title	Description
41	Education Some College	Percentage of jobs for workers with Educational Attainment: Some college or Associate degree.
42	Education Advanced Degree	Percentage of jobs for workers with Educational Attainment: Bachelor's degree or advanced degree.
43	Male	Percentage of jobs for workers with Sex: Male.
44	Female	Percentage of jobs for workers with Sex: Female.
45	Create Date	Date on which data file was created.
46	Year	Year on which the original data is based on.

Based on these characteristics, we chose United States Census Longitudinal Employer-Household Dynamics (LEHD, [3]) dataset. This dataset provides information on employers and employees across the country. This information includes categories such as employees earning, age, ethnicity, education level, ...[2]. The metadata of this dataset is presented in Table 1. LEHD is aggregated over census blocks which are small geographical regions usually equivalent to a city block. Also, the data is produced yearly, therefore providing enough details both on a spatial and temporal level. This dataset is being used by a wide span of scientists and analysts from economists to urban researchers. As such, it can be fused with a rich set of contextual knowledge from various fields, and therefore it can be a

[2]Details of LEHD data structure is available at http://lehd.ces.census.gov/data/lodes/LODES7/LODESTechDoc7.0.pdf

good candidate for making meaningful knowledge discoveries.

Currently, the visualization tools dedicated to LEHD are limited, and they often work on aggregations of the original data, hence not visualizing it with fine details. The LEHD dataset in its entirety is very large (100GB). Therefore we have limited our work to the state of Pennsylvania from years 2002 to 2011[3]. This subsection of LEHD has 2 845 527 data entries and 44 dimensions.

An interesting feature of the LEHD data is that it includes missing values. 14 of the columns (columns on race, ethnicity, education, and gender) have been collected only in the last three years of the data (2009–2011) and the first seven years of the data is missing for those columns. Nevertheless, Census has decided to include those columns for the missing years, but represent them with zero. This can become confusing as it may be misinterpreted for an actual value of zero for those measures. We decided to keep these values in the data to see how the existence of missing values affects the knowledge discovery process of participants. That being said, participants are explicitly told about the existence of missing values, and during their training session, they see an example of missing values and learn what to expect in those cases. As we will see in the next chapters, still many participants had problems with correctly interpreting missing values, and many forgot about it during the experiment.

Another note regarding the LEHD dataset is that the raw data from LEHD does not have any latitude and longitude columns. We added these columns, later on, to make it possible to show the data points on a map. Also, the raw data shows the absolute values of different measures. For example, instead of indicating the percentage of jobs in a certain category, the raw data shows the actual number of jobs for that category in that census block. Based on a few preliminary tests we performed with volunteer participants, we decided to change these values from their absolute format to a percentage format. Here, most columns are divided by the value of the total number of jobs in that row. This effectively shows the percentage of different categories of jobs and makes the final visualizations easier to interpret.

The rest of this section provides several examples on exploring LEHD data using EVA. These examples are based on the author's personal explorations and some initial experiments

[3]The extracted subset used in the experiments is 322MB.

performed with volunteer participants. These examples elucidate how knowledge discovery can happen with EVA, how the detailed visualizations and interactive capabilities of EVA facilitate those discoveries, and how to interpret various visualizations derived from the LEHD dataset. Overall, these examples will provide a better picture of the actual experiment setup described in Section 3.3.

3.2.1 Example: Income Distribution

The first example is a simple visualization of income (Figure 2). Each dot represents one element of the data. The longitude dimension of each data instance is assigned to the visual dimension X (the horizontal orientation of the figure). The latitude dimension is assigned to the visual dimension Y (the vertical orientation of the figure)[4]. The visual dimension of Color represents the ratio between the number of jobs with an income of $3333 or more per month with the total number of jobs. Therefore, a pixel with bright red color shows a relatively wealthy neighborhood while a pixel with yellow color shows a low-income area. There are 2.8 million pixels in the visualization. From this visual representation, it is easy to locate the main population poles of the state, such as Philadelphia on the bottom right corner or Pittsburgh on the left side. It is also possible to distinguish the major geological features of the area such as the distinctive Appalachian Mountains in the middle of the map. The other important observation is the non-uniform distribution of wealth throughout the state. Most of high-end income earning neighborhoods are concentrated in the suburbs of Philadelphia and Pittsburgh while the regions in the middle are usually less populated and often have a lower amount of income. Figure 3 shows a zoomed in version of Figure 2, focusing on Pittsburgh. This picture also includes a Google Map helper in the background. This layer can be helpful in distinguishing the exact location of each census block. Based on this map, the main wealthy neighborhoods are seen in the middle of the picture, where the University of Pittsburgh and Carnegie Mellon University are located.

In Figure 4, we have utilized all the 3 spatial dimensions. Here, besides assigning longitude and latitude to X and Y, we have assigned the total number of jobs in each location

[4]The latitude and longitude measures represent the central location of the corresponding census block.

to dimension Z. By rotating the visualization, the user can look at the high-income levels (as color) and the total number of jobs (as elevation) at the same time. It is again easy to find the major population hubs. Also, it is evident that there is a more complex relationship between income level and the number of jobs. For example, by looking at Philadelphia at the bottom right corner, we can see areas of high income (red) and low income (yellow) with almost the same number of jobs adjacent to each other. Another interesting example is State College, home of Pennsylvania State University, located in the center of the map. This small city has a relatively low number of jobs, but the color of those jobs shows a high-income region, representative of its higher education employment sector. It should be noted that most of the visual objects in a point cloud are obscuring each other; therefore it is essential to have interactive capabilities. Through rotation, zooming, and panning, the user has a much better affordance for understanding the general outline of the visual space.

The last visual dimension available in EVA is Time. By assigning a data dimension to time, we can create an animation and control it through the bottom slider. Figure 5 shows the high-end income range percentages over a course of 10 years. As it is evident from comparing Figure 5(a) to Figure 5(b), the percentage of people with higher incomes is increasing over the decade. This can be due to the inflation in income[5] or a real increase in the overall earnings. The time slider plays an important role in revealing this pattern as the user should go back and forth in time multiple times to better perceive the gradual change in earnings. Again, the interactive nature of visualization is vital in the knowledge discovery step. The same data is represented in a different view in Figure 6. Here, instead of the usual assignment of years to Time dimension, we have assigned it to the Z axis. This results in a series of planes dissecting the data according to their year. This is useful for looking at the general trend. For example, the region in the front of the picture in Figure 6 is Philadelphia. We can see the lower layer (corresponding to the year 2002) has more blue dots (corresponding to low-income neighborhoods). As we go up in the layers, we are going forward in time, and we can see the shrinking of blue regions and the growth of higher-income neighborhoods.

[5]The income values in the raw data provided by Census is not corrected for inflation.

3.2.2 Example: Race

Figure 7 looks at the distribution of races in the city of Philadelphia over the course of three years (from 2009 to 2011). The green regions represent neighborhoods with a majority of Whites while purple regions show neighborhoods with a majority of the workforce from African American community. The first observation is the segregation between these two communities. Neighborhoods are mostly dominated by only one race while in between there are some small border neighborhoods that accommodate a more balanced mixture of both races. The other observation is the relatively fast shifts in the population proportions of some border neighborhoods within a course of a few years. For example, the region marked as **A** in Figure 7(a) shows an area that is mostly composed of African Americans in 2009. However, as we go forward in time to the year 2011 (Figure 7(b)), this area becomes a more mixed race neighborhood. The opposite phenomenon is happening in region **B** where it is changing from a mixed community to a more single-race neighborhood.

The next example shows an accidental discovery. Here, the exploration was not driven by a hypothesis. Instead, it was the exploratory nature of the tool that led to an unexpected visualization. This later resulted in the formation of new hypotheses. When working with geolocated data such as LEHD, it is often common to visualize data on a map. Figure 8 shows a visualization of LEHD data to view it outside of a geospatial representation. Here, each dot corresponds to one census block (i.e., neighborhood) on the map. The number of jobs for males has been assigned to the X dimension, and the number of jobs for females has been assigned to Y dimension. Furthermore, the total number of jobs in each neighborhood has been assigned to the Z dimension. Viewing the final visualization from a perpendicular angle, we come up with Figure 8. Here, a dot on the right-hand side represents a neighborhood with a higher percentage of the workforce being male, while a dot on the left-hand side shows a region with a higher percentage of females in the workforce. The elevation shows the total number of jobs. As can be expected, most of the neighborhoods are located in the middle, with an almost 50–50 percent distribution of jobs between men and women. However, the unexpected feature of this visualization is the one-sided distribution of red dots. Here, we have assigned the number of jobs for African Americans to the Color dimension. Therefore

the red dots show neighborhoods with a majority of the workforce from African American community. Seeing that most of these dots are on the female side of the graph, we can hypothesize that either there is a high unemployment rate among African American men or that they are working in areas with a majority of the workforce coming from the other races and hence their presence is not visible. In either case, the exploratory nature of EVA plus the ability to go through many projections of the data in a short amount of time was crucial in creating this visualization and therefore forming new hypotheses about the nature of the data. It can be imagined that even randomly going through several different projections of the data can reveal some interesting patterns that are not evident in the first place, due to the lack of initial hypotheses in the mind of the analyst.

3.2.3 Example: Age

One of the main goals behind developing EVA was to empower experts in their data analysis. Because of this, we have conducted four in-depth interviews and joint sessions with 14 field experts as they were working with their data using EVA. This step was invaluable in shaping EVA and understanding its capabilities and limitations. The next two examples have resulted from this collaboration.

Figure 9 shows the relationship between age and the total number of jobs. The horizontal axis represents the percentage of jobs for ages 55 or older. The vertical axis is the percentage of jobs for ages 30 to 54. The color represents the total number of jobs: red shows a higher number of jobs in the corresponding census block while green shows a lower number. The year dimension has been assigned to Time to animate the changes over the course of 10 years. Figure 9(a) shows this relationship for year 2002. As it can be seen, almost 60% of jobs belong to the middle age sector while about 15% of jobs are occupied by the older age sector. As we go forward in time—year 2005 in Figure 9(b) and year 2008 in Figure 9(c)—the central point with the highest number of jobs moves from the middle age group towards the older sector. This can be more evident in Figure 9(d) for the year 2011 where middle age sector on average holds 50% of jobs while the older sector has a 20% share. A plausible hypothesis is based on the *baby boomers* event. A consequence of higher birth rates within

the 1950s, there is now a large population of workforce moving from the middle age sector to the older age sector.

3.2.4 Example: Healthcare

The final example explores the relationship between African American workforce and jobs in the healthcare sector. Figure 10(a) shows residence areas with a majority of people working in healthcare related jobs (the red dots) in the city of Pittsburgh. Figure 10(b) shows the racial distribution of the same region, where red dots represent neighborhoods with a majority of African Americans and green dots show neighborhoods mostly consisting of Whites. As it can be seen, there is an interesting correlation between where African Americans live and where healthcare related jobs have a higher percentage of the total number of jobs. This can lead to a hypothesis that the healthcare services employ many African Americans.

3.2.5 Discussion

These examples show the capabilities of EVA in action as it is being used for a real-world problem. We can summarize EVA's contributions in three aspects:

1. High Resolution
2. Explorability
3. Responsiveness

High Resolution points to EVA's ability in showing as many data points as possible on a screen without aggregating them into overall summaries. Many tools use aggregation techniques to improve their ability in working with larger datasets, but this reduces the clarity of the final visualization and hides the fine details of the data. Knowledge discovery can be highly dependent on a number of details a user can see.

In the *Explorability* front, EVA provides common interactive techniques (e.g., zoom, pan, time slider, ...) plus easy navigation between multiple projections of data through its dimension assignment tool. Our initial experiments showed that the ability to view data from multiple perspectives is crucial in understanding the data and finding the *WOW* moments

where the analyst observes some unexpected pattern. These moments usually lead to deeper investigations, new hypothesis generation, and sometimes to new discoveries.

Finally, the *Responsiveness* aspect of EVA fully utilizes its other features. Knowledge discovery is a memory intensive process; the analyst should form a series of assumptions and questions in her mind, and then create a series of visualizations, looking at different characteristics of the data in each step. It is important to remember all of these steps and their possible interpretations. If there is an extended waiting period between consecutive steps, the user can easily forget her previous observations and then the knowledge discovery process is being interrupted. EVA is designed from the ground up to address this issue by fully utilizing local computing resources available to make fast and smooth transitions from one visualization to the other. This is a fundamental feature in data exploration, especially when data size and complexity grows.

It is worth noting that EVA should be used in conjunction with a statistical tool. The main purpose of EVA is to facilitate hypothesis generation. It will also show visual representations of the data so the analyst can perform an initial test for each hypothesis. However, coming up with a final accurate and reliable answer is the job of a statistical tool. Another important note about EVA is the role of experts in shaping it. EVA has benefited from many experts from its inception. The choice of data, its visual characteristics (such as color palettes used, ...) has been formed through many joint sessions with analysts from various backgrounds. Their real-time feedback while working with their data on EVA has also been tremendously helpful in recognizing EVA's capabilities as well as its limitations.

3.3 EXPERIMENT DESIGN

This thesis follows two important questions: (1) How can visual analytics tools facilitate knowledge discovery in big data, and (2), what are the technical requisites of an effective visual analytics tool for such use cases. To answer the first question, an open-ended knowledge discovery session was carried out where participants used EVA to explore the LEHD dataset and find interesting facts in it. To answer the second question, a slightly altered

version of EVA was used which introduced latency in its query response time. This helped to understand how a hypothetical system with a different architecture affects the knowledge discovery performance of the participants. Section 3.3.1 gives an overview on the methodologies used for evaluating visual analytics tools and discusses the aspects that are relevant to the goals of this thesis. Section 3.3.2 discusses the implementation of latency in EVA and how it changes the behavior of EVA from its normal settings. Finally, Section 3.3.3 describes the specifics of the user study experiment carried out.

3.3.1 Methodologies in Experiment Design

Various methods have been proposed for evaluating visual analytics systems. For example, Blascheck et al. [13] implemented a tool for analyzing visual analytics tools. They use interaction logs, think-aloud recordings, and eye tracking logs to assess the performance of a visual analytics tool. After all these information is recorded during the experiment session, the analyst looks at visual representations of the collected data to find patterns in user behavior. These patterns then let the analyst assess the performance of the visual analytics tool in question. A similar effort is presented by Nguyen et al. [56]. They introduce a tool for measuring user interactions in knowledge discovery experiments. In their method, they focus on assessing browser-based online sensemaking tools. They collect user interaction logs and later on present those logs via a visual system to the analyst. These logs and visualizations form the basis for analyzing the effectiveness of the visual analytics tool in facilitating knowledge discovery.

Shneiderman and Plaisant [75] present an in depth study on evaluating information visualization tools. They argue the best method for assessing visual analytics tools is a long term and systematic observation of the users of the tool in their work environment and to see if the tool helps them achieve their goals. They then expand on their ideas and propose multiple aspects for a methodical analysis of a visual analytics tool:

1. Multidimensional: Using observations, interviews, surveys, as well as automated logging to assess user performance and interface efficacy and utility.
2. In-depth: Intense engagement of the researchers with the expert users.

3. Long-term: Longitudinal studies for observing expert users over a long period of use of the tool.
4. Case Studies: Detail reports on how a few experts use the tool in real world scenarios.

Interestingly, they argue that evaluating a tool through defining specific tasks is not a good idea, as the task-based experiments are by definition limiting the participant and hindering the creativity required for knowledge discovery.

The use of think-aloud recordings and user interaction logs is common across similar research who look at the performance of users in various visual analytics tools [62, 90, 35, 49]. Additionally, some studies such as [62] go beyond the typical think-aloud quantitative measures and select a few participants for in-depth and step by step analysis. Such an approach helps to better understand how successful and unsuccessful participants use the tool and what differences in their strategies have contributed to their overall performance.

Based on the approaches proposed in the studies above, the user study of this thesis is designed around an open-ended discovery session. Participants first learn how to use the tool and then freely explore the data, trying to find as many facts as possible without any direct assistance from the experimenter. No specific task is defined (besides the general directive on finding as many facts as possible) to not limit participants' creativity. All their interactions were recorded via a think-aloud protocol where they were asked to talk loudly about their thoughts. These think-aloud recordings were later on used for a qualitative analysis of user performance. Also, their interactions with the tool were recorded by the software with the aim of recreating their every move during the experiment and later on performing a quantitative analysis on those interaction logs in conjunction with information extracted from the think-aloud data. Two surveys were taken as well: one pre-test questionnaire to get some general background about the participants and their familiarity with the relevant concepts, and one post-test questionnaire to evaluate if participants' beliefs about visual analytics tools had changed during the experiment.

3.3.2 Implementation of Latency in EVA

A major objective of this thesis is to evaluate how incremental visualization systems affect knowledge discovery. Considering that exponential growth in dataset sizes may leave visual analytic tool designers no choice but to use incremental techniques or even approximations, it is important to understand how these design choices shape the discoveries we make on big data problems. EVA, by design, provides rapid query response times and creates the intended visualization in full details. For testing an incremental scenario, EVA needed to be altered. This altered version of EVA works the same as the original EVA, except that the visualizations appear gradually on the screen.

The latency algorithm is intended to simulate a throttled network connection. This is a typical case in incremental systems where the data should be streamed over to the client in small chunks. The network bandwidth often limits the size of these chunks and the frequency of receiving them. As a result, EVA's latency algorithm tries to mimic a slow network connection with a bandwidth of 6Mbps. The data points are then revealed to the user as if they are being downloaded gradually and hence appearing in random chunks.

The latency algorithm works by starting from a random element in the actual raw data (all available in the client section of the tool). It then examines the data elements sequentially. Initially, all elements are hidden. Depending on the simulated network speed, a quota is calculated on the number of points that can be revealed in each render frame. This quota determines what percentage of randomly iterated elements can be visualized at that rendering frame. This algorithm in its simplest form would resemble an incremental visualization algorithm. However, current approaches to incremental visualizations suggest that the system should be aware of where the user is looking at and prioritize downloading and showing those portions of the data first. To mimic this behavior, EVA also looks at the current view and from the set of randomly selected elements, first, reveals the points of the currently visible view of the user. The outcome is a visualization that fills the entire visible space with a uniform speed, but upon zooming in on a specific region, it speeds up the visualization process of that region first. In practice, when the user looks at the zoomed out view (e.g., the entire map of Pennsylvania), it takes up to 1 minute for the visualization to complete.

On the other hand, if the user zooms on a specific portion of the view (e.g., the city of Pittsburgh), it will take up to 10 seconds for the visualization of that region to complete. If the user then zooms out again, she will see that the rest of the visualization is still being completed, until the time required for all data being downloaded (as in the simulation) is met.

3.3.3 Experiment Implementation

40 participants have been selected from the University community. Half of them were randomly selected for the latency mode (both during the practice session and also during the actual experiment), and another half worked with the tool in the normal mode (without any latency). The entire experiment takes 1.5 hours. Participants had to fill out a consent form and were free to stop the experiment whenever they decided to leave. Participants received a $20 compensation.

Participants start the experiment by answering a pre-test questionnaire shown in Tables 2. This questionnaire gathers some general information about the background of the participants and their familiarity and exposure to visual analytics tools and concepts. They then receive a 15 minutes training. The training is based on a predefined script to make sure all participants would get the same training. During the training, they are in control of the mouse and are sitting in front of the monitor. The experimenter sits beside them and tells them how to interact with the tool and what is the function of each part of the tool and what is the interpretation of each visualization. The practice session starts by creating a simple map-based visualization. Participants learn how to use the mouse to navigate (zoom, pan, rotate) in the tool. They also learn about the data being used for the experiment. They then gradually add to the complexity of the visualization. First, they add color to see how income is being distributed across the geographical representation of the data. They also learn how to manipulate colors and change their distributions. They then add time to create a temporal visualization and see how income has changed over the years. They then add the total number of jobs to the Z axis to create a 3-dimensional visualization and understand how they can look at multiple variables (like income and the total number of jobs in

this example) at the same time. They also create another 3-dimensional visualization where they use Z axis for time, effectively creating a dissection of the data for each year. This is to show them they can create unconventional visualizations using the tool. Finally, they create a non-map based visualization which demonstrates the relationship between income, high-level education (bachelors degree or higher) and the total number of jobs over time. This helps them to understand how to read correlations between variables. Also, because education column has missing data for several years, it is a good example to show them how they should interpret visualizations with missing data.

After the training phase, the actual experiment starts. From this point forward, any question regarding the correct interpretation of a visualization is not answered. Only questions relating to the functionality of various aspects of the tool are answered (mainly as a reminder for example if they forget what did a button do). Before they start exploring the data, a video recorder is started (with participants' consent). The video recording is from behind and does not capture their face, but only shows their screen and records their audio. Also, an embedded interaction recorder is started which every second records the position of the mouse, camera, and all settings of the tool; enough information to generate the exact view the user is looking at that moment. Users are asked to talk loudly about everything they are thinking about, the steps they are taking, things that they want to do, things that are confusing to them, and more importantly to state whenever they have a discovery and find a new fact in the data.

The main exploration phase of the experiment takes 1 hour. During this phase, participants can use any aspect of the tool and are free to explore the data in any way they want. There is no specific task besides just exploring the data and trying to make correct discoveries as much as possible. After the 1 hour is finished, they are asked to answer a post-test questionnaire shown in Table 3. This questionnaire mainly helps to understand if the experiment has affected their perspective on visual analytics tools and their importance in knowledge discovery.

All the interaction logs, pre and post-test questionnaires, and think-aloud videos are collected and later on used for the analysis. Also, the experimenter took notes as participants were working with the tool to augment the collected data.

The objective of the experiment is to evaluate these research hypotheses:

Hypothesis 1 The introduction of incremental latency in EVA as a Human-Data Interaction system operating on large and high-dimensional data will decrease the quantity and quality of discoveries performed by its users.

Hypothesis 2 There are distinct observable differences between the strategies utilized by high performer and low performer participants.

By evaluating the first hypothesis, we intend to investigate how incremental latency affects knowledge discovery, and by evaluating the second hypothesis, we intend to propose design guidelines that would encourage all users to adopt the strategies used by successful participants.

Table 2: Questions asked from participants before they start the experiment.

Column	Question	Answer
1	Education Level	High School or Lower, Undergraduate Level, Graduate Level
2	Field of Study or Profession	e.g., Computer Science
3	Gender	Male or Female
4	Age	
5	Do you collect personal data such as health tracking information or data from smart home appliances?	No, Sometimes, Regularly

Continuation of Table 2

Column	Question	Answer
6	Have you had any previous education in statistics?	Never, Undergraduate Level, Graduate Level, Advances (e.g., as a profession)
7	Have you ever worked with mapping tools such as Google Earth or Google Maps?	No, Sometimes, Regularly
8	Have you ever played 3D computer games (e.g., driving, first person shooter, ...)?	No, Sometimes, Regularly
9	Have you ever used 3D design tools such as Google SketchUp or AutoCAD?	No, Sometimes, Regularly
10	Have you ever used data analysis tools such as Microsoft Excel, SPSS, SAS?	No, Sometimes, Regularly
11	Have you ever used visual data exploration tools such as Tableau Software, GapMinder?	No, Sometimes, Regularly

Continuation of Table 2

Column	Question	Answer
12	If so, could you please give some examples?	
13	Have you ever seen EVA (Explorable Visual Analytics) tool before?	No, Yes
14	Have you ever used EVA (Explorable Visual Analytics) tool before?	No, Sometimes, Regularly
15	Have you ever worked with demographics data (e.g., U.S. Census)?	No, Sometimes, Regularly
16	Have you ever encountered information visualizations on topics such as income inequality, race segregation, unemployment rate, etc.?	No, Sometimes, Regularly
17	If so, could you please give some examples?	

Continuation of Table 2

Column	Question	Answer
18	If there are some information visualization examples that you like more, could you please describe what the reason was?	
19	How long have you lived in Pennsylvania?	Give an approximate number in the total number of years.
20	Are you interested in learning about workforce and demographics of Pennsylvania?	No, I occasionally read some related news, I am interested and I do follow relevant news, I actively gather and analyze relevant data
21	Do you consider yourself knowledgeable in workforce and demographics information regarding Pennsylvania?	No, I am familiar with it, I am an expert in the field
22	Do you think visualization is beneficial in finding new knowledge?	No, Sometimes, Always or most of the times

Continuation of Table 2

Column	Question	Answer
23	Do you think active participation in exploring a dataset is important in knowledge discovery? For example, would watching a video on income inequality be different than actively plotting a visualization on the same topic? Please explain.	

Table 3: Questions asked from participants after they completed the experiment.

Column	Question	Answer
1	Do you think visualization is beneficial in finding new knowledge?	No, Sometimes, Always or most of the times

Continuation of Table 3

Column	Question	Answer
2	Do you think active participation in exploring a dataset is important in knowledge discovery? For example, would watching a video on income inequality be different than actively plotting a visualization on the same topic? Please explain.	
3	This tool is available online (a URL will be given to you). Would you use it again?	Yes, No
4	Are there any specific questions you want to explore more?	Please give some examples.
5	You can bookmark and share your favorite views in the tool. Are there any specific stories you like to share with your friends or family?	

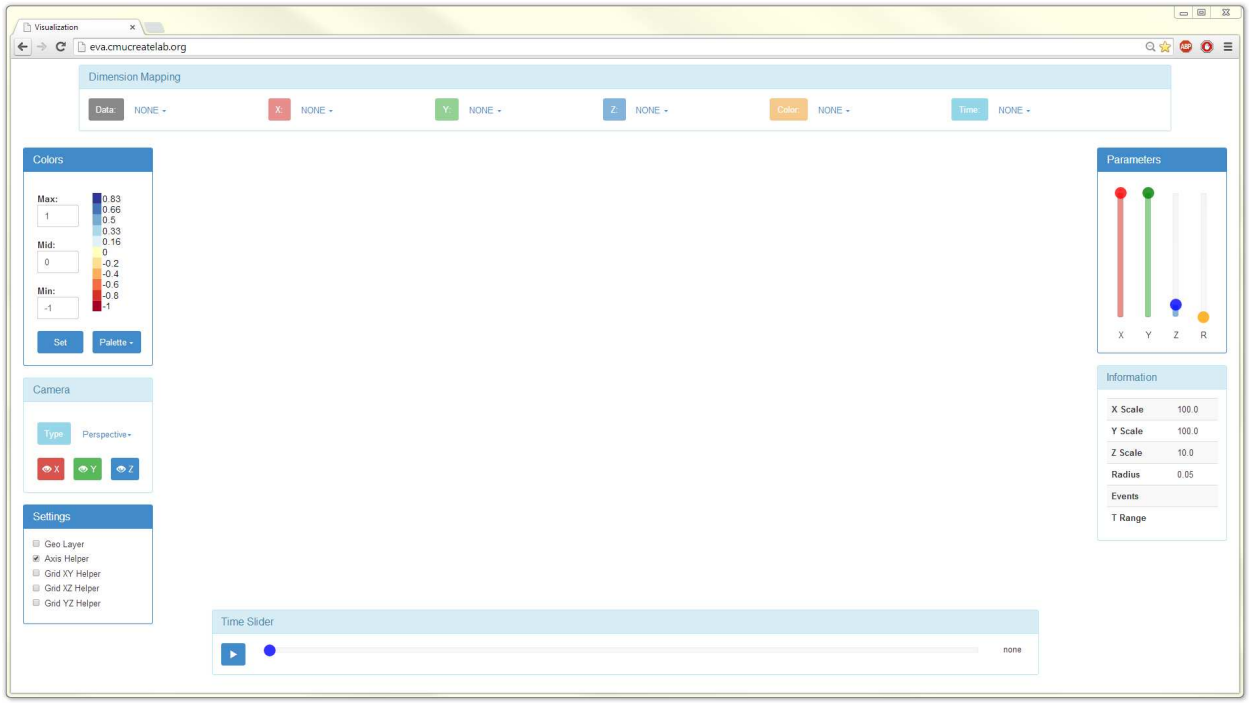


Figure 1: The main user interface of EVA shown in a web browser.

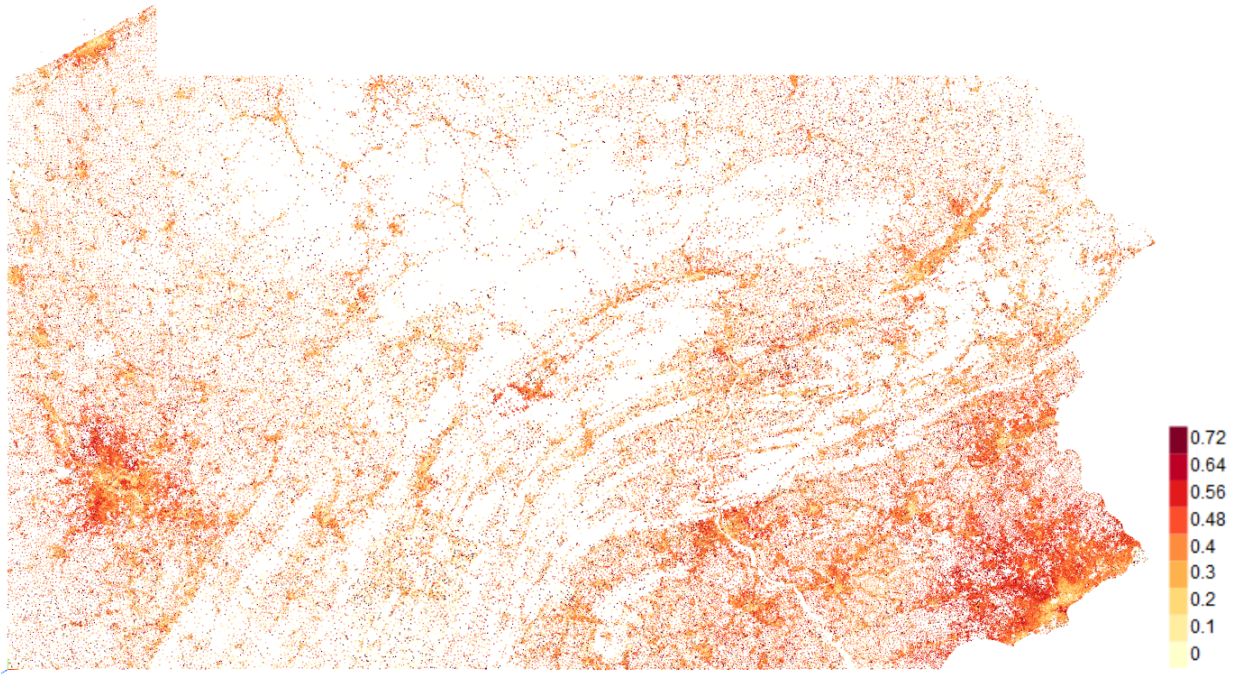


Figure 2: This figure provide a visual representation of earnings more than $3333 per month for the state of Pennsylvania. Each dot represents the center point of the corresponding census block. Red areas show regions with a higher percentage of residents in high-end income range. The color palette on the right shows the minimum percentage of employees with the aforementioned income level in each census block. The two major cities of Pittsburgh and Philadelphia are discernable as dense regions on the left and lower right side of the image respectively. The center of these cities is drawn with an orange color while their suburbs are colored red. It indicates that people who live in the downtown of these two cities have on average a lower income in comparison to those who live in the suburbs.

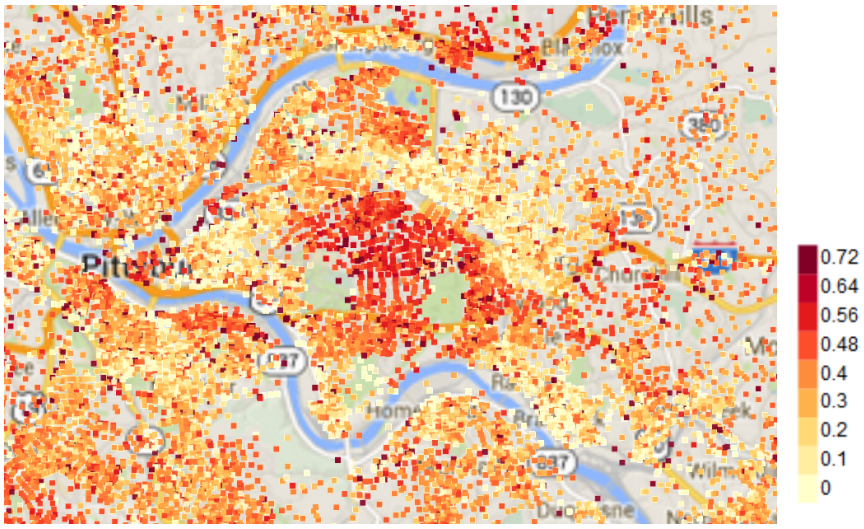


Figure 3: Earnings more than $3333 per month for Pittsburgh.

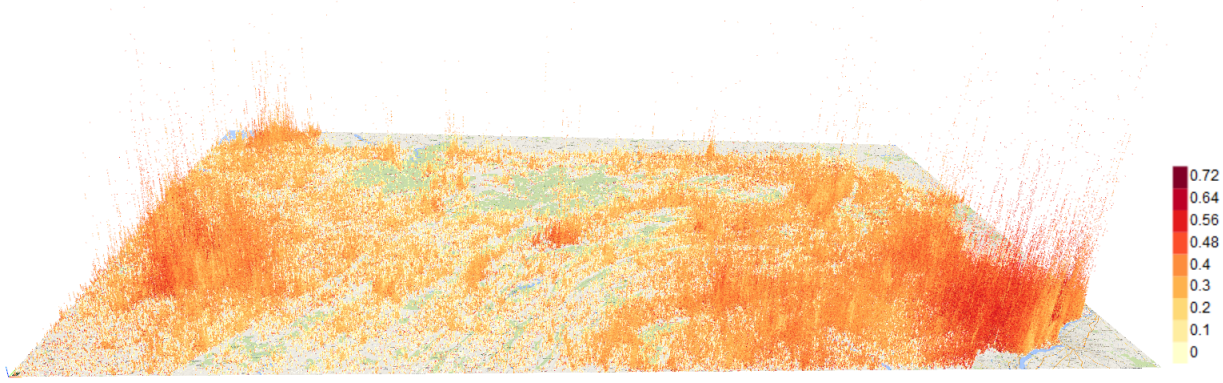
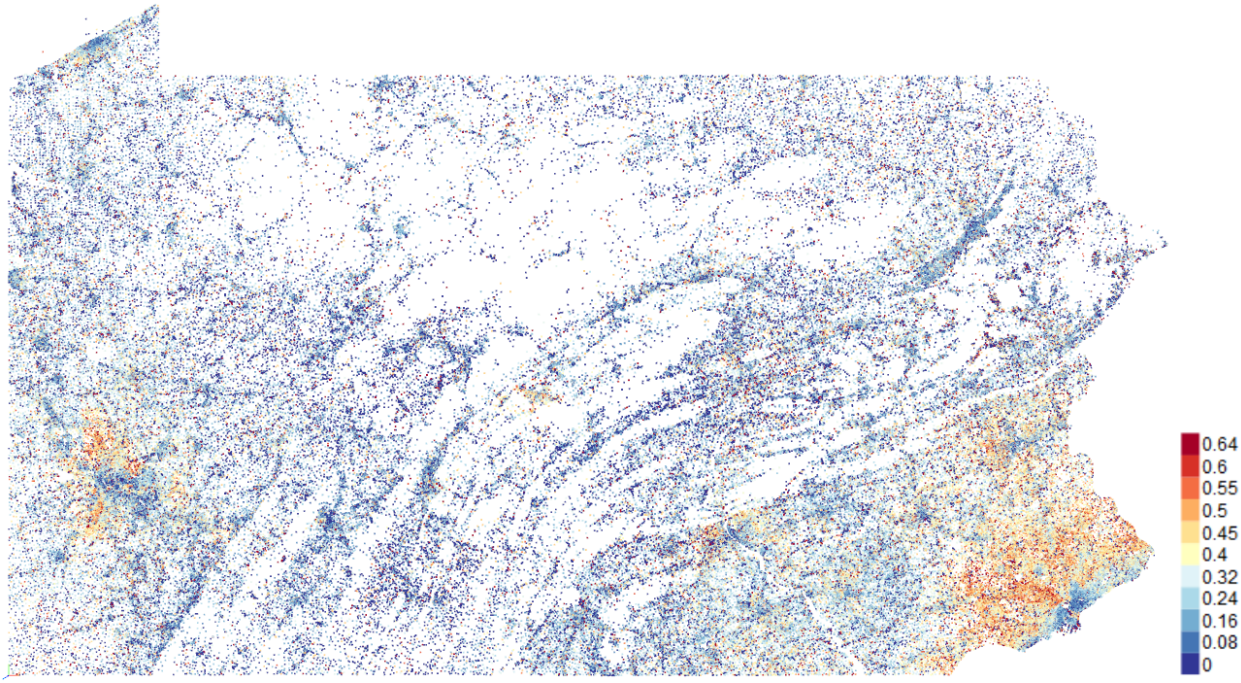
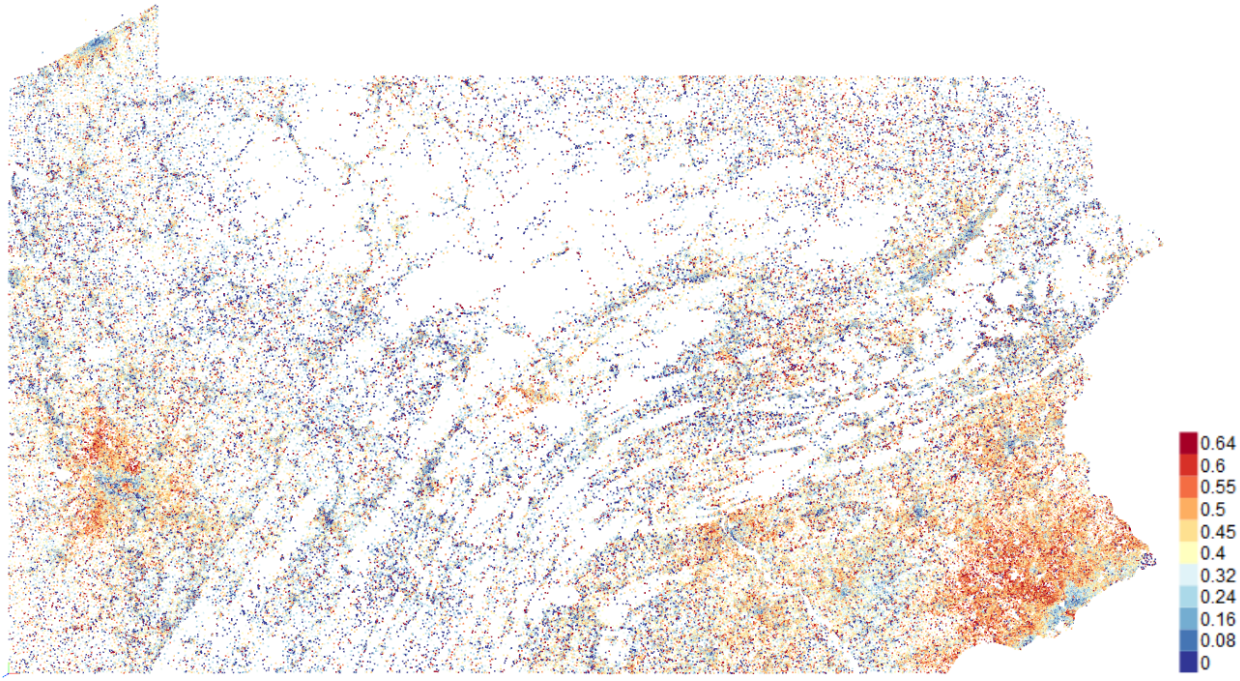


Figure 4: Earnings more than $3333 per month (as color) combined with total number of jobs (as elevation).



(a) Income in Year 2002



(b) Income in Year 2011

Figure 5: Earnings more than $3333 per month in years 2002 and 2011.

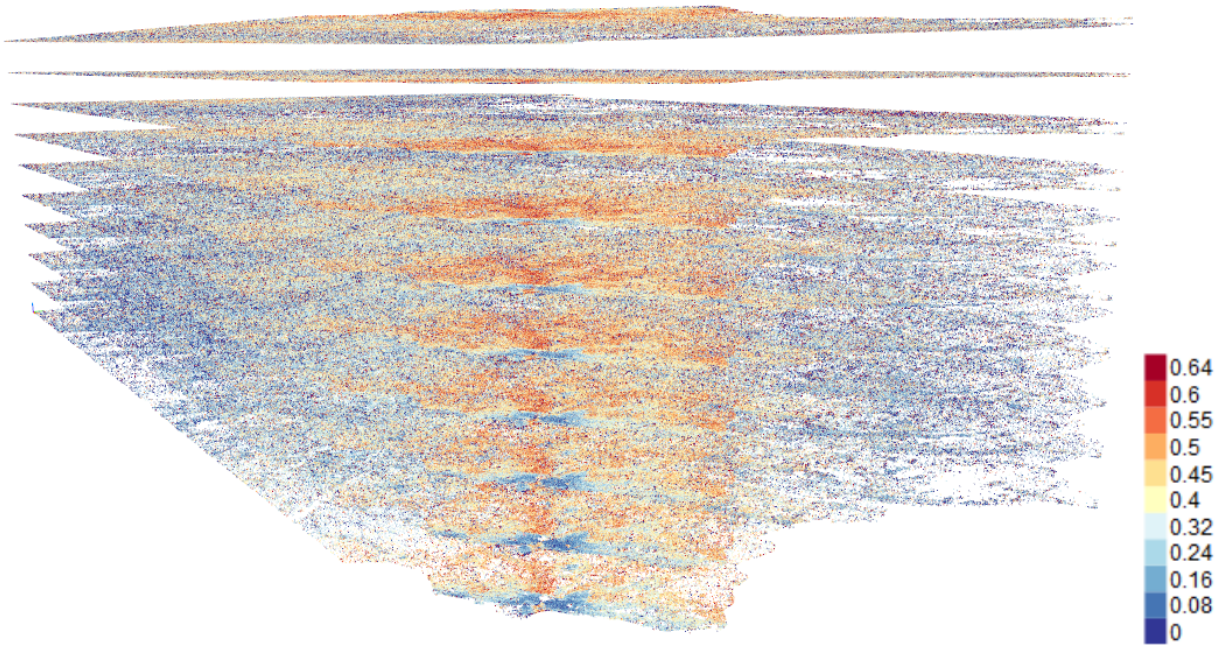
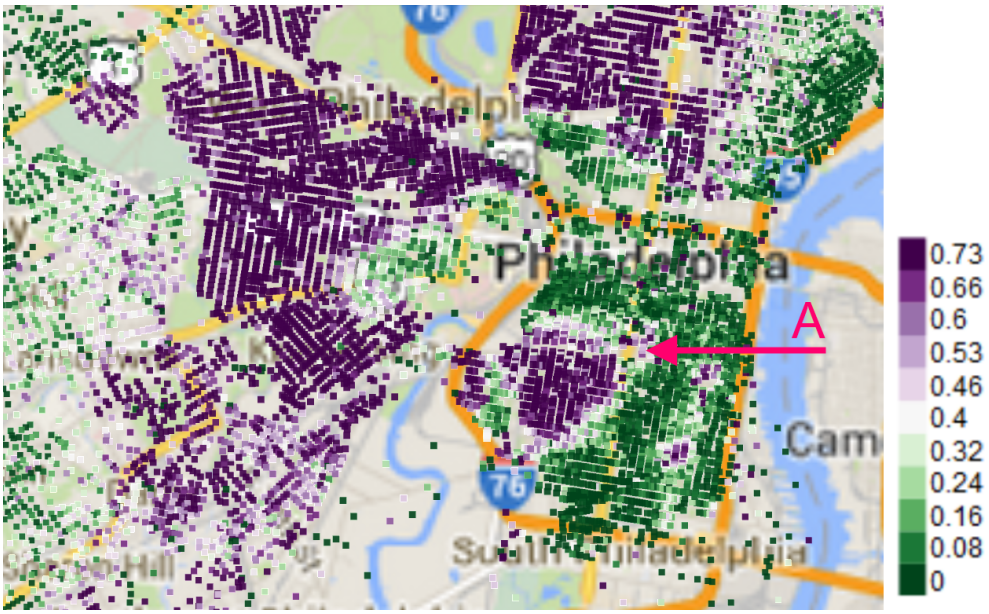
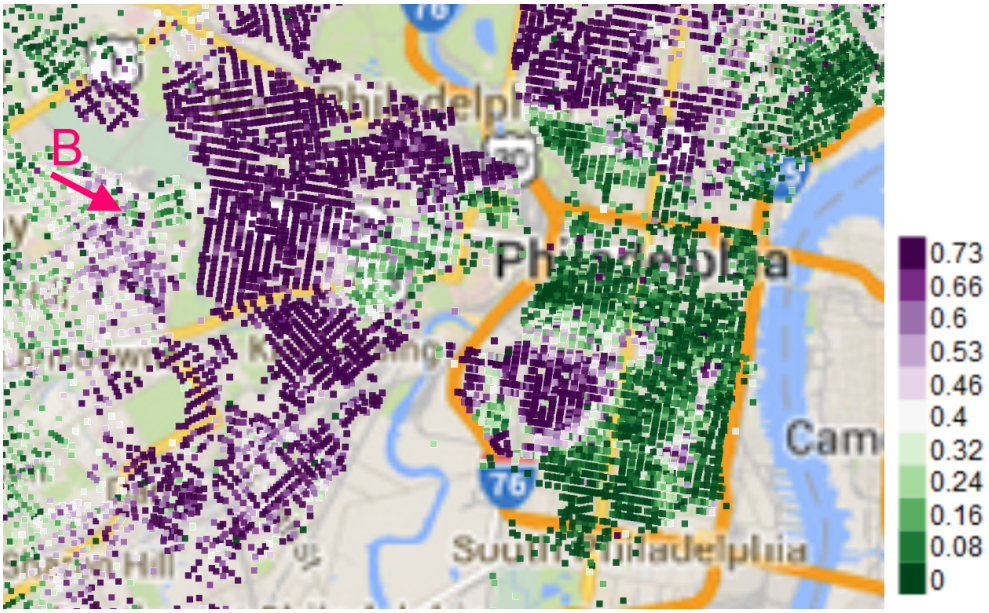


Figure 6: Earnings more than $3333 per month. The year dimension from the data is assigned to the Z dimension in the visual space.



(a) 2009



(b) 2011

Figure 7: This figure demonstrates the distribution of employees based on their race. Purple areas represent neighborhoods with a majority of African American workforce while the green areas represent neighborhoods with a majority of Whites. (a) shows this distribution in year 2009 and (b) is for year 2011.

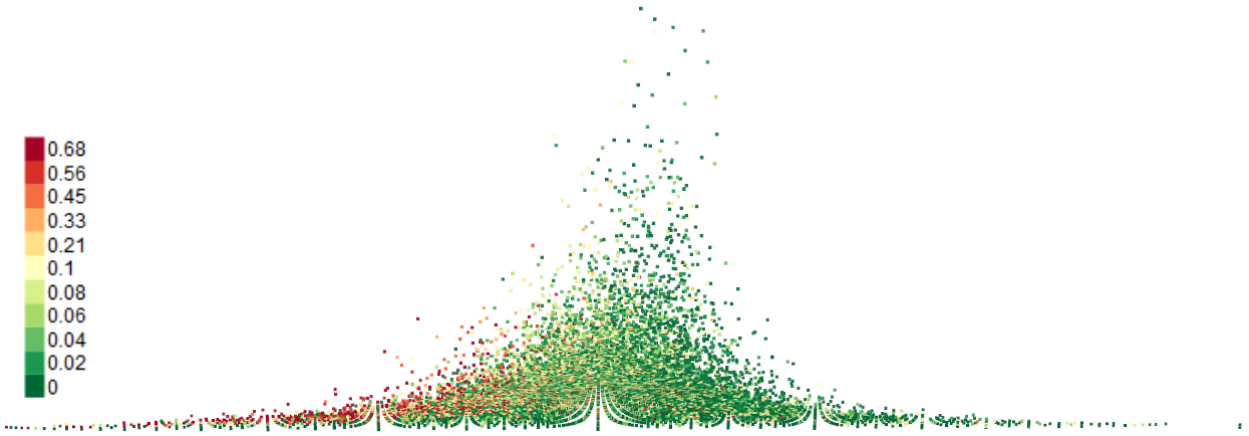
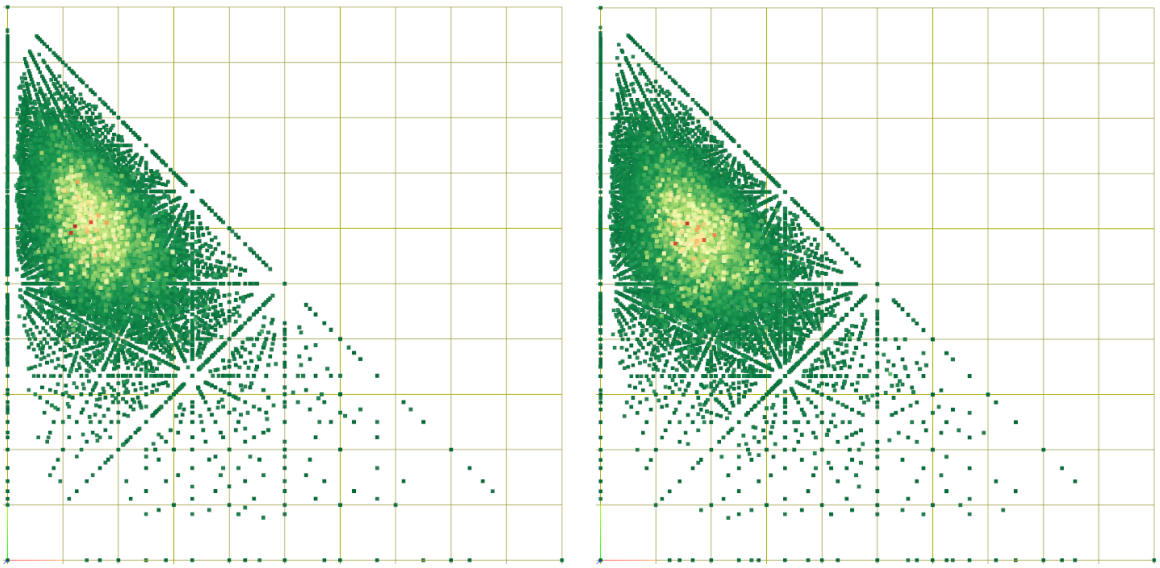
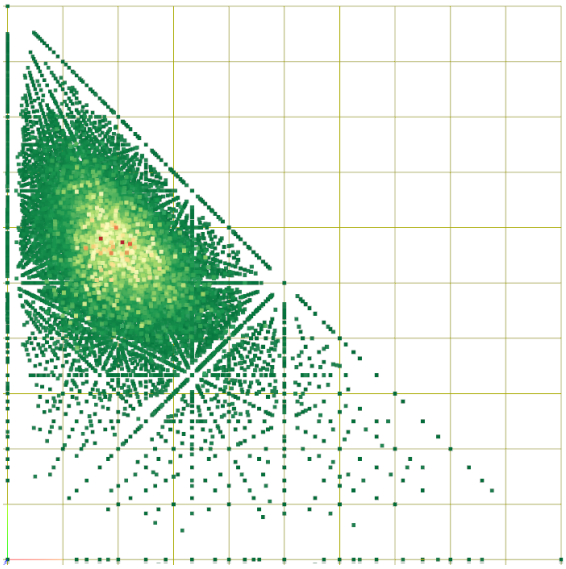


Figure 8: The relationship between race, gender, and the total number of jobs. The dots on the right-hand side represent neighborhoods where a majority of the workforce are men. The dots on the left-hand side are areas where the majority of working people are women. The elevation shows the relative total number of jobs. The color indicates the percentage of African Americans in that neighborhood (red shows higher percentage of African Americans in that census block).

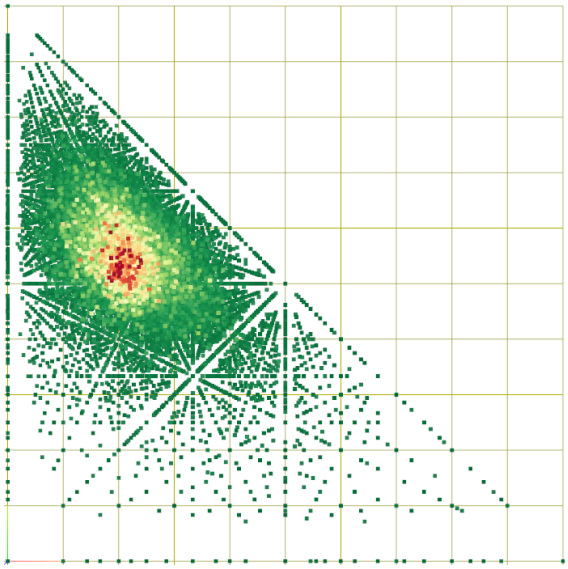


(a) 2002

(b) 2005

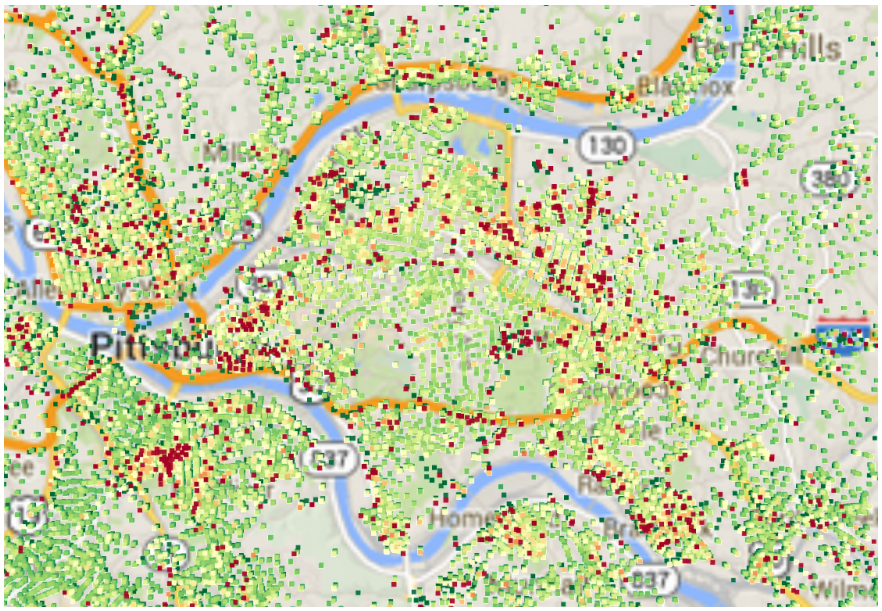


(c) 2008

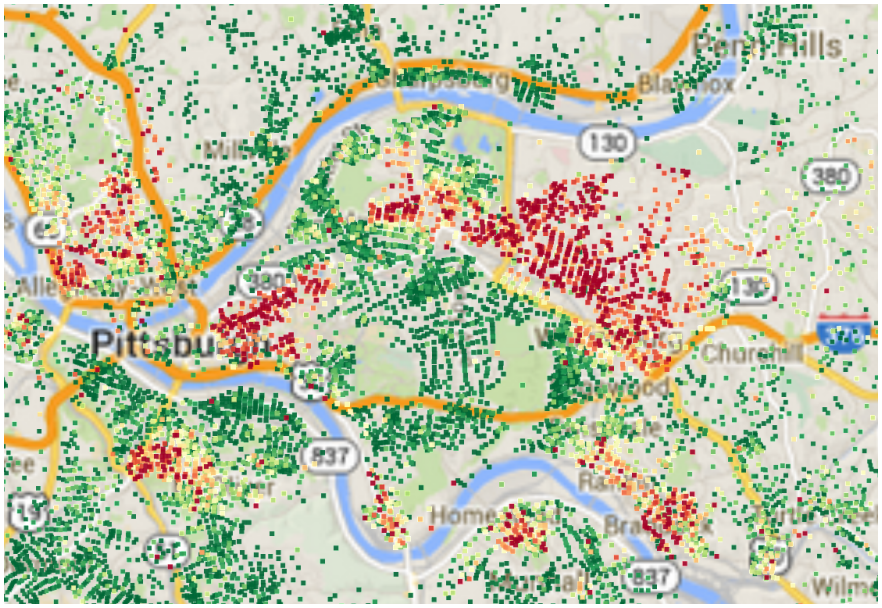


(d) 2011

Figure 9: The relationship between age and total number of jobs. The X axis (horizontal) shows the percentage of jobs for ages 55 or older. The Y axis (vertical) shows the percentage of jobs for ages 30 to 54. Color represents total number of jobs in each neighborhood where red corresponds to a larger number of jobs and green corresponds to a smaller number of jobs. The relationship has been visualized for four different years.



(a) Geographical distribution of people working in healthcare and social assistance



(b) Geographical distribution of African American workforce

Figure 10: The relationship between jobs in healthcare sector and the African American community. In (a), the red dots show residence areas with a majority of workforce in healthcare and social assistance. The red dots in (b) represent neighborhoods with a majority of employees from African American community.

4.0 RESULTS

This chapter is organized as follows. Sections 4.1 discusses how we cleaned the raw data acquired from the user interaction logs and prepared it for analytics. This raw data by itself is not sufficient to answer the central questions of this thesis as we also need to know when discoveries happen and how those discoveries correlate with user interactions. Section 4.2 discusses an indirect approach used to guess discovery events by only looking at user logs. The goal here was to avoid watching all 40 videos and marking discoveries by hand. Nevertheless, this approach did not lead to satisfying results. Consequently, a new direct approach was pursued where all videos were observed and marked. This approach is explained in Section 4.3.

After preparing all raw data and discovery markings, we focus on answering the primary research objectives. Section 4.4 defines the performance measures used to evaluate the success and failure of participants. Then in Section 4.5, we follow the first objective of this thesis and assess how latency affects these performance measures. Finally, in the last two sections, we focus on the strategies adopted by successful and unsuccessful participants. This helps us answer the second objective of this thesis on proposing relevant guidelines for designing Human-Data Interaction tools. Section 4.6 provides quantitative measures on strategies used by all participants. These strategies are then investigated in more detail in Section 4.7 where we look into four selected participants on the extreme sides of the performance measure scale.

4.1 DATA PREPARATION

During the experiment sessions, user interactions were recorded in a raw JSON format. A sample of this data is shown in Figure 11. This information, captured every second, is sufficient to build a complete snapshot of EVA's state. For example, *dimensionXIndex: 2* means the user has assigned the second dimension of the LEHD data (which is the longitude column) to the X axis. The otheR-values show the position of camera, sliders, and helper modules such as the map layer. For each participant, a raw data with 3600 JSON objects is collected, spanning the total one-hour duration of the experiment. These raw JSON files are then imported into Matlab for further analysis.

4.1.1 Views

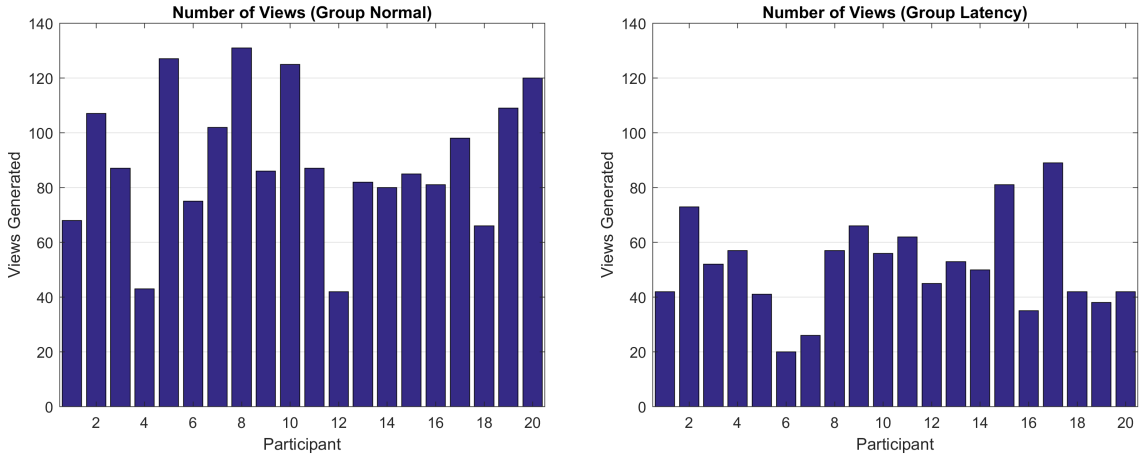
Dimension Assignment is a core functionality of EVA. Participants start their exploration by selecting a visual dimension (say X axis) on the dimension selection toolbar, clicking on the dropdown list in front of it, and then assigning one of the dimensions of the data to that visual dimension. This assignment creates a projection of the original high-dimensional data into the lower-dimensional visual space. Each one of these distinct projections is called a view.

A view provides a subspace where users can visually explore and examine a subset of the data. Based on the observations performed during the experiment sessions, most of the time a view corresponds with a single discovery (although in many cases no discoveries happen in a view and on rare occasions more than one discovery happen in a view). Moreover, when users work with a view, they often have a single question in mind or are testing a single hypothesis. This means there is often a one-to-one correspondence between views and distinct questions and discoveries. As such, views play a central role in understanding user interaction logs and evaluating discovery events. Therefore, as the first step of our analysis, we need to calculate the list of views for each participant and derive some basic statistics on them.

Figure 12 represents the number of views generated by each participant, separated by

```
{
  "type": "snapshot",
  "snapshot": {
    "timeStamp": 1437409224145,
    "datasetIndex": 2,
    "dimensionXIndex": 2,
    "dimensionYIndex": 1,
    "dimensionZIndex": 38,
    "dimensionCIndex": 6,
    "dimensionTIndex": -1,
    "sliderX": 1,
    "sliderY": 1,
    "sliderZ": 1,
    "sliderR": 100,
    "scaleX": 100,
    "scaleY": 100,
    "scaleZ": 100,
    "scaleR": 0.05,
    "paletteIndex": 0,
    "maxColor": 1,
    "midColor": 0.8,
    "minColor": 0.5,
    "axisHelper": true,
    "xyHelper": false,
    "xzHelper": false,
    "yzHelper": false,
    "geoHelper": true,
    "cameraType": "perspective",
    "cameraPosition": {
      "x": 43.75289926816136, "y": 33.9561166795997, "z": 128.63683348914154
    },
    "cameraRotation": {
      "x": 0, "y": 0, "z": 0
    },
    "cameraUp": {
      "x": 0, "y": 1, "z": 0
    },
    "cameraSide": {},
    "control": {
      "x": 43.75289926816136, "y": 33.9561166795997, "z": 0
    },
    "currentFrame": 0,
    "latencyStatus": false
  }
}
```


Figure 11: This Figure represents a sample line of JSON log from the raw data collected during the experiment. These parameters collect the complete state of EVA at each snapshot. One such JSON snapshot is taken for each second of the experiment.



(a) Views Generated in Group Normal

(b) Views Generated in Group Latency

Figure 12: Number of views generated by participants in group Normal (a) and group Latency (b).

group Normal and group Latency. Figure 13 presents a histogram of the same data, showing the distribution of views generated by each group. These samples follow a normal distribution as the Kolmogorov-Smirnov Test (KS-Test) of normality cannot reject the null hypothesis that data comes from a normal distribution (with a P-value of 0.719 for group Normal and a P-value of 0.894 for group Latency). Table 4 represents the average and standard deviation of the number of views generated by each group. An F-Test cannot reject the null hypothesis that samples have similar variance (with a P-value of 0.105). Using a 2-sided unpaired T-Test, we can confirm that the difference between the average number of views generated by each group is statistically significant (with a P-value of 0.000).[1] As it can be expected, participants in group Latency generated fewer views, mainly because they had to wait before each view had sufficient visible points—suitable for any visual pattern recognition.

[1]Throughout the analysis, significance level is assumed to be 5%.

Table 4: Average and standard deviation of number of views generated by each group.

	Average	STD
Group Normal	90.050	25.272
Group Latency	51.350	17.267

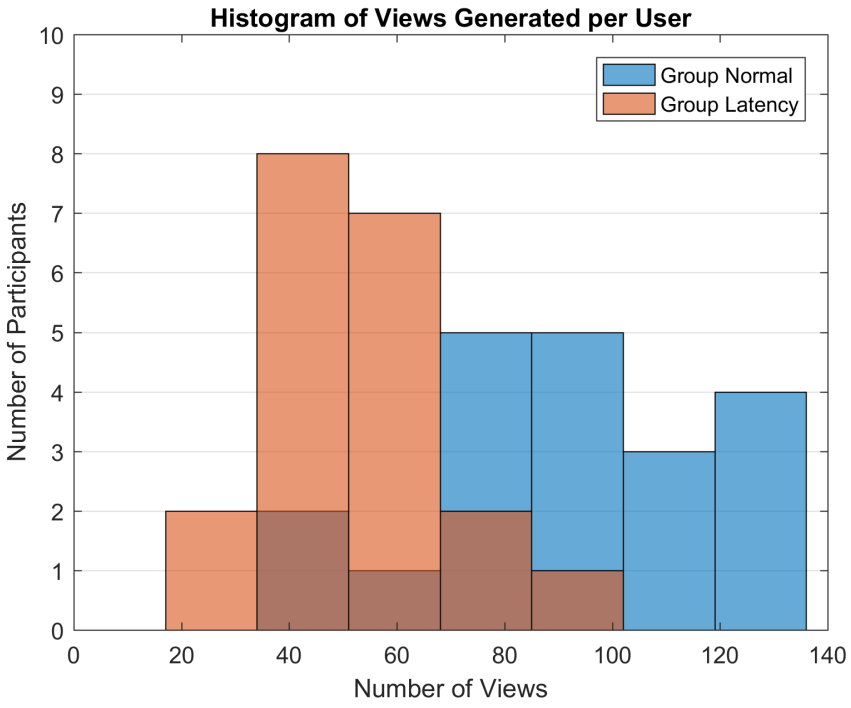
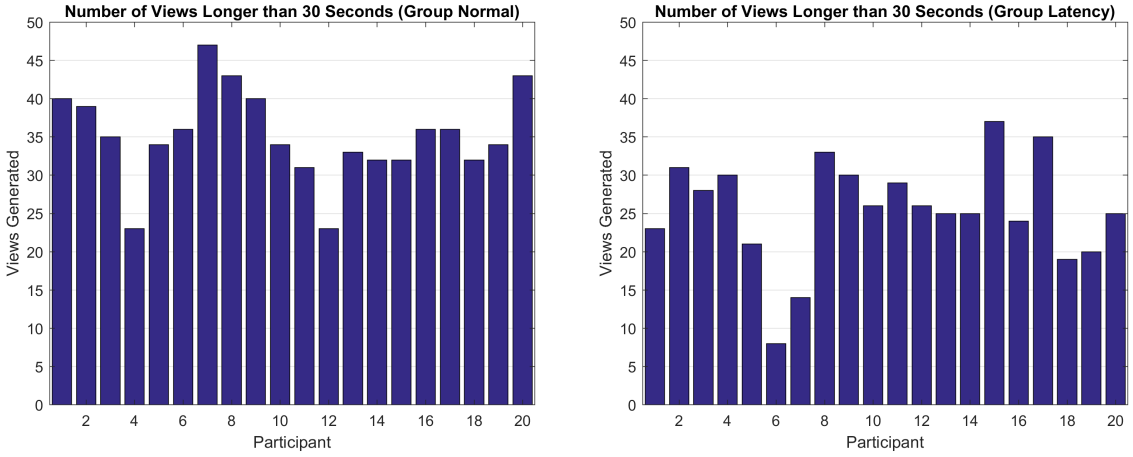


Figure 13: Histogram of views generated by users in both groups.

4.1.2 Acceptable Views

Although views are a core variable in analyzing user behavior, not all views are treated equally by participants. The reason is users can only change one dimension at a time. As a result, when they are changing one view into another intended view, they pass over several transitory views. For example, a user may start with a map-based view where X axis represents Longitude and Y axis represents Latitude. She then decides to look at a different projection of the data and starts assigning Income to X axis and Time to the Y axis. During this transition, a transitory view is generated where X axis represents Income, but Y axis still represents Latitude. This transitory view is visible only for a few seconds until the user finalizes her assignments. To avoid including such views into our analysis, we put a threshold of 30 seconds for a view to be acceptable. This threshold eliminates unintended views and represents the actual navigation performed by the user. Throughout the rest of this chapter, we only consider *acceptable views*, i.e., views with a lifespan of at least 30 seconds, as the basis of our analysis.

Figure 14 represents the number of 30 seconds-long or longer views generated by each participant, separated by group Normal and group Latency. Figure 15 presents a histogram of the same data, showing the distribution of acceptable views generated by each group. These samples follow a normal distribution as the KS-Test cannot reject the null hypothesis that data comes from a normal distribution (with a P-value of 0.712 for the group Normal and a P-value of 0.881 for the group Latency). Table 5 represents the average and standard deviation of the number of acceptable views generated by each group. An F-Test cannot reject the null hypothesis that samples have similar variance (with a P-value of 0.530). Using a 2-sided unpaired T-Test, we can confirm that the difference between the average number of acceptable views generated by each group is statistically significant (with a P-value of 0.000). Although the number of acceptable views is considerably lower than the number of all views created, the relationship between groups remains similar, where group Latency participants generated fewer acceptable views that group Normal.



(a) Acceptable Views Generated in Group Normal (b) Acceptable Views Generated in Group Latency

Figure 14: Number of views generated by participants in group Normal (a) and group Latency (b) where each view is kept for more than 30 seconds.

Table 5: Average and standard deviation of the number of views generated by each group, where each view is kept more than 30 seconds.

	Average	STD
Group Normal	35.150	5.976
Group Latency	25.450	6.917

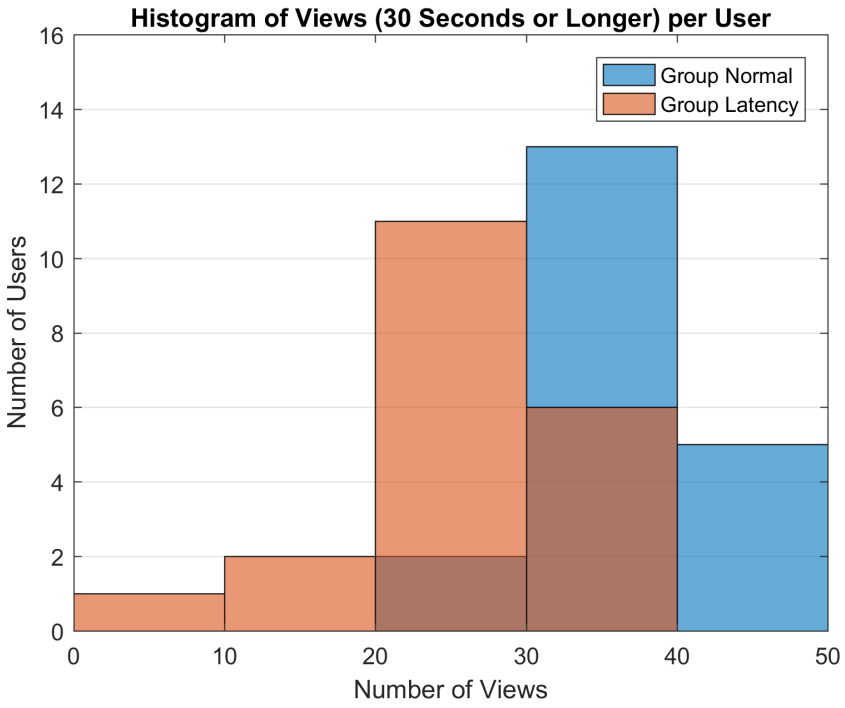
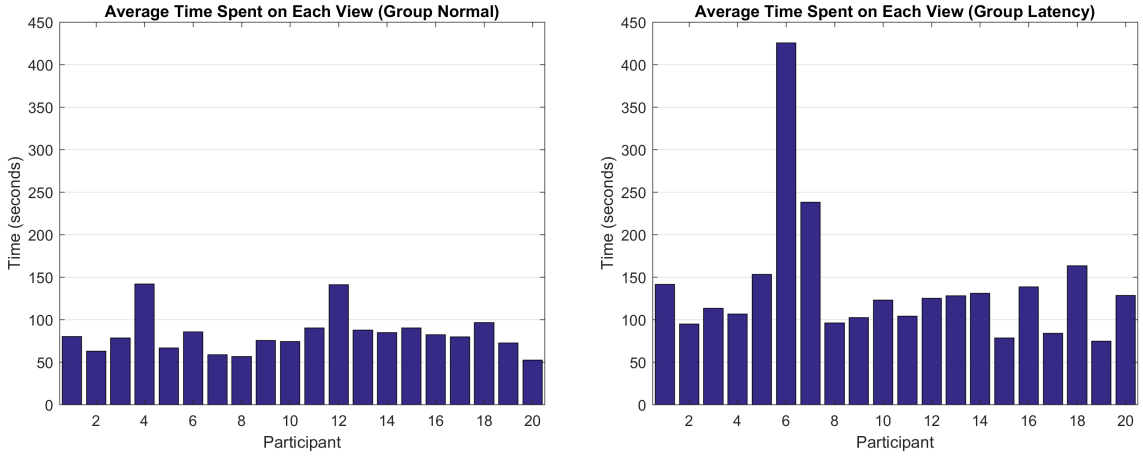


Figure 15: Histogram of views generated by users in both groups. Each view is kept for at least 30 seconds.

4.1.3 Lifespan of Views

Figure 16 represents the average amount of time (in seconds) each participant spent on each view, separated by group Normal and group Latency. Figure 17 presents a histogram of the same data, showing the distribution of average time spent on views by each group. These samples follow a normal distribution as the KS-Test of normality cannot reject the null hypothesis that data comes from a normal distribution (with a P-value of 0.229 for group Normal and a P-value of 0.071 for group Latency). Table 6 represents the average and standard deviation of average time spent on views by each group. Using a 2-sided unpaired T-Test, we can confirm that the difference between the average time spent on views by each group is statistically significant (with a P-value of 0.004).

As expected, participants in group Latency had to wait for each view to complete and consequently generated 0.72 times fewer views than group Normal. As a result, group Latency participants on average spent more time on each view they generated (1.66 times more than group Normal). This means if we compare these groups just based on their raw performance (e.g., the total number of views generated, total number of discoveries, ...), group Normal would always come ahead, and this will not illuminate how latency affects the knowledge discovery process. Instead, we need to normalize our performance measures before we can compare them. Therefore, we have adopted a normalization method whereby all performance measures are calculated on a *per view* basis. For example, later on, when we look at the total number of discoveries as a performance measure, we compare both groups based on the average number of discoveries they generate per each view. Then, we can measure whether the only effect of latency on knowledge discovery is to slow it down, or if it changes the chance of making a discovery even after we normalize for the effects of slow visualizations.



(a) Average Time for Group Normal

(b) Average Time for Group Latency

Figure 16: Average time spent on each view by participants in group Normal (a) and group Latency (b).

Table 6: Average and standard deviation of the average time spent on each view by each group.

	Average	STD
Group Normal	82.989	23.293
Group Latency	137.629	76.912

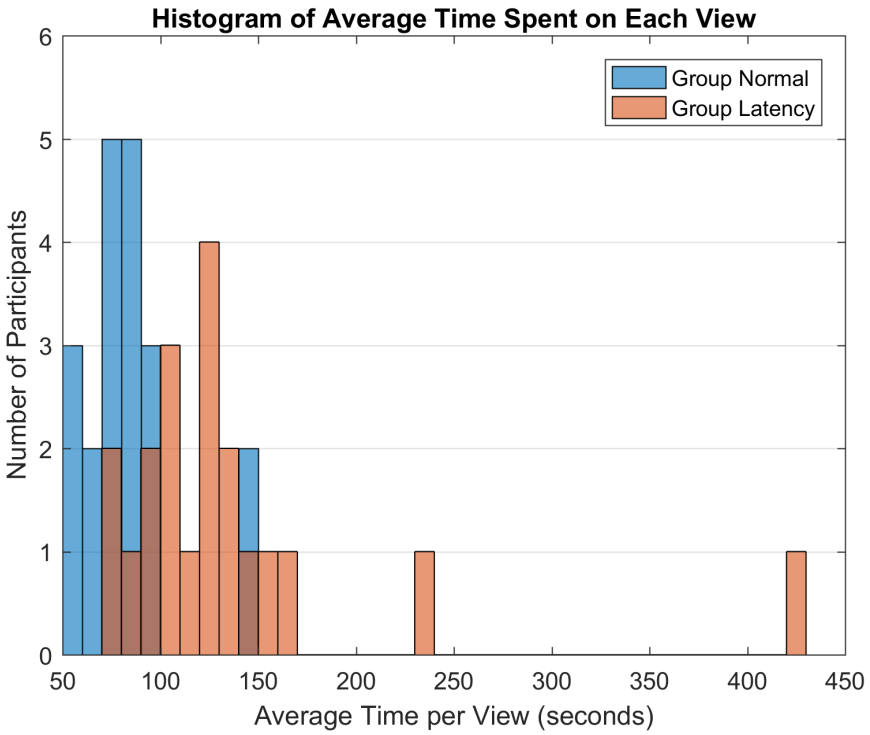


Figure 17: Histogram of average time spent on each view by users in each groups.

4.2 ESTIMATING DISCOVERIES: INDIRECT APPROACH

To answer how latency affects knowledge discovery and how successful participants utilize EVA, we first need to identify when a discovery happens. During the experiment, we do not have any direct method for logging a discovery event; instead, we are using a think-aloud approach where participants disclose a discovery whenever they believe they have enough evidence to know a new fact about the data. To analyze these events and correlate them with the user logs, we need to look at all 40 user interaction videos and mark discoveries. This process is time-consuming and to avoid it, we first adopted an indirect discovery measurement approach.

The assumption behind the indirect approach is that we can derive quantitative measures out of user interaction logs and use those measures as an indicator of actual discoveries happening in an experiment. For example, we may be able to show that users who interact with the views more often are more likely to produce discoveries. We can then use this indirect measure to rank participants based on their discovery scores and then analyze the effects of latency on their performance.

To validate these indirect measures of success, we need to correlate them with actual discovery events. It is possible to use a smaller subset of participants (e.g., four participants) and mark the discoveries in those videos and then investigate if our indirect measures do indeed correlate with the actual discoveries. If so, we can then assume these indirect measures are reasonable approximations of the real discovery events and use them instead for the remaining participants.

Even before we evaluate our indirect measures with a few selected participants, it is reasonable to expect a level of correlation between these indirect measures. For example, if our measures are a valid approximation of the actual discovery events, we expect to see similar rankings for all participants no matter what measure we use. Nevertheless, as we will see in Section 4.2.6, each measure results in a different ranking of participants, with no obvious correlation between them. Even comparing these rankings with the notes taken during the experiment sessions did not reveal any correlation between the measures and the initial assessment of participants' knowledge discovery prowess. Eventually, we decided

to forgo the indirect approach and mark discovery events in all 40 videos, as described in Section 4.3.

In the rest of this section, we introduce measures used in the indirect approach and then analyze their consistency in predicting user performances.

4.2.1 Measure A: Total Number of Views

Views have been defined in Section 4.1.2. For reference, a histogram of views generated by users in each group is shown in Figure 15. Measure A assumes that if a participant generates more views, she is more likely to make discoveries and therefore she should get a higher performance score.

4.2.2 Measure B: Total Interaction Time

The second measure we consider is interaction time. When a user loads a view, she can then interact with it in several ways such as changing the camera position, changing the color distribution, playing with time slider or changing scaling factors. Every second the user changes one of these factors is considered an interaction period. Measure B calculates in total, how many seconds a user interacts with views. We then assume if a user interacts more often with views, she is more likely to make discoveries.

The histogram in Figure 18 represents the distribution of interaction times for the participants in both groups. The distribution of interaction times for group Normal is more skewed towards higher interaction times, indicating that they are more likely to interact with a view than group Latency participants.

4.2.3 Measure C: Average Jump Distance

Another aspect of user behavior is jump distance. This measure calculates throughout an experiment, on average, how many dimensions change from one view to the next. For example, if a user starts with a view where X axis represents Year and Y axis represents Income, and in the next view she keeps these two dimensions the same but adds a third

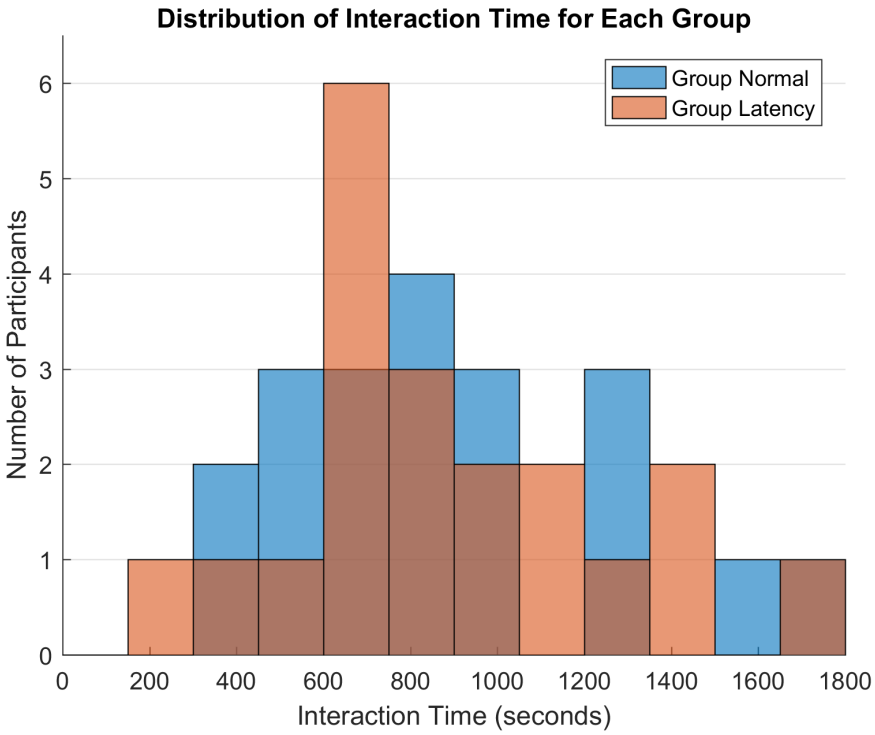


Figure 18: Distribution of interaction time for each group.

dimension of Z axis assigned to Age, then the jump distance is 1. In a hypothetical scenario where a user changes all five dimensions of EVA in every transition (i.e., from one acceptable view to the next acceptable view), the average jump distance measure would be at its maximum value of 5.

The intention behind this score is to figure out which users tend to explore more connected stories and which ones treat each exploration phase as a separate effort to find a fact. During the experiments, some participants learned to make discoveries using a certain pattern (e.g., map-based visualizations) and then used that pattern over and over, each time changing only a small aspect of their view. On the other hand, some participants constantly changed their views into entirely different projections of the data, hoping to find new clues in their hunt for discoveries.

It is not obvious how average jump distance correlates with discoveries. On the one hand, it may be possible that a user who creates more *distant* views is increasing her chance of stumbling upon something new and interesting. On the other hand, working with similar views often means the user is more familiar with the pattern of discovery in that type of view and may be more efficient in interpreting a similar view.

Figure 19 represents the histogram of average jump distance measures for each group. The majority of participants jump between 1.2 to 1.8 dimensions in consecutive views. An initial observation suggests that latency does not affect this behavior among the two groups.

4.2.4 Measure D: Average Dimensions

This measure calculates on average, how many dimensions are assigned by a user to each view. Some participants tended towards more complex views, often creating five-dimensional projections of the data while others mostly created two-dimensional views. This measure can also have non-trivial relationship with discovery events. It is possible that participants who create higher-dimensional views increase their chance of finding discoveries that connect several dimensions of the data. The drawback is that complex views are harder to interpret correctly and may decrease one's chance of making sense of a view.

Figure 20 presents a histogram of the average number of dimensions selected by par-

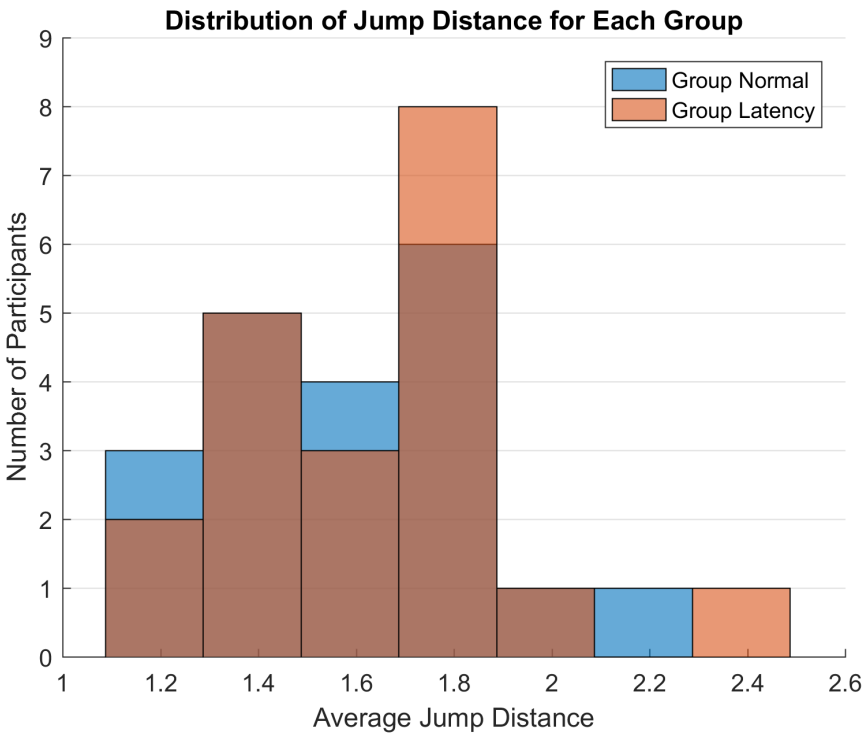


Figure 19: Distribution of average jump distance for each group.

ticipants of each group. An initial observation suggests that participants of group Latency generate more complex views (4 dimensions per view) in comparison to group Normal (3.4 dimensions per view). A plausible reason for this behavior may be because Latency mode participants had to wait for each view more. Hence they might have decided to utilize their time more efficiently and assign more data dimensions to their view. The complex visualizations they created gave them a larger subset of the data to explore.

4.2.5 Measure E: Average Uniqueness

The final measure represents the average uniqueness of views generated by each participant. A unique view is one that no other participant in their representative group has generated. The intuition behind this measure is that participants who tend to build unique views are better at exploring the data and may have a higher chance of making discoveries.

In general, participants in group Normal created 519 distinct views. The top ten most visited views are shown in Table 7. Participants in group Latency created 385 distinct views. The top ten most visited views are shown in Table 8. Map-based views were among the most common views generated by multiple participants in both groups. Surprisingly, the empty view was common as well, indicating that on several occasions, participants had not decided on any projections for at least 30 seconds.

Table 7: The top ten most created views in group Normal. The first column shows the total number of times participants in the group have generated a view. The second column shows how much time (in seconds) has been spent on that view. The other columns show the dimension assignment for that view. Unassigned columns are shown by —.

Total Count	Total Time	Dimension X	Dimension Y	Dimension Z	Color	Time
12	706	Longitude	Latitude	—	—	—

Continuation of Table 7

Total Count	Total Time	Dimension X	Dimension Y	Dimension Z	Color	Time
8	428	—	—	—	—	—
7	462	Longitude	Latitude	—	Percentage of jobs for race African American	—
6	732	Percentage of jobs for education bachelor degree or advanced degree	Percentage of jobs for race white	Percentage of jobs earning more than $3K per month	—	Year
5	301	Percentage of jobs for education bachelor degree or advanced degree	Percentage of jobs earning $3K per month	Percentage of jobs for race African American	—	Year

Continuation of Table 7

Total Count	Total Time	Dimension X	Dimension Y	Dimension Z	Color	Time
5	281	Longitude	Latitude	—	Percentage of jobs for race White	—
4	486	Longitude	Latitude	—	Total Jobs	Year
4	449	Longitude	Latitude	—	Percentage of jobs for education bachelor degree or advanced degree	Year
4	385	Longitude	Latitude	—	Percentage of jobs earning $3K per month	Year
4	294	Longitude	Latitude	—	Percentage of jobs for males	Year

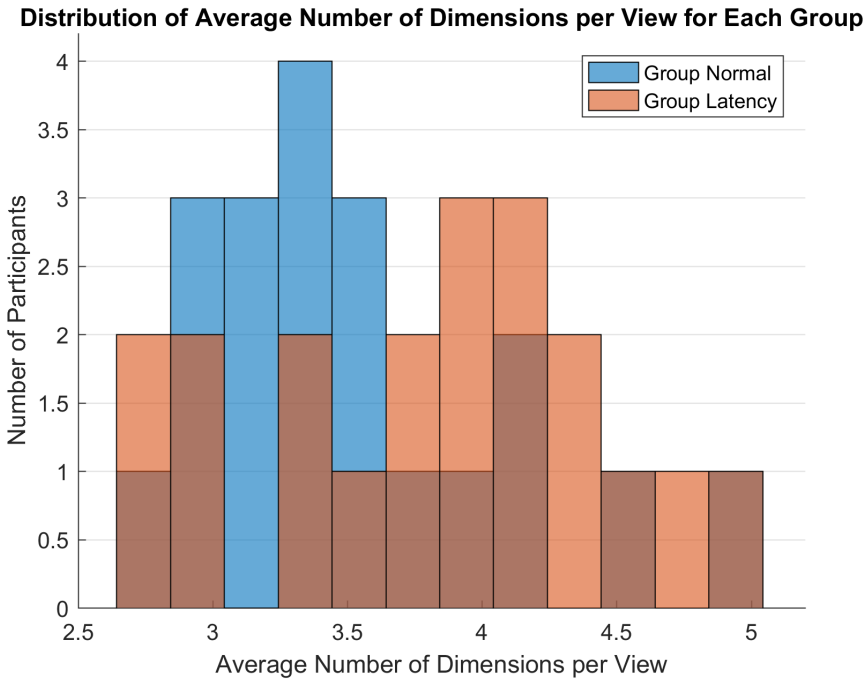


Figure 20: Distribution of average number of dimensions per view for each group.

Table 8: The top ten most created views in group Latency. The first column shows the total number of times participants in the group have generated a view. The second column shows how much time (in seconds) has been spent on that view. The other columns show the dimension assignment for that view. Unassigned columns are shown by —.

Total Count	Total Time	Dimension X	Dimension Y	Dimension Z	Color	Time
18	964	Longitude	Latitude	—	—	—
5	267	Longitude	Latitude	—	Percentage of jobs for race White	—
4	1041	Longitude	Latitude	Percentage of jobs for males	Percentage of jobs for males	—
4	856	Longitude	Latitude	Percentage of jobs for age less than 29	Percentage of jobs for age less than 29	—
4	678	Percentage of jobs for age less than 29	Percentage of jobs earning $3K per month	—	—	—

Continuation of Table 8

Total Count	Total Time	Dimension X	Dimension Y	Dimension Z	Color	Time
4	487	Longitude	Latitude	—	Percentage of jobs earning $3K per month	Year
4	462	Longitude	Latitude	—	Percentage of jobs in professional, scientific, technical	Year
4	461	Longitude	Latitude	—	Percentage of jobs in agriculture, forestry, fishing, hunting	—
4	405	Longitude	Latitude	—	Percentage of jobs for race African American	—

Continuation of Table 8

Total Count	Total Time	Dimension X	Dimension Y	Dimension Z	Color	Time
4	263	—	—	—	—	—

Figure 21 compares views generated by group Normal vs. group Latency. Here, each circle represents one view. Views on the horizontal axis are only seen by participants in group Normal and the ones are the vertical axis are only seen by participants in group Latency. Green circles represent map-based views and blue circles represent non-map views. The values on each axis represent the time spent on that view (in seconds). An initial observation indicates that most of the shared views between the two groups are map-based ones. The most visited views shared by both groups are empty view and empty map view[2]. These views are not directly useful for knowledge discovery but are often a bridge between different exploration phases. Hence we can conclude that such bridges are the most common experience for the participants of both groups, while the actual discoveries are more diverse.

Calculating the average uniqueness measure is as follows. First, we select all acceptable seconds for each participant. An acceptable second is a second of the experiment where the user is looking at an acceptable view (i.e., a view that has been explored for at least 30 seconds). Then we measure how many other participants in that group had generated the same exact view sometime during their exploration. $U(t)$ is then defined as the total number of participants who had generated the view selected at time t. Note that we assume all spatial permutations of a view are similar. For example, assigning Latitude to X and Longitude to Y is considered a similar view as assigning Longitude to X and Latitude to Y. The average uniqueness measure for a participant is then calculated by $(\sum_t 20 - U(t))/T$, where T is the total number of acceptable seconds in an experiment. If all the views generated by a

[2]In the empty map view, X and Y axes are assigned to Longitude and Latitude, but all the other dimensions are unassigned.

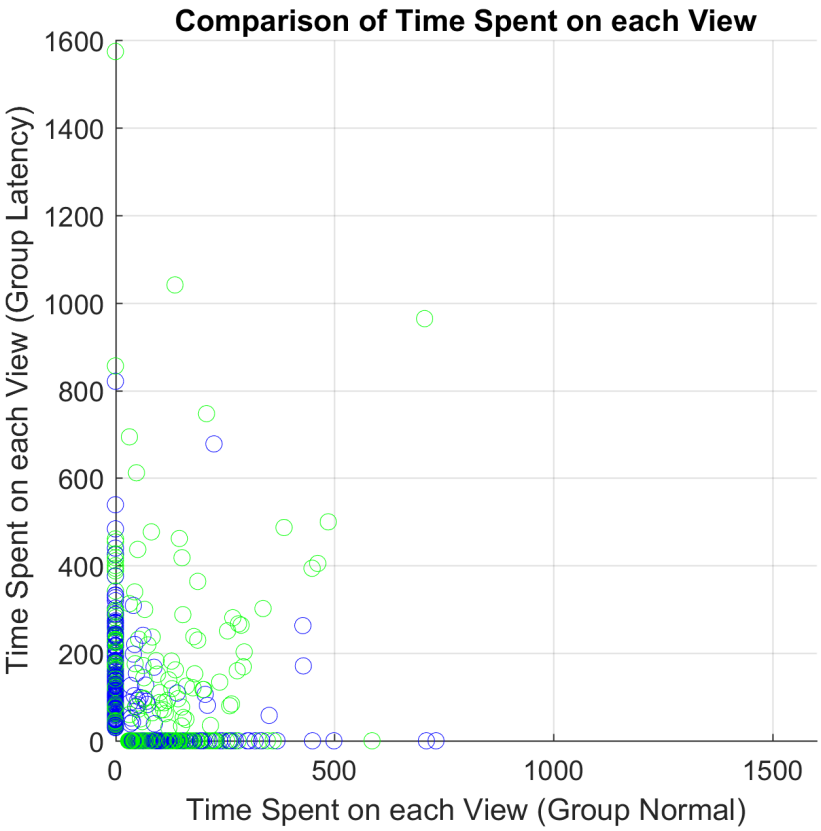


Figure 21: Comparison of time spent on each view between groups Latency and Normal.

participant are unique, her score would be 20. If all the views are generated by others as well, her score would be 0. The histogram of Figure 22 represents the distribution of average uniqueness scores for the two groups. The majority of views are unique; therefore the scores are between 17.5 to 20. Also, an initial observation indicates both groups have a similar distribution of uniqueness measures.

4.2.6 Analyzing Indirect Measures

Now, we can look at all these measures together. If our indirect measures are reliable indicators of the actual discovery events, we can expect to see two patterns:

1. A general difference between the participants of group Normal and group Latency, indicating the effect of latency on discovery events.
2. A pattern within each group, where different measures would rank participants similarly, indicating that all measures point to the same underlying discovery events.

Figure 23 represents the rankings of all participants based on each measure. Each line corresponds to one participant. Lower rankings represent higher scores for that measure. Ideally, we would expect to see orange lines on one end and blue lines on the other end of the spectrum; indicating that our measures can distinguish between group Latency and group Normal participants. Furthermore, we would expect to see each set of lines moving in concert with other lines of the same group, indicating that our measures are correlated and result in similar rankings. Nevertheless, on an initial observation, neither of these patterns are observed. Pair-wise Spearman's Rank correlations between all pairs of measures do not reveal any patterns as well. The only statistically significant result is for measures *total views* and *average dimensions per view* in group Latency (with an R-value of -0.6 and a P-value of 0.007).

In another attempt, a clustering method was utilized to separate participants of each group into two subgroups, all based on the measures evaluated for each participant. Nevertheless, this approach also did not produce consistent results as the clustering outcome was highly dependent on the choice of initial randomization values.

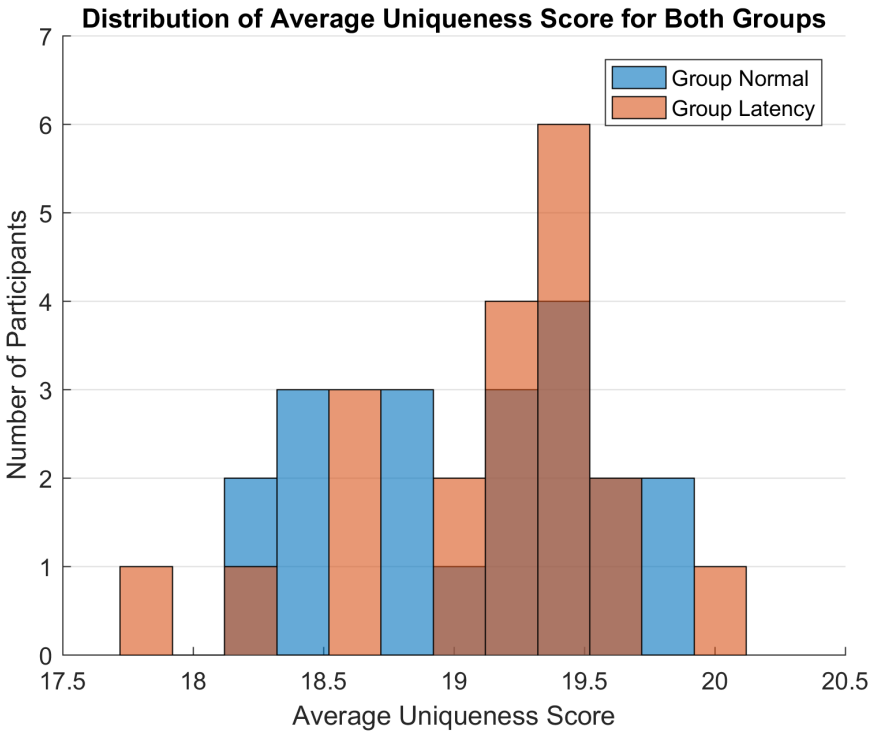


Figure 22: Distribution of average uniqueness scores for both groups.

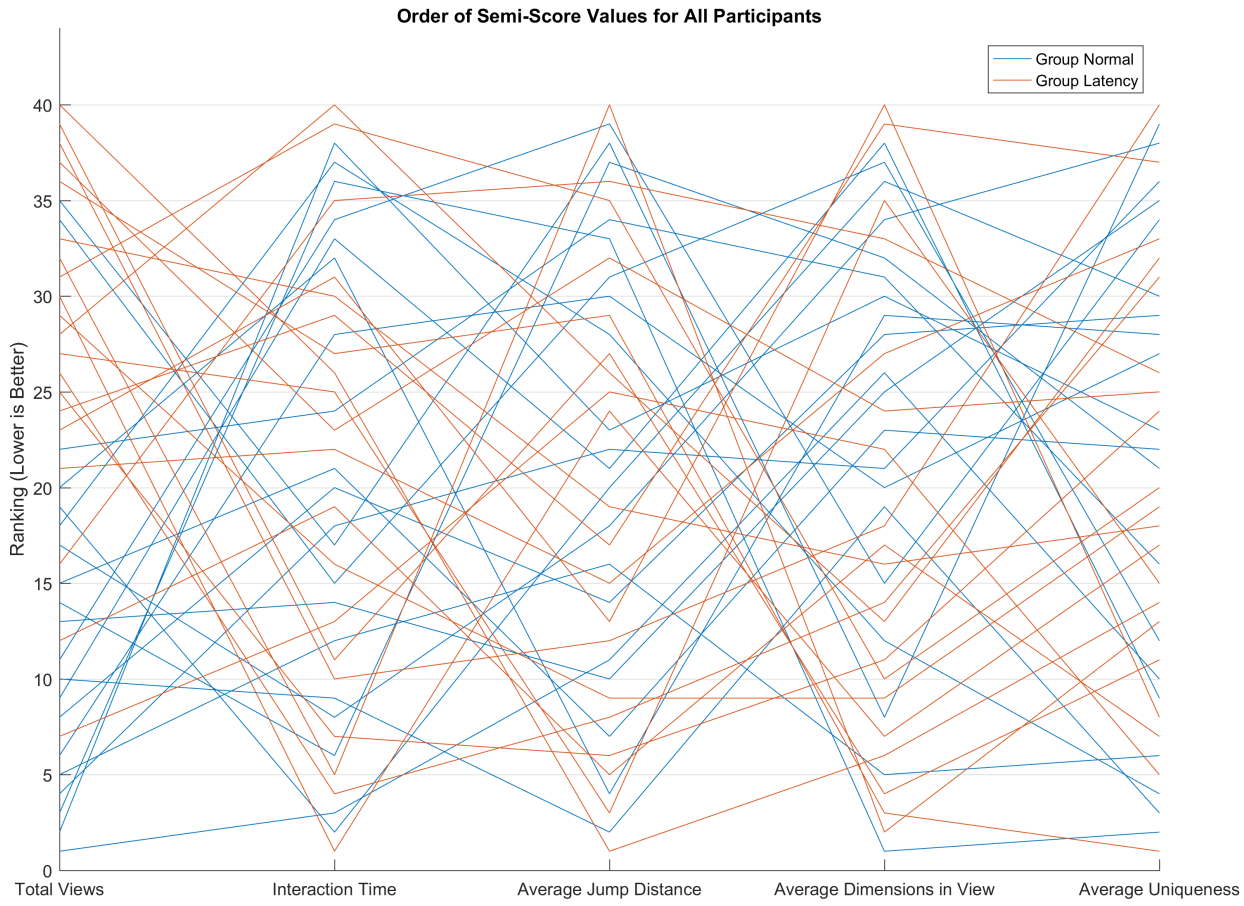


Figure 23: Rankings of all participants for each measure.

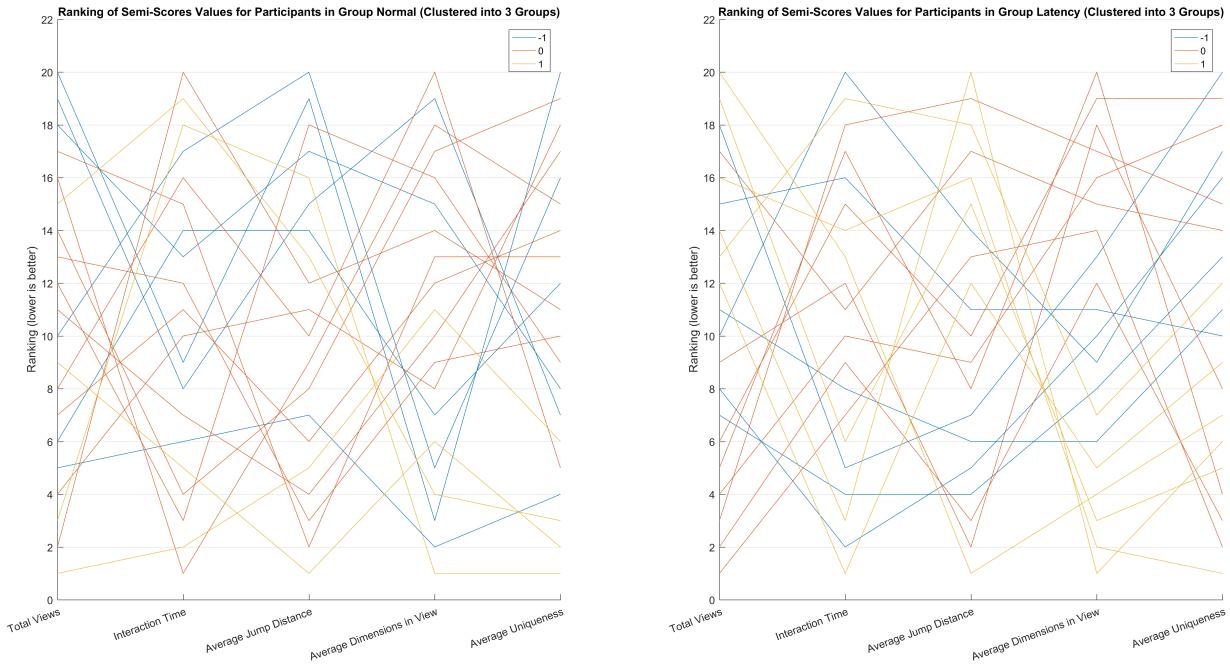


Figure 24: Rankings of measures for all participants, clustered into three categories.

A third approach was then utilized to evaluate if stable clusters can be built by repeating the clustering algorithm over several iterations. Here, we first build a graph for each group where each node represents one participant. We then run a clustering algorithm (generating two clusters) with a random initial state and then observe which participants end up in different groups. The value of the edges between such participants is then increased, indicating that they should be distanced from each other. We then repeat this process for 1000 times. Then, a nonlinear program solver was utilized to solve a spring-damper model to sort all participants based on a singular ranking. They were then divided into three groups, based on their closeness scores. This approach produced more consistent results. Figure 24 represents a clustering of participants into three subgroups based on this algorithm.

An initial observation suggests that *average uniqueness* is a good measure for consistently ranking participants. Also, for group Latency, *average number of dimensions* seems to be another reasonable measure. Nevertheless, Spearman's correlation between the derived

subgroups does not produce consistent results, as the results change dramatically from one initial condition in clustering to another. Therefore, it is not possible to conclude if some measures are correlated.

In another assessment, we looked at each measure and tried to investigate whether the two distant clusters of participants suggested by each score (in each group) are from distributions with different medians or not. Utilizing the Mann-Whitney U-Test (MW U-Test), none of the measures show any significant results, and the results also change from one iteration of the clustering to the next. Overall, it was not obvious how these measures would indicate to a consistent underlying phenomenon. Consequently, this indirect method was abandoned for a direct approach of marking discoveries.

4.3 MARKING DISCOVERIES: DIRECT APPROACH

As it was discussed in the previous section, the indirect method for measuring participant performance did not result in reliable metrics. Consequently, we decided to measure discovery events by looking at all 40 think-aloud videos and mark discoveries as they happen.

Marking discoveries requires a coding framework with a precise definition of when a discovery event happens. Chen et al. [21] define an insight (i.e., a discovery in our case) as a complex concept about the data combined with the contextual knowledge already in the mind of the user. This process is furthermore accompanied by a subjective and objective analysis of the fact. The insight is then the combination of fact, contextual knowledge, and evaluation of it. They also provide a comprehensive taxonomy of the types of facts:[3]

Value Most people in downtown Pittsburgh, earn more than $3K per month.
Distribution People in the cities earn more than people in the suburbs.
Difference There are more jobs for people with a bachelors degree.
Extreme This neighborhood in Pittsburgh has the lowest number of jobs.
Rank Retail jobs are the number one source of income in Pennsylvania.

[3]Here, the examples are altered to represent the LEHD dataset used in our experiment.

Categories There are two categories of jobs, one for people with high education and the other one for the rest.

Cluster People in the cities have similar characteristics.

Outliers People who work in healthcare industry in the city of Pittsburgh are earning more than other healthcare professionals.

Association The higher the educational attainment, the higher the income.

Trend Income is growing over the years.

Meta Fact The first seven years of data on education, is missing.

Compound Fact The correlation between high education and high income, is different in the cities compared to the rural areas.

In our coding framework, anytime a user mentions any of these types of facts, we consider it a discovery event. Note that we do not differentiate between various types of facts as we intend to increase the power of our analysis. The only additional measure we take is whether the discovery was true or false. This is done by looking at the view generated by the user and evaluating whether the interpretation mentioned by the user is a correct interpretation of the view and supported by the data or not. It is also worth noting that in time series datasets, it is possible to observe spurious correlations between the variables under study (for instance, if they are both correlated with time). Nevertheless, in our study, we have assumed that discovery events are independent from each other.

The coding framework is summarized as follows:

Event Mark the time when a user makes a discovery.

- The discovery can be explicitly announced by the participant, e.g., *"there is a positive correlation between income and education"*, or *"I think my question about the relationship between income and education cannot be answered with this tool"*.
- Some participants did not talk very much. In such cases they were prompted from time to time and asked about what they saw and if they had made any discoveries, e.g., "What are you seeing now?", *"I see income and education are positively correlated"*.

Correctness Assess whether the discovery is correct or incorrect.

- Sometimes, participants believe they have found something in the data while their conclusion is wrong. To measure the quality of their discoveries, we measure whether each discovery was right or not. For example, participant assigns the percentage of jobs with high education to Z axis on a map view but then interprets high elevations points as places where many people live (which is an incorrect interpretation) and also have high education. As another example, participant creates a scatter plot of percentage of education vs. percentage of wealth. Each dot represents a neighborhood. Therefore, the correct interpretation is that neighborhoods with high education levels tend to have a high income. However, the participant says *individuals* who have high income also have high education, which is an incorrect interpretation.

A Matlab Graphical User Interface software was built to mark discovery events and synchronize their occurrence time with user logs. Figure 25 demonstrates a screenshot of this tool. A sample of the resulted tagged data is shown in Figure 26. These discovery markings in conjunction with the log of acceptable views (extracted from the raw data logs) provide the necessary data for the analysis presented in the next sections.

Figure 27 presents the discovery events for all participants. Each line corresponds to one participant where blue lines are for group Normal and red lines are for group Latency. Every time participants make a correct discovery, they gain a score, and every time they make an incorrect discovery, they lose a score.

4.4 PERFORMANCE MEASURES

Marking discoveries provides the raw material necessary for evaluating participants' performance. The next step is to define performance measures derived from these discovery markings. Chan et al. [18] provide such a set of performance measures. In their paper they propose several factors for assessing the quality of discoveries:

- **Quantity**: Quantity of ideas generated is one of the key indicators of creativity.
- **Quality**: An expert in the domain assesses each discovery and evaluates its quality.

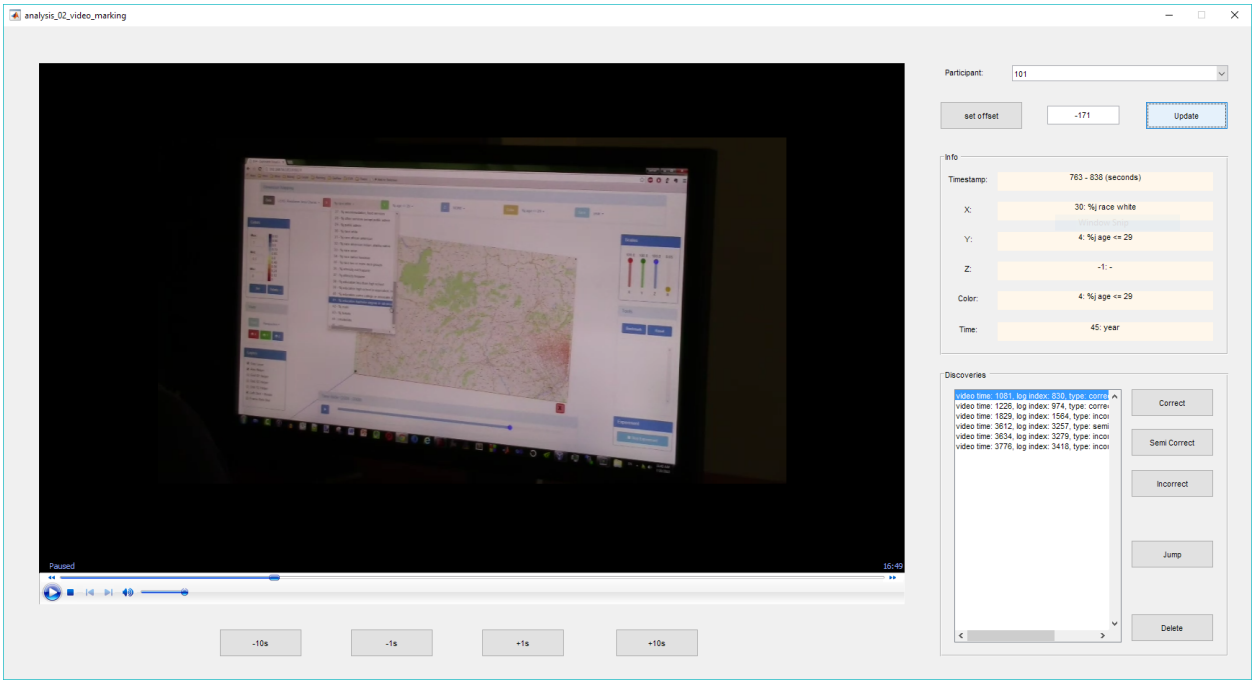


Figure 25: This software was built and used to watch the video of participants, synchronize it with the logs saved during the experiment, and then tag discovery events.

```
{
  "time_index": 847,
  "video_time": 862.1586008,
  "discovery_score": 1,
  "offset": 12
}
```

Figure 26: This Figure represents a sample line of video markings data. Each discovery is represented by a similar set of values. These values indicate which line of the raw JSON log is corresponding with this discovery, and which second in the video is corresponding with that JSON log. Also, it shows whether the discovery was correct or incorrect.

- **Breadth**: The percentage of the total space of possible solutions that was searched by a participant.
- **Novelty**: The percentage of participants who generated this idea.

Inspired by this work, we adopted an altered version of these measures, tailored to EVA's unique experiment settings. Henceforth, our performance measures are:

- **Quantity**: the percentage of views which result in a discovery. Note that by measuring the percentage of views instead of the actual number of discoveries, we are normalizing for the waiting time imposed on group Latency.
- **Correctness**: the percentage of discoveries that are factually correct.
- **Depth**: the number of dimensions selected while making a discovery. The assumption is that discoveries with higher-dimensional views present more complex facts (i.e., facts that include several aspects of the data at the same time). Also, we look at map-based dimensions vs. non-map-based ones as another way of creating more in-depth discoveries. Here, the assumption is that non-map-based views are harder to interpret and require an in-depth analysis of the data.

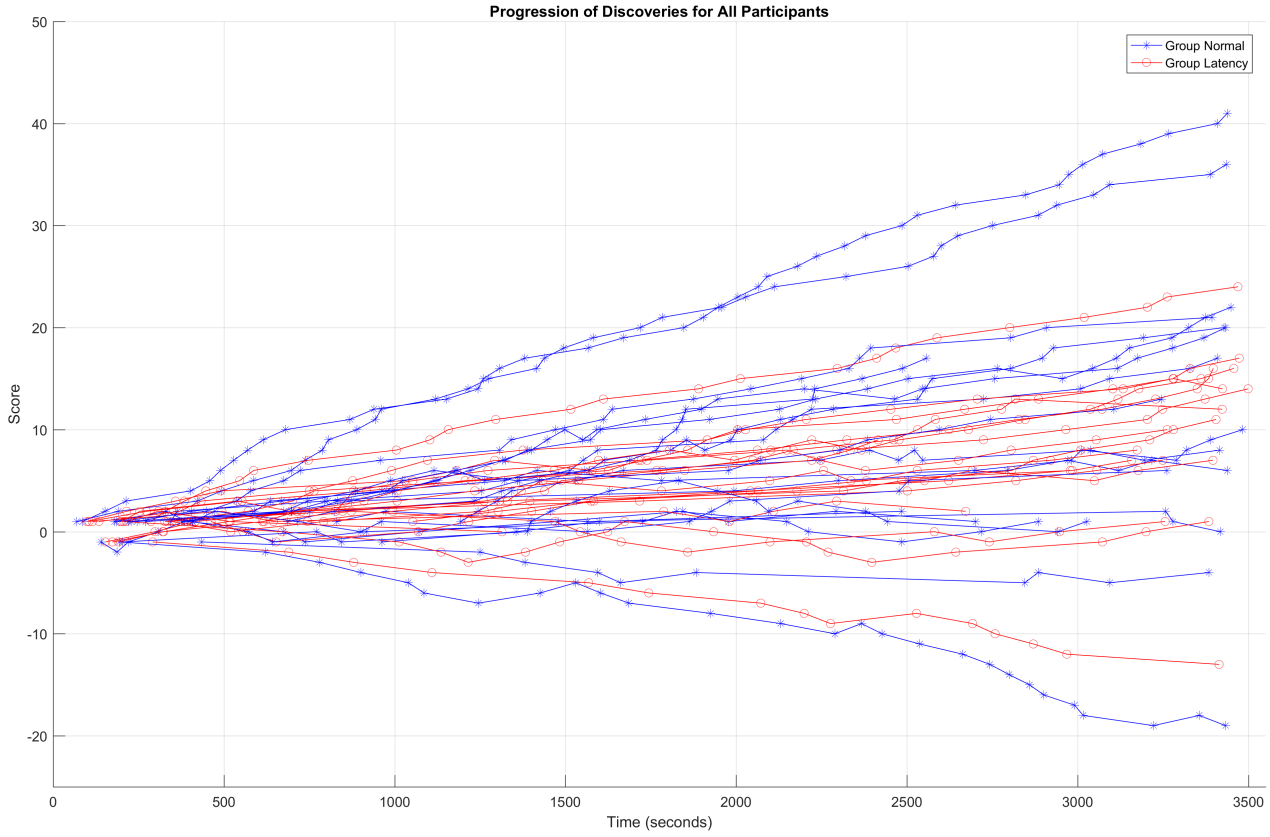


Figure 27: Each line represents the performance of one participant during the experiment. Every time they make a correct discovery, they gain a score. Every time they make an incorrect discovery, they lose a score.

- **Breadth**: the percentage of unique discoveries. A unique discovery is one that only one participant in a group has made. The aim here is to measure if one group generates more unique discoveries and therefore covers a larger subset of the discovery space.

4.5 EFFECT OF LATENCY ON PERFORMANCE

The next sections evaluate how latency affects knowledge discovery in big data. Our analysis is based on the four performance measures of quantity (Section 4.5.1), correctness (Section 4.5.2), depth (Section 4.5.3), and breadth (Section 4.5.4) of discoveries. Section 4.5.5 discusses the role of other factors such as demographics and experience of participants in their performance.

4.5.1 Quantity of Discoveries

The first performance measure calculates a participant's chance of making a discovery per view. Table 9 represents the number of views and discoveries generated by each group. Group Normal participants had a 51% chance of making a discovery in each view, while in group Latency, this chance was 52%. Table 10 represents the contingency table for the number of views with and without a discovery for each group. The Chi-Squared Test (CH-Test) results in a P-Value of 0.1450; therefore, we cannot reject the no nonrandom association between the categorical variables. We can then conclude that latency does not affect a participant's chance of making a discovery per each view generated.

4.5.2 Correctness of Discoveries

This performance measure calculates the percentage of discoveries that are factually correct. Table 11 represents the contingency table for the number of correct and incorrect discoveries for each group. 80.5% of discoveries in group Normal and 81.6% of discoveries in group Latency are correct. The CH-Test results in a P-value of 0.7579; therefore, we cannot reject the no nonrandom association between the categorical variables. We can then conclude

Table 9: This table shows the percentage of views that result in discoveries in each group.

	Discoveries	Views	Discoveries/Views
Group Normal	359	703	0.51
Group Latency	267	509	0.52

Table 10: This contingency table represents the portion of views in each group that lead to a discovery. The odds ratio is 0.8433 and the P-value for the Chi-Squared Test is 0.1450. Hence, both groups have the same chance of making a discovery per view.

	Views with a Discovery	Views without a Discovery
Group Normal	310	393
Group Latency	246	263

Table 11: This contingency table represents the portion of correct discoveries in each group. The odds ratio is 0.9280 and the P-value for the Chi-Squared Test is 0.7579. Hence, both groups have the same chance of making a correct discovery.

	Correct Discoveries	Incorrect Discoveries
Group Normal	289	70
Group Latency	218	49

that latency does not affect a participant's chance of making a correct discovery per each discovery they make.

Table 12 represents the average and standard deviation of scores in each group. A participant's score is the number of correct discoveries she has made minus the number of incorrect ones. A KS-Test of normality validates that the score values of participants' in both groups follow a normal distribution (a P-value of 0.95 for group Normal and 0.97 for group Latency). A Two-Sampled F-Test shows these distributions have a different variance (P-value of 0.01). A Two-Sided Unpaired Welch Corrected T-Test is not able to reject the null hypothesis of distributions having similar means (P-value of 0.50). Hence, we can conclude that the score of participants in both groups Normal and Latency follow a distribution with similar scores but a different variance. This suggests that the added speed of the Normal group does not inherently change participants' ability to reach correct discoveries.

4.5.3 Depth of Discoveries

The depth of a discovery can be viewed from multiple perspectives. The aim is to measure the complexity of each discovery and correspond more complex ones as an in-depth analysis of the data. This complexity is assessed via two approaches. First, we look at the map-based views. In general, it was easier for participants to analyze map-based views. The familiar layer of

Table 12: Average and standard deviation of scores in each group.

	Average	Standard Deviation
Group Normal	10.15	13.91
Group Latency	7.75	7.40

the map provided a contextual layer that shaped many of their discoveries. Nevertheless, some participants tried to create abstract (i.e., non-map-based) views. Such views required more effort to interpret, and as such, their corresponding discoveries are considered more complex. We will also look at the number of dimensions in each view as another way of assessing its complexity. Participants had an easier time with 2 or 3-dimensional views, but some generated 5-dimensional views as well. Discoveries resulted from such views are considered as another type of complex discoveries.

4.5.3.1 Maps-Based Views Table 13 represents the number of map-based and non-map-based views for each group. 53% of views in group Normal and 48% of them in group Latency were map-based. CH-Test results in a P-value of 0.1160; therefore, we cannot reject the no nonrandom association between the categorical variables. Hence, we can conclude that latency does not change the percentage of map views participants generate.

Table 14 represents the contingency table for the number of discoveries for map and non-map views in group Normal. The corresponding CH-Test has a P-value of 0.8268. It, therefore, suggests that in group Normal, generating a map view does not increase one's chance of making a discovery (correct or incorrect). Similarly, Table 15 represents the contingency table for the number of discoveries for map and non-map views in group Latency. The corresponding CH-Test has a P-value of 0.1408. It, therefore, suggests that in group Latency too, generating a map view does not increase one's chance of making a discovery.

Table 16 represents the contingency table for the number of correct discoveries vs. in-

Table 13: This contingency table represents the number of views in each group with and without a map layer. The odds ratio is 1.2084, the confidence interval is $[0.9617, 1.5185]$ and the P-value for the Chi-Squared Test is 0.1160. Hence, both groups have the same chance of making a map view.

	Map Views	Non-Map Views
Group Normal	373	330
Group Latency	246	263

Table 14: This contingency table represents the relationship between map and non-map views with discoveries in group Normal. The odds ratio is 1.0331 and the P-value for the Chi-Squared Test is 0.8268. Hence, this suggests that in group Normal, being in a map view does not increase one's chance of making a discovery.

	Discovery	No Discovery
Map View	192	207
Non-Map View	167	186

Table 15: This contingency table represents the relationship between map and non-map views with discoveries in group Latency. The odds ratio is 1.2945 and the P-value for the Chi-Squared Test is 0.1408. Hence, this suggests that in group Latency, being in a map view does not increase one's chance of making a discovery.

	Discovery	No Discovery
Map View	138	119
Non-Map View	129	144

correct ones for map and non-map views in group Normal. The corresponding CH-Test has a P-value of 0.0322, rejecting the null hypothesis of no nonrandom association between the categorical variables. It, therefore, suggests that in group Normal, being in a map view, there is a 1.8 times higher chance of making a correct discovery (vs. an incorrect one). Similarly, Table 17 represents the contingency table for the number of correct discoveries vs. incorrect ones for map and non-map views in group Latency. The corresponding CH-Test has a P-value of 0.00, rejecting the null hypothesis of no nonrandom association between the categorical variables. It, therefore, suggests that in group Latency, being in a map view, there is a 6.4 times higher chance of making a correct discovery (vs. an incorrect one). Table 18 represents a contingency table for the number of correct and incorrect discoveries in map views, for both groups. The corresponding CH-Test has a P-value of 0.022, rejecting the null hypothesis of no nonrandom association between the categorical variables. It, therefore, suggests that in comparing both groups, there is a 2.55 times higher chance of making a correct discovery in a map view in group Latency vs. group Normal.

What these findings suggest is that map views significantly increase one's chance of making correct discoveries, especially when we have latency. Analyzing these results from a different perspective, we can conclude that the faster approach of the Normal group can be more suitable for cases where we need to make abstract discoveries outside of the contextual

Table 16: This contingency table represents the relationship between map and non-map views with correct and incorrect discoveries in group Normal. The odds ratio is 1.8290 with a confidence interval of $[1.0772, 3.1053]$, and the P-value for the Chi-Squared Test is 0.0322 . Hence, this suggests that in group Normal, being in a map view, there is a 1.8 times higher chance of making a correct discovery.

	Correct Discovery	Incorrect Discovery
Map View	163	29
Non-Map View	126	41

layer of a map.

4.5.3.2 Number of Dimensions Figure 28 represents a histogram per group on the number of dimensions per each view in that group and their corresponding frequency. All views generated by the participants in a group are pooled together. Furthermore, each histogram demonstrates what portion of those views leads to no discoveries, correct discoveries, or incorrect ones.

We first focus on the distribution of the number of dimensions per view in each group. The KS-Test of normality for both groups results in a P-value of 0.00, suggesting that none of these distributions are normally distributed. The One-Sided Mann-Whitney U-Test of comparing these two distributions results in a P-value of 0.0384, suggesting that the distribution of the number of dimensions in group Normal has a lower median than the similar distribution in group Latency. This result indicates that when participants have to wait (i.e., the latency mode), they tend to generate more complex views.

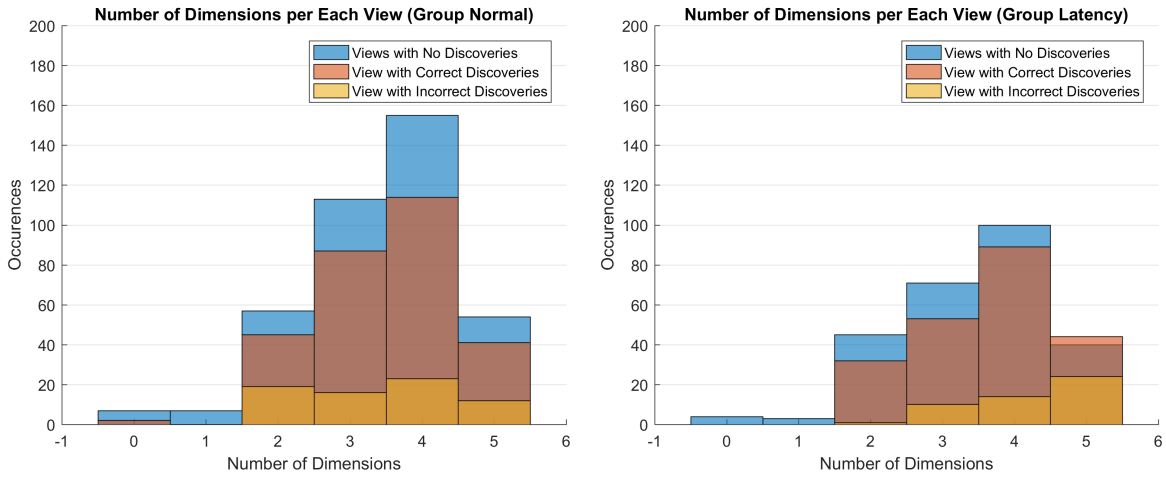
Continuing our analysis, we then look at the six distributions of the number of dimensions for views with no discoveries, views with correct discoveries and views with incorrect discoveries, both for group Normal and Latency. The KS-Test of normality results in a P-

Table 17: This contingency table represents the relationship between map and non-map views with correct and incorrect discoveries in group Latency. The odds ratio is 6.4419 with a confidence interval of $[2.9771, 13.9394]$ and the P-value for the Chi-Squared Test is 0.00. Therefore, this suggests that in group Latency, being in a map view, there is a 6.4 times higher chance of making a correct discovery.

	Correct Discovery	Incorrect Discovery
Map View	129	9
Non-Map View	89	40

Table 18: This contingency table represents the relationship between group Normal and Latency, and correct and incorrect map-based discoveries. The odds ratio is 0.3921 with a confidence interval of $[0.1793, 0.8578]$ and the P-value for the Chi-Squared Test is 0.022. Therefore, this suggests that when users generate map views, there is 2.55 times higher chance of making a correct discovery in group Latency in comparison to group Normal.

	Correct Discovery in Map	Incorrect Discovery in Map
Group Normal	163	29
Group Latency	129	9



(a) Histogram of Dimension per View for Group Normal

(b) Histogram of Dimensions per View for Group Latency

Figure 28: This figure represents how many dimensions have been selected per each view for each group. It also shows what portion of the views lead to no discoveries, which portion lead to correct discoveries and which portion to incorrect ones.

value of 0.00 for all these distributions, suggesting they do not follow a normal distribution. Therefore, we need to use Mann-Whitney U-Test to compare these distributions.

In group Normal, the U-Test for comparing views with no discoveries with views with correct discoveries results in a P-value of 0.67. Comparing the distribution of views with no discoveries with views with incorrect discoveries, we get a P-value of 0.61. Finally, comparing the distribution of views with incorrect discoveries with views with correct discoveries results in a P-value of 0.43. This suggests that in group Normal, the number of dimensions is similar between views with no discoveries, views with correct discoveries and views with incorrect discoveries. In other words, irrespective of the number of dimensions, a user has the same chance of making a correct or incorrect discovery in group Normal.

In group Latency, the One-Side U-Test of comparing the distribution of views with no discoveries with the distribution of views with correct discoveries results in a P-value of 0.04. A similar test on comparing views with no discoveries with views with incorrect discoveries results in a P-value of 0.00. Also, comparing views with incorrect discoveries with views with correct discoveries results in a P-value of 0.00. In all these cases, the null hypothesis of equal median between the distributions is rejected. This suggests that in group Latency, the median number of dimensions for views with no discoveries is less than the median number of dimensions of views with correct discoveries, which is itself less than the median of the number of dimensions of views with incorrect discoveries.

In comparing the two groups, the U-Test between the views with no discoveries in group Normal with similar views in group Latency results in a P-value of 0.9577, suggesting that views that do not result in a discovery have a similar level of complexity (dimension-wise) in both groups. A One-Sided U-Test between the views with incorrect discoveries in group Normal with similar views in group Latency results in a P-value of 0.00, suggesting that incorrect discoveries in group Latency are more complex than in group normal. The U-Test between the views with correct discoveries in group Normal and Latency results in a P-value of 0.0629, suggesting that correct discoveries have a similar level of complexity in both groups.

Our analyses indicate that in group Normal, there is no statistically significant difference between the number of dimensions in a view and the chance of making a correct, incorrect,

Table 19: This contingency table represents the portion of unique views in each group. The odds ratio is 0.9283 and the P-value for the Chi-Squared Test is 0.5526. Hence, both groups have the same chance of generating unique views.

	Unique Views	Non-Unique Views
Group Normal	417	286
Group Latency	311	198

or no discovery. On the other hand, in group Latency, participants have more discoveries in complex views, yet if the view becomes too complicated (i.e., high-dimensional), participants will generate more mistakes. Also, comparing the two groups together, they only differ in the complexity of incorrect views. There, latency makes participants generate more mistakes when they work with higher dimensional views.

4.5.4 Breadth of Discoveries

This performance measure calculates the percentage of views that lead to a unique discovery, where a unique discovery is one that only one participant in that group has made. The assumption is that if the participants of a group produce more unique discoveries, then those discoveries are covering a wider range of the explorable space and hence that group is more productive.

Table 19 represents the contingency table for the number of unique and non-unique views produced by each group. 59.3% of views in group Normal and 61.1% of views in group Latency are unique. The CH-Test results in a P-value of 0.5526, therefore, we cannot reject the no nonrandom association between the categorical variables. We can then conclude that latency does not affect the number of unique views a participant generates.

Table 20 represents the contingency table for the number of views with and without a

Table 20: This contingency table represents the relationship between unique views and the chance of making a discovery in a view. The odds ratio is 1.3204 with a confidence interval of $[0.9858, 1.7686]$ and the P-value for the Chi-Squared Test is 0.0638. Therefore, with a significance level of 5%, we cannot reject the null hypothesis of no nonrandom association between uniqueness of a view and the chance of making a discovery in that view in group Normal.

	Views with Discovery	Views without Discovery
Unique Views	225	220
Non-Unique Views	134	173

discovery and the number of unique and non-unique views in group Normal. The CH-Test results in a P-value of 0.0638, therefore, we cannot reject the no nonrandom association between the categorical variables. We can then conclude that in group Normal, uniqueness of a view is not correlated with the chance of making a discovery in that view.

Table 21 represents the contingency table for the correctness or incorrectness of discoveries and the uniqueness of their corresponding views in group Normal. 76.9% of discoveries in unique views are correct while 86.6% of discoveries in non-unique views are correct. The CH-Test results in a P-value of 0.0276, therefore, rejecting the null hypothesis of no nonrandom association between the categorical variables. We can then conclude that in group Normal, non-unique views (i.e., common views) result in 1.94 times more correct discoveries than unique views.

Table 22 represents the contingency table for the number of views with and without a discovery and the number of unique and non-unique views in group Latency. The CH-Test results in a P-value of 0.3273, therefore, we cannot reject the no nonrandom association between the categorical variables. We can then conclude that in group Latency, uniqueness of a view is not correlated with the chance of making a discovery in that view.

Table 21: This contingency table represents the relationship between unique views and the chance of making a correct discovery in that view in group Normal. The odds ratio is 0.5162 with a confidence interval of $[0.2875, 0.9270]$ and the P-value for the Chi-Squared Test is 0.0276. Therefore, we can reject the null hypothesis and conclude that in group Normal, there is a 1.94 times higher chance of making a correct discovery vs. an incorrect one, when the view is not unique.

	Views with Correct Discovery	Views with Incorrect Discovery
Unique Views	173	52
Non-Unique Views	116	18

Table 22: This contingency table represents the relationship between unique views and the chance of making a discovery in a view. The odds ratio is 1.2016 with a confidence interval of $[0.8470, 1.7046]$ and the P-value for the Chi-Squared Test is 0.3273. Therefore, we cannot reject the null hypothesis of no nonrandom association between uniqueness of a view and the chance of making a discovery in that view in group Latency.

	Views with Discovery	Views without Discovery
Unique Views	169	155
Non-Unique Views	98	108

Table 23: This contingency table represents the relationship between unique views and the chance of making a correct discovery in that view in group Latency. The odds ratio is 0.1173 with a confidence interval of $[0.0407, 0.3375]$ and the P-value for the Chi-Squared Test is 0.000. Therefore, we can reject the null hypothesis and conclude that in group Latency, there is a 8.53 times higher chance of making a correct discovery vs. an incorrect one, when the view is not unique.

	Views with Correct Discovery	Views with Incorrect Discovery
Unique Views	124	45
Non-Unique Views	94	4

Table 23 represents the contingency table for the correctness or incorrectness of discoveries and the uniqueness of their corresponding views in group Latency. 73.4% of discoveries in unique views are correct while 95.9% of discoveries in non-unique views are correct. The CH-Test results in a P-value of 0.000, therefore, rejecting the null hypothesis of no nonrandom association between the categorical variables. We can then conclude that in group Latency, non-unique views result in 8.53 times more correct discoveries than unique views.

Table 24 represents the contingency table for the number of correct discoveries in unique and non-unique views for each group. The CH-Test results in a P-value of 0.5245, therefore, we cannot reject the no nonrandom association between the categorical variables. We can then conclude that latency does not increase one's chance of making correct discoveries in unique views vs. non-unique views.

4.5.5 Other Affecting Factors

All participants answered a questionnaire on their background and experience with data analytics tool. Here, we will examine these factors and evaluate if they have any effect on

Table 24: This contingency table represents the relationship between correct discoveries in unique and non-unique views in both groups. The odds ratio is 1.1306 with a confidence interval of $[0.7914, 1.6150]$ and the P-value for the Chi-Squared Test is 0.5245. Therefore, we cannot reject the null hypothesis of no nonrandom association between latency and the chance of making a correct discovery in unique vs. non-unique views.

	Correct Discoveries in Unique Views	Correct Discoveries in Non-Unique Views
Group Normal	173	116
Group Latency	124	94

the Quality Performance Measure.

Each group had 20 participants. 15 participants in group Normal and 16 in group Latency were graduate students. The age median was 27.5 for group Normal and 24.5 for group Latency. 14 participants in group Normal had an education in Computer Science or a related field, while 15 participants in group Latency had a similar education. Each group had 7 female and 13 male participants. 18 participants in group Normal and 19 participants in group Latency had taken at least an undergraduate level course in statistics. In each group, 18 participants had worked with geospatial tools such as Google Earth. 14 participants in group Normal and 15 in group Latency had played 3D computer games. 9 participants in group Normal and 10 in group Latency had experience in working with 3D design tools such as AutoCAD. In each group, 19 participants had experience in working with data analytics tools such as Microsoft Excel. Only 3 participants in group Normal and 2 in group Latency had experience in working with visual data exploration tools such as Tableau Software. 3 participants in group Normal and 4 participants in group Latency had seen EVA before, but only 1 participant in group Latency had worked with EVA as well. 4 participants in group Normal and 9 in group Latency had worked with US Census or similar demographics data.

Table 25: This contingency table represents the relationship between views with correct discoveries and gender in group Normal. The odds ratio is 1.0839 and the P-value for the Chi-Squared Test is 0.6293. Therefore, we can conclude that in group Normal, both males and females have the same success rate in making correct discoveries.

	Views with a Correct Discovery	All Other Views
Male	201	314
Female	88	149

Only 1 participant in each group assumed themselves knowledgeable in the workforce and demographics information of the state of Pennsylvania. 14 participants in group Normal and 15 in group Latency mentioned they find visualization beneficial in finding new knowledge.

Table 25 represents the contingency table for the role of gender in making correct discoveries in group Normal. 39.0% of views generated by males and 37.1% of views generated by females in this group resulted in a correct discovery. The CH-Test results in a P-value of 0.6293, therefore, failing to reject the null hypothesis of no nonrandom association between the categorical variables. We can then conclude that in group Normal, gender does not affect the performance measure of quality (i.e., chance of making a correct discovery per view).

Table 26 represents the contingency table for the role of gender in making correct discoveries in group Latency. 39.8% of views generated by males and 43.5% of views generated by females in this group resulted in a correct discovery. The CH-Test results in a P-value of 0.4072, therefore, failing to reject the null hypothesis of no nonrandom association between the categorical variables. We can then conclude that in group Latency, gender does not affect the performance measure of quality.

Table 27 represents the contingency table on the effect of previous exposure to 3D tools in making correct discoveries in group Normal. Participants who had used a 3D tool before had a 42.5% chance of making a correct discovery in a view, while others' chance was 34.7%.

Table 26: This contingency table represents the relationship between views with correct discoveries and gender in group Latency. The odds ratio is 0.8579 and the P-value for the Chi-Squared Test is 0.4072. Therefore, we can conclude that in group Latency, both males and females have the same success rate in making correct discoveries.

	Views with a Correct Discovery	All Other Views
Male	137	207
Female	81	105

The CH-Test results in a P-value of 0.0297, therefore, rejecting the null hypothesis of no nonrandom association between the categorical variables. We can then conclude that in group Normal, participants who had previously worked with 3D tools were 1.39 times more likely to make correct discoveries in a view.

Table 28 represents the contingency table on the effect of previous exposure to 3D tools in making correct discoveries in group Latency. Participants who had used a 3D tool before had a 44.2% chance of making a correct discovery in a view, while others' chance was 37.4%. The CH-Test results in a P-value of 0.1314, therefore, failing to reject the null hypothesis of no nonrandom association between the categorical variables. We can then conclude that in group Latency, having previous exposure to 3D tools does not affect one's chance of making correct discoveries.

Table 29 represents the contingency table on the effect of previous exposure to EVA (i.e., if the participant has seen EVA before) in making correct discoveries in group Normal. Participants who had seen EVA before had a 55.8% chance of making a correct discovery in a view, while others' chance was 35.1%. The CH-Test results in a P-value of 0.000, therefore, rejecting the null hypothesis of no nonrandom association between the categorical variables. We can then conclude that in group Normal, participants who had seen EVA before (3 participants in total) were 2.3 times more likely to make correct discoveries in a view.

Table 27: This contingency table represents the relationship between views with correct discoveries and having previous experience in using 3D tools for group Normal. The odds ratio is 1.3913 with the confidence interval of $[1.0361, 1.8684]$ and the P-value for the Chi-Squared Test is 0.0297. Therefore, we can conclude that in group Normal, people who had used 3D tools before are 1.4 times more likely to make correct discoveries.

	Views with a Correct Discovery	All Other Views
Used 3D Tools Before	153	207
Not Used 3D Tools Before	136	256

Table 28: This contingency table represents the relationship between views with correct discoveries and having previous experience in using 3D tools for group Latency. The odds ratio is 1.3249 and the P-value for the Chi-Squared Test is 0.1314. Therefore, we can conclude that in group Latency, having prior experience with 3D tools does not affect one's chance of making correct discoveries.

	Views with a Correct Discovery	All Other Views
Used 3D Tools Before	129	163
Not Used 3D Tools Before	89	149

Table 29: This contingency table represents the relationship between views with correct discoveries and having previous exposure to EVA for group Normal. The odds ratio is 2.3347 with the confidence interval of $[1.5719, 3.4676]$ and the P-value for the Chi-Squared Test is 0.000. Therefore, we can conclude that in group Normal, people who had seen EVA before are 2.3 times more likely to make correct discoveries.

	Views with a Correct Discovery	All Other Views
Seen EVA Before	67	53
Not Seen EVA Before	222	410

Table 30 represents the contingency table on the effect of previous exposure to EVA in making correct discoveries in group Latency. Participants who had seen EVA before had a 55.2% chance of making a correct discovery in a view, while others' chance was 38.0%. The CH-Test results in a P-value of 0.0028, therefore, rejecting the null hypothesis of no nonrandom association between the categorical variables. We can then conclude that in group Latency, participants who had seen EVA before (4 participants in total) were 2.0 times more likely to make correct discoveries in a view.

Table 31 represents the contingency table on the effect of having prior experience in working with US Census data in making correct discoveries in group Normal. Participants who had worked with Census data before had a 41.8% chance of making a correct discovery in a view, while others' chance was 37.6%. The CH-Test results in a P-value of 0.3876, therefore, failing to reject the null hypothesis of no nonrandom association between the categorical variables. We can then conclude that in group Normal, having previous experience in working with Census data does not affect one's chance of making correct discoveries.

Table 32 represents the contingency table on the effect of having prior experience in working with US Census data in making correct discoveries in group Latency. Participants

Table 30: This contingency table represents the relationship between views with correct discoveries and having previous exposure to EVA for group Latency. The odds ratio is 2.0094 with the confidence interval of $[1.2857, 3.1405]$ and the P-value for the Chi-Squared Test is 0.0028. Therefore, we can conclude that in group Latency, people who had seen EVA before are 2.0 times more likely to make correct discoveries.

	Views with a Correct Discovery	All Other Views
Seen EVA Before	53	43
Not Seen EVA Before	165	269

who had worked with Census data before had a 40.1% chance of making a correct discovery in a view, while others' chance was 41.9%. The CH-Test results in a P-value of 0.7229, therefore, failing to reject the null hypothesis of no nonrandom association between the categorical variables. We can then conclude that in group Latency, having previous experience in working with Census data does not affect one's chance of making correct discoveries.

4.6 ANALYZING STRATEGIES: QUANTITATIVE APPROACH

The second objective of this thesis is to propose design guidelines for Human-Data Interaction tools, specifically with the aim of solving challenges that arise in working with large and complex datasets. To answer this objective, we can analyze how participants worked with EVA; in particular, what were the differences between successful and unsuccessful participants? Such differences can point to important features of the tool that helped successful participants to be more productive. It can also help us to recognize what are the missing features that could have helped unsuccessful participants achieve a better result. In analyzing

Table 31: This contingency table represents the relationship between views with correct discoveries and having previous experience in using US Census data for group Normal. The odds ratio is 1.1919 with the confidence interval of $[0.8212, 1.7298]$ and the P-value for the Chi-Squared Test is 0.3876. Therefore, we can conclude that in group Normal, having prior experience with Census data does not affect one's chance of making correct discoveries.

	Views with a Correct Discovery	All Other Views
Had Experience with Census	59	82
No Experience with Census	230	381

Table 32: This contingency table represents the relationship between views with correct discoveries and having previous experience in using US Census data for group Latency. The odds ratio is 0.9300 with the confidence interval of $[0.6566, 1.3170]$ and the P-value for the Chi-Squared Test is 0.7229. Therefore, we can conclude that in group Latency, having prior experience with Census data does not affect one's chance of making correct discoveries.

	Views with a Correct Discovery	All Other Views
Had Experience with Census	96	143
No Experience with Census	122	169

the user behavior, we take two approaches. This section focuses on a quantitative approach where we analyze user interaction logs and discovery markings to investigate how the strategies used by participants changed over time and if throughout the experiment, successful participants learned different strategies than others. In Section 4.7, we take a qualitative approach and analyze four extreme cases (i.e., very successful vs. very unsuccessful) in detail. There, our focus is on the data exploration patterns utilized by each participant, and we will analyze how the differences between those strategies lead to very different outcomes.

In the quantitative approach, we look at six measures to gauge the change in user behavior over time. Section 4.6.1 discusses how the rate of discoveries changes during the experiment. Section 4.6.2 investigates if the amount of time participants interact with the tool changes as they become more familiar with the tool. Another measure, discussed in Section 4.6.3, is the similarity of consecutive discoveries. This measure indicates if participants generate coherent stories or if their discoveries are a series of unrelated facts. Sections 4.6.4 and 4.6.5 focus on the complexity of discoveries generated and answer if the discoveries generated by the end of the experiment are more abstract and high-dimensional or not. Finally, Section 4.6.6 looks at the diversity of discoveries and measures if the rate of generating unique discoveries changes over time.

Each of these measures calculates the change ratio of the underlying phenomenon they are representing. The change ratio is defined as:

$$\text{change ratio} = \frac{(measure_{\text{second half}} - measure_{\text{first half}})}{measure_{\text{first half}}} \tag{4.1}$$

Here, a participant's experiment is divided into two halves. Then the measure is calculated for each half, and the resulting change ratio is compared against the participant score which is the number of correct discoveries minus incorrect ones. Change ratio was used instead of just differencing first half from the second half to further emphasize on the performance gain or loss of a participant against herself. For example, if we just measure differences, a participant who has made 20 discoveries in the first half and 24 discoveries in the second half of the experiment, would look similar to someone who made 2 discoveries in the first half and 6 ones in the second half. Yet the second participant has become 3 times more productive in the second half in comparison to herself. Such patterns can be better

reflected by the change ratio of the measure in question.

After calculating the change ratio of each measure, we answer three main questions:

1. How this measure changes over time?
2. Is this change correlated with the participant score?
3. Does latency affect this correlation?

Also, to increase the power of our analysis, besides investigating each group separately, we will look at the combined values of all participants as well.

4.6.1 Rate of Discoveries

Rate of discoveries measures the change ratio of the number of discoveries made in each half of the experiment. The aim is to investigate whether participants become more productive as they use the tool and if this behavior is different for successful participants.

4.6.1.1 Change of Measure over Time First, we look at each group separately. In each group, we examine the total number of discoveries made by each participant in the first half and the second half of the experiment. This results in two distributions for each half, each containing 20 numbers for all the participants in that group. We then measure if these distributions are from the same set of parameters or not. This result indicates whether the discovery rate of participants in that group changes significantly as they become more familiar with the tool.

In group Normal, the KS-Test of normality confirms that both distributions are normal (P-value of 0.39 for the distribution in the first half, and a P-value of 0.47 for the distribution in the second half). An F-Test yields a P-value of 0.86, confirming that both distributions have similar variances. The corresponding Unpaired T-Test results in a P-value of 0.60, suggesting that in group Normal, in both the first half and the second half of the experiment, the total number of discoveries made by participants comes from similar normal distributions. In other words, the rate of discoveries in group Normal does not change during the experiment.

In group Latency, the KS-Test confirms that both distributions are normal (P-value of 0.75 for the distribution in the first half, and a P-value of 0.70 for the distribution in the second half). An F-Test yields a P-value of 0.6, confirming that both distributions have similar variances. The corresponding Unpaired T-Test results in a P-value of 0.39, suggesting that in group Latency, in both the first half and the second half of the experiment, the total number of discoveries made by participants comes from similar normal distributions. In other words, the rate of discoveries in group Latency does not change during the experiment.

4.6.1.2 Correlation with Performance Score Figure 29 represents the change ratio of the rate of discoveries for all participants in conjunction with their scores. The R-value of the correlation for the combined group (both group Normal and Latency) is -0.01 (with a P-value of 0.96). The R-value for group Normal is -0.05 (with a P-value of 0.84) and for group Latency is 0.07 (with a P-value of 0.78). These results indicate that there is no significant correlation between a participant's score (i.e., her performance) and the change ratio of the number of discoveries they make in the first and second half of the experiment.

Instead of assessing the correlation between the change ratio of the number of discoveries and participant scores, we can combine participants into two groups of high performers (top 10 participants with the highest scores) and low performers (bottom 10 participants with the lowest scores) and then measure if they produce different discoveries in each half of the experiment. Table 33 represents the contingency table for the number of discoveries made in each half by high performers and low performers in group Normal. The P-value for the CH-Test is 0.8242. Therefore, we cannot reject the null hypothesis of no nonrandom association between the categorical variables and can conclude that in group Normal, participants generate discoveries with a consistent rate throughout the experiment, no matter what is their performance level. A similar table for group Latency is presented in Table 34. The P-value for the CH-Test is 0.7062. Therefore, we cannot reject the null hypothesis of no nonrandom association between the categorical variables and can conclude that in group Latency too, participants generate discoveries with a consistent rate throughout the experiment, no matter what is their performance level.

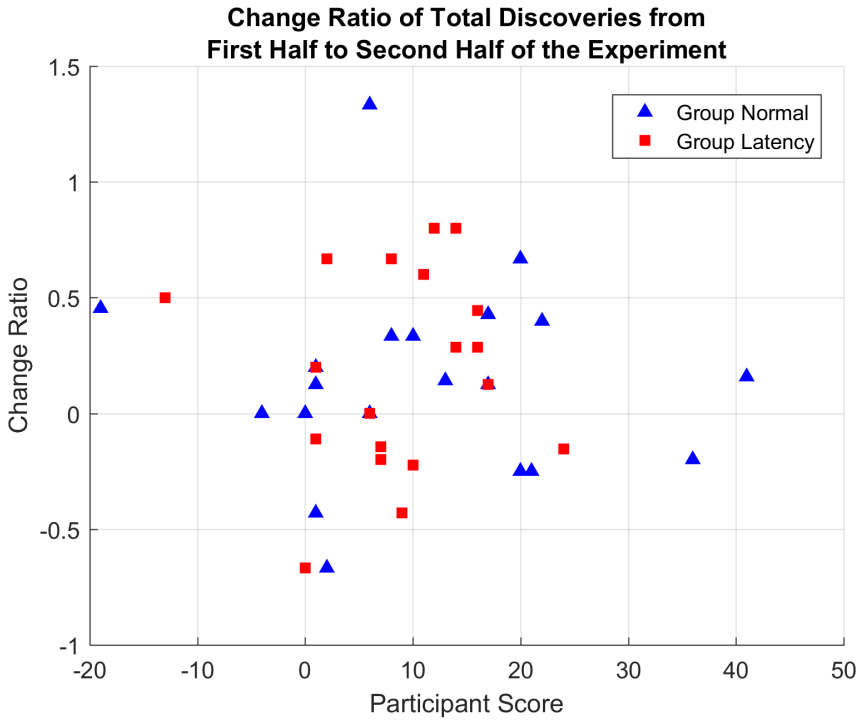


Figure 29: This graph represents the change ratio (from the first half to the second half of the experiment) of total number of discoveries for all participants.

Table 33: This contingency table represents the relationship between score values and the pace of discoveries throughout the experiment for group Normal. The odds ratio is 1.0569 with a P-value for the Chi-Squared Test of 0.8242. Therefore, we can conclude that in group Normal, participants' performance (i.e, rate of discoveries) remains the same throughout the experiment.

	Discoveries in First Half	Discoveries in Second Half
High Performers	114	123
Low Performers	57	65

Table 34: This contingency table represents the relationship between score values and the pace of discoveries throughout the experiment for group Latency. The odds ratio is 0.8857 with a P-value for the Chi-Squared Test of 0.7062. Therefore, we can conclude that in group Latency, participants' performance (i.e, rate of discoveries) remains the same throughout the experiment.

	Discoveries in First Half	Discoveries in Second Half
High Performers	75	88
Low Performers	51	53

4.6.1.3 Role of Latency

Table 35 represents the contingency table for the role of latency on the number of discoveries in the first and second half of the experiment. The odds ratio is 1.02 with the P-value of the Chi-Squared Test of 0.94. Therefore, we cannot reject the null hypothesis of no nonrandom association between the categorical variables. This suggests that having latency does not change the pace of discoveries throughout the experiment.

4.6.2 Interaction Time

Here, we look at the change ratio of the total amount of interaction time in each half of the experiment by each participant. The aim is to investigate whether participants interact more or less with the tool as they progress through the experiment and if this behavior is different for successful participants. Each second of a user log is analyzed to measure the interaction time. If between two consecutive seconds the position or orientation of the camera changes, or if the position of sliders (Time slider and scales) changes, or if the color distribution changes, that second is counted towards the total interaction time of that user.

Table 35: This contingency table represents the relationship between latency and the number of discoveries happening throughout the experiment. The odds ratio is 1.0179 with the P-value of the Chi-Squared Test of 0.9356. Therefore, we can conclude that having latency does not change the pace of discoveries throughout the experiment.

	Discoveries in First Half	Discoveries in Second Half
Group Normal	171	188
Group Latency	126	141

4.6.2.1 Change of Measure over Time We start our analysis by investigating each group separately. In each group, we examine the total interaction time of each participant in the first half and the second half of the experiment. We then analyze whether the distribution of these numbers for all 20 participants in each group is different between the first and second half.

In group Normal, the KS-Test of normality confirms that both distributions are normal (P-value of 0.81 for the distribution in the first half, and P-value of 0.38 for the distribution in the second half). An F-Test yields a P-value of 0.04, indicating that these distributions have different variances. The corresponding 2-Sided Unpaired Welch Corrected T-Test results in a P-value of 0.43, suggesting that in group Normal, in both the first half and the second half of the experiment, total interaction time from each participant comes from similar normal distributions. In other words, the rate of interaction in group Normal does not change during the experiment.

In group Latency, the KS-Test confirms that both distributions are normal (P-value of 0.83 for the distribution in the first half, and P-value of 0.81 for the distribution in the second half). An F-Test yields a P-value of 0.03, indicating that these distributions have different variances. The corresponding 2-Sided Unpaired Welch Corrected T-Test results in a P-value of 0.31, suggesting that in group Latency, in both the first half and the second half

of the experiment, total interaction time from each participant comes from similar normal distributions. In other words, the rate of interaction in group Latency does not change during the experiment.

4.6.2.2 Correlation with Performance Score Figure 30 represents the change ratio of the interaction time for all participants in conjunction with their scores. The R-value of the correlation for the combined group (both group Normal and Latency) is 0.21 (with a P-value of 0.19). The R-value for group Normal is 0.37 (with a P-value of 0.11), and for group Latency is 0.05 (with a P-value of 0.85). These results indicate that there is no significant correlation between a participant's score and the change ratio of the total interaction time they make in the first and second half of the experiment. That being said, there seems to be a very weak correlation between the interaction time and score for participants of group Normal, suggesting that successful participants in comparison to unsuccessful ones in group Normal, interact more with the tool in the second half of the experiment.

We can also combine participants into two groups of high performers and low performers and then measure if they interact differently in each half of the experiment. Table 36 represents the contingency table for the total interaction time in each half by high performers and low performers in group Normal. The P-value for the CH-Test is 0.0210. Therefore, we can reject the null hypothesis of no nonrandom association between the categorical variables and can conclude that in group Normal, high performers interact consistently throughout the experiment, while low performers interact less with the tool as they progress through the experiment. A similar contingency table for group Latency is presented in Table 37. The P-value for the CH-Test is 0.6163. Therefore, we cannot reject the null hypothesis of no nonrandom association between the categorical variables and can conclude that in group Latency, a participant's performance level does not indicate how much her interaction time would change throughout the experiment.

4.6.2.3 Role of Latency Table 38 represents the contingency table for the role of latency on the interaction time in the first and second half of the experiment. The odds ratio is 0.9625 with the P-value of 0.4944 for the Chi-Squared Test. Therefore, we cannot reject the null

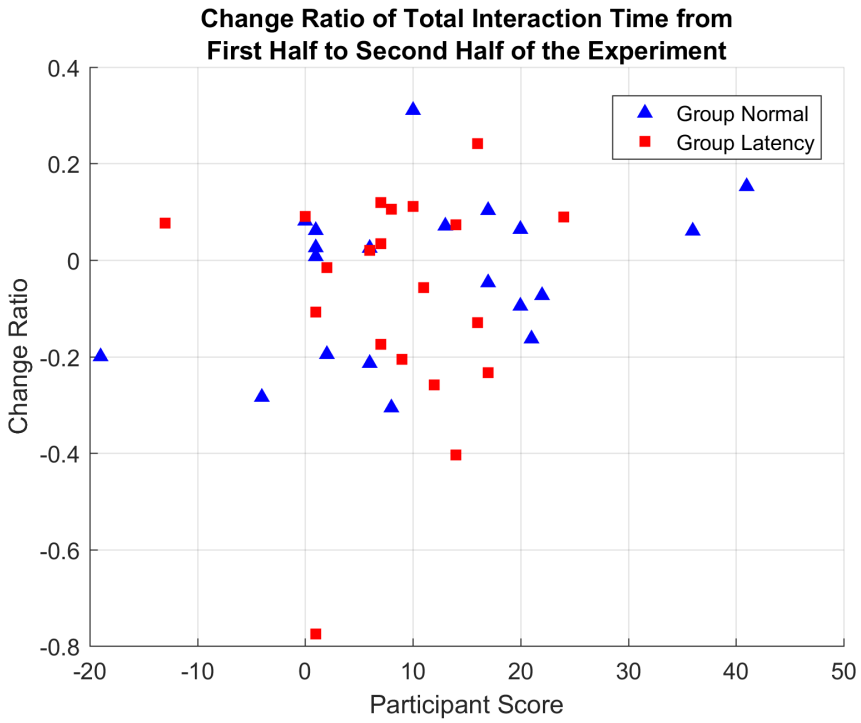


Figure 30: This graph represents the change ratio (from the first half to the second half of the experiment) of total interaction time for all participants.

Table 36: This contingency table represents the relationship between score and change in average interaction time for group Normal. The odds ratio is 0.8827 with a confidence interval of $[0.7945, 0.9806]$ and with a P-value for the Chi-Square Test of 0.0210. Therefore, we can reject the null hypothesis which means that in group Normal, those with low scores, on average interact less with the tool as the experiment progresses.

	Interaction Time in First Half	Interaction Time in Second Half
High Performers	1419	1456
Low Performers	1410	1277

Table 37: This contingency table represents the relationship between score and change in average interaction time for group Latency. The odds ratio is 1.0286 with a P-value for the Chi-Squared Test of 0.6163. Therefore, we cannot reject the null hypothesis which means that in group Latency, the average interaction time with the tool does not change during the experiment based on the performance level of the participants.

	Interaction Time in First Half	Interaction Time in Second Half
High Performers	1367	1253
Low Performers	1311	1236

hypothesis of no nonrandom association between the categorical variables. This suggests that having latency does not affect the change in the amount of interaction time in each half of the experiment.

4.6.3 Coherence of Discoveries

In this measure, we evaluate the change ratio of the similarity of consecutive discoveries in each half of the experiment. Sometimes, when a participant makes a discovery, she uses that discovery as a basis for the next discovery. For example, when a user finds a wealthy neighborhood, she may try to figure out what is the dominant profession of that neighborhood. Such connected discoveries build a storyline comprised of coherent facts. To measure such coherent discoveries, we have assumed that they should share most of their dimensions. Therefore, for each participant and each half of her experiment, we calculate the average jump distance (i.e., number of dimensions that differ from one discovery to the next) between their consecutive discoveries. We then evaluate the change ratio of this measure to find out if participants tend to generate more or less coherent discoveries by the end of their experiment, and if this behavior is different for successful participants.

Table 38: This contingency table represents the relationship between latency and the change in the average interaction time. The odds ratio is 0.9625 with a P-value for the Chi-Square Test of 0.4944. Therefore, we can conclude that having latency does not affect the change in the amount of interaction time throughout the experiment.

	Interaction Time in First Half	Interaction Time in Second Half
Group Normal	1414	1366
Group Latency	1339	1245

4.6.3.1 Change of Measure over Time First, we assess each group separately. In each group, we examine the average jump distance of each participant in each half of the experiment. We then analyze whether the distribution of these numbers for all 20 participants in each group is different between the first and second half.

In group Normal, the KS-Test of normality confirms that both distributions are normal (P-value of 0.60 for the distribution in the first half, and P-value of 0.36 for the distribution in the second half). An F-Test yields a P-value of 0.18, indicating that these distributions have similar variances. The corresponding Unpaired T-Test results in a P-value of 0.93, suggesting that in group Normal, in both the first half and the second half of the experiment, average jump distances between consecutive discoveries come from similar normal distributions. In other words, the similarity between consecutive discoveries in group Normal remains the same during the experiment.

In group Latency, the KS-Test of normality confirms that both distributions are normal (P-value of 0.68 for the distribution in the first half, and P-value of 0.24 for the distribution in the second half). An F-Test yields a P-value of 0.45, indicating that these distributions have similar variances. The corresponding Unpaired T-Test results in a P-value of 0.93, suggesting that in group Latency, in both the first half and the second half of the experiment, average jump distances between consecutive discoveries come from similar normal distributions. In

other words, the similarity between consecutive discoveries in group Latency remains the same during the experiment.

4.6.3.2 Correlation with Performance Score Figure 31 represents the change ratio of the average jump distance for all participants in conjunction with their scores. The R-value of the correlation for the combined group (Normal and Latency) is 0.06 (with a P-value of 0.70). The R-value for group Normal is 0.08 (with a P-value of 0.74) and for group Latency is 0.05 (with a P-value of 0.84). These results indicate that there is no significant correlation between a participant's score and the change ratio of their average jump distant in the first and second half of the experiment.

4.6.4 Complexity of Discoveries

This measure evaluates the change ratio of the average number of dimensions selected for each discovery, in each half of the experiment. The aim is to assess if participants tend to create more or less complex discoveries (i.e., discoveries with a higher number of dimensions selected) as they progress through the experiment, and in particular if this behavior is different for successful participants.

4.6.4.1 Change of Measure over Time We start our analysis by looking at each group separately. In each group and for each participant, we measure the average number of dimensions per discovery in each half of the experiment. We then analyze whether the distribution of these numbers for all 20 participants in each group is different between the first and second half.

In group Normal, the KS-Test of normality confirms that both distributions are normal (P-value of 0.92 for the distribution in the first half, and P-value of 0.49 for the distribution in the second half). An F-Test yields a P-value of 0.60, indicating that these distributions have similar variances. The corresponding Unpaired T-Test results in a P-value of 0.57, suggesting that in group Normal, in both the first half and the second half of the experiment, the average values of the number of dimensions selected per discovery come from similar normal

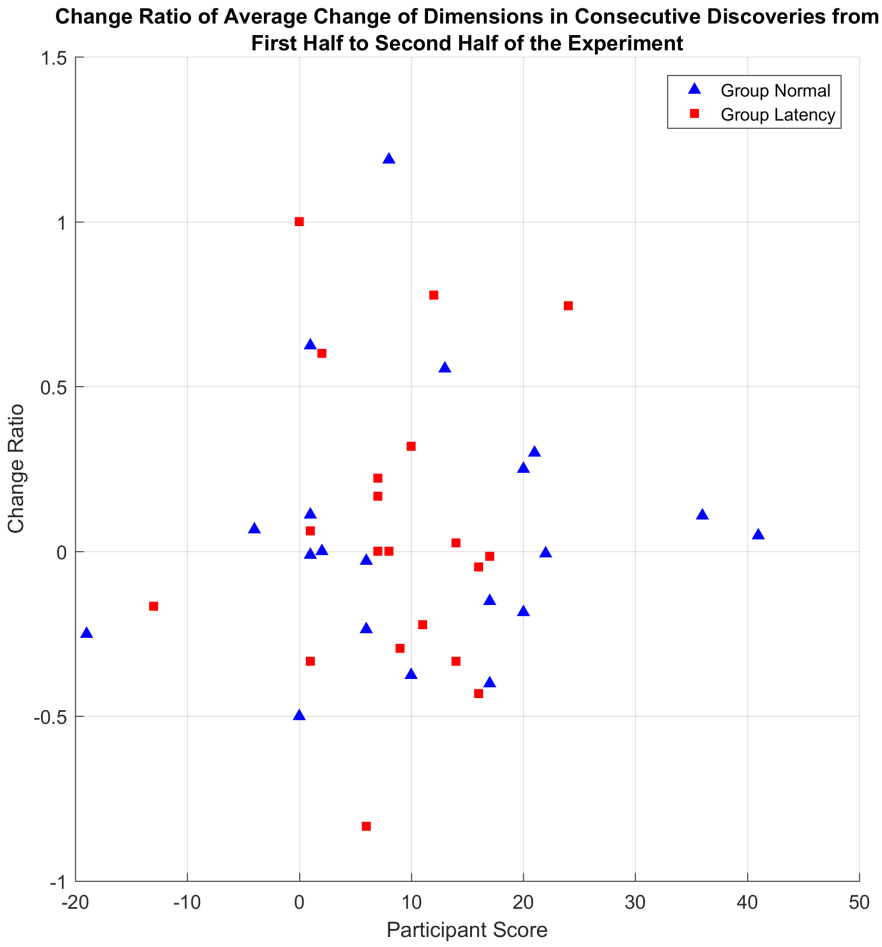


Figure 31: This graph represents the change ratio (from the first half to the second half of the experiment) in the average number of dimensions changed during consecutive discoveries for all participants.

distributions. In other words, the complexity of discoveries in group Normal remains the same throughout the experiment.

In group Latency, the KS-Test of normality confirms that both distributions are normal (P-value of 0.67 for the distribution in the first half, and P-value of 0.63 for the distribution in the second half). An F-Test yields a P-value of 0.12, indicating that these distributions have similar variances. The corresponding Unpaired T-Test results in a P-value of 0.46, suggesting that in group Latency, in both the first half and the second half of the experiment, the average values of the number of dimensions selected per discovery come from similar normal distributions. In other words, the complexity of discoveries in group Latency remains the same throughout the experiment.

4.6.4.2 Correlation with Performance Score Figure 32 represents the change ratio of the average number of dimensions per discovery for all participants in conjunction with their scores. The R-value of the correlation for the combined group (Normal and Latency) is 0.07 (with a P-value of 0.68). The R-value for group Normal is 0.14 (with a P-value of 0.57) and for group Latency is 0.05 (with a P-value of 0.84). These results indicate that there is no significant correlation between a participant's score and the change ratio of the average complexity of their discoveries in the first and second half of the experiment.

4.6.5 Abstraction Level of Discoveries

This measure evaluates the change ratio of the percentage of discoveries with a map layer, in each half of the experiment. The aim is to assess if participants tend to create higher or fewer number of abstract discoveries as they progress through the experiment, where abstract discoveries are the ones without a map layer. Also, we will examine whether successful participants generate map-based discoveries in a consistent manner or if they tend to generate higher or fewer number of them as they get near the end of their session.

4.6.5.1 Change of Measure over Time First, we look at each group separately where for each participant, we measure the percentage of discoveries with a map layer in each half

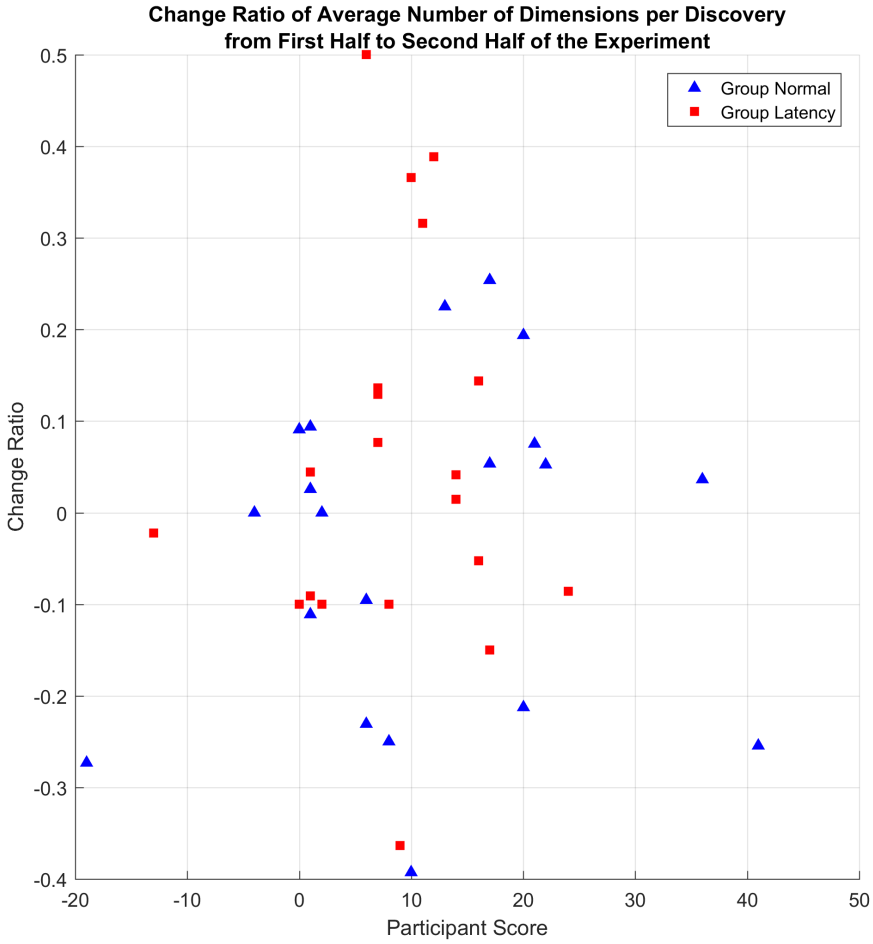


Figure 32: This graph represents the change ratio (from the first half to the second half of the experiment) in the average number of dimensions selected per discovery for all participants.

of the experiment. We then analyze whether the distribution of these numbers for all 20 participants in each group is different between the first and second half.

In group Normal, the KS-Test of normality confirms that both distributions are normal (P-value of 0.73 for the distribution in the first half, and P-value of 0.59 for the distribution in the second half). An F-Test yields a P-value of 0.28, indicating that these distributions have similar variances. The corresponding Unpaired T-Test results in a P-value of 0.17, suggesting that in group Normal, in both the first half and the second half of the experiment, the percentage of discoveries with a map layer comes from similar normal distributions. In other words, the abstraction level of discoveries in group Normal remains the same throughout the experiment. That being said, the Right-Sided T-Test results in a weak statistically significant result with a P-value of 0.09, suggesting that participants in group Normal tend to generate less map-based discoveries as they progress through the experiment.

In group Latency, the KS-Test of normality confirms that both distributions are normal (P-value of 0.33 for the distribution in the first half, and P-value of 0.56 for the distribution in the second half). An F-Test yields a P-value of 0.36, indicating that these distributions have similar variances. The corresponding Unpaired T-Test results in a P-value of 0.39, suggesting that in group Latency, in both the first half and the second half of the experiment, the percentage of discoveries with a map layer comes from similar normal distributions. In other words, the abstraction level of discoveries in group Latency remains the same throughout the experiment.

4.6.5.2 Correlation with Performance Score Figure 33 represents the change ratio of the percentage of map-based discoveries for all participants in conjunction with their scores. The R-value of the correlation for the combined group (Normal and Latency) is 0.05 (with a P-value of 0.75). The R-value for group Normal is -0.13 (with a P-value of 0.59) and for group Latency is 0.34 (with a P-value 0.15). These results indicate that there is no significant correlation between a participant's score and the change ratio of the percentage of their map-based discoveries in the first and second half of the experiment. That being said, it may be possible to assume a weak positive correlation between the score and the change ratio of map-based discoveries for group Latency. This may suggest that in group

Latency, successful participants tend to generate more map-views and therefore more map-based discoveries as they progress in the session.

4.6.6 Diversity of Discoveries

This measure evaluates the change ratio of the percentage of unique discoveries, in each half of the experiment. The aim is to assess if participants tend to create higher or fewer number of unique discoveries as they progress through the experiment, where unique discoveries are the ones that only one participant in the group has generated. To compare discoveries, we only look at the similarity of dimensions selected while the participant made that discovery. Also, we will examine whether successful participants generate unique discoveries with a constant rate or if they generate more (or less) unique discoveries as they progress through the experiment.

4.6.6.1 Change of Measure over Time First, we look at each group separately where for each participant, we measure the percentage of unique discoveries in each half of the experiment. We then analyze whether the distribution of these numbers for all 20 participants in each group is different between the first and second half.

In group Normal, the KS-Test of normality confirms that both distributions are normal (P-value of 0.84 for the distribution in the first half, and P-value of 0.79 for the distribution in the second half). An F-Test yields a P-value of 0.12, indicating that these distributions have similar variances. The corresponding Unpaired T-Test results in a P-value of 0.53, suggesting that in group Normal, in both the first half and the second half of the experiment, the percentage of unique discoveries comes from similar normal distributions. In other words, the uniqueness of discoveries in group Normal remains the same throughout the experiment.

In group Latency, the KS-Test of normality confirms that both distributions are normal (P-value of 0.77 for the distribution in the first half, and P-value of 0.89 for the distribution in the second half). An F-Test yields a P-value of 0.03, indicating that these distributions do not have similar variances. The corresponding Unpaired Welch-Corrected T-Test results in a P-value of 0.77, suggesting that the percentage of unique discoveries in group Latency

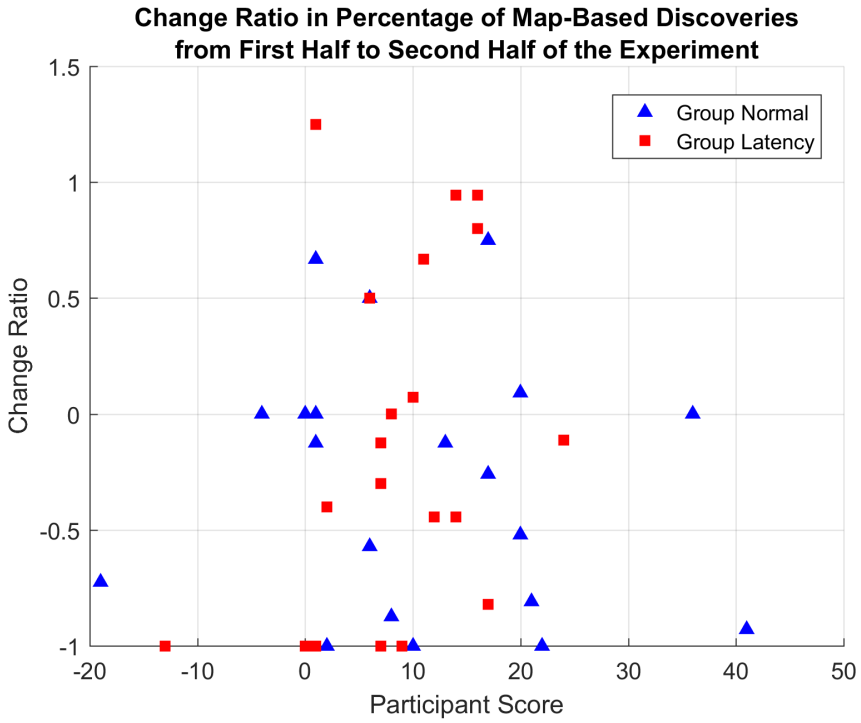


Figure 33: This graph represents the change ratio (from the first half to the second half of the experiment) of the percentage of map-based discoveries for all participants.

remains the same throughout the experiment.

4.6.6.2 Correlation with Performance Score Figure 34 represents the change ratio of the percentage of unique discoveries for all participants in conjunction with their scores. The R-value of the correlation for the combined group (Normal and Latency) is 0.23 (with a P-value of 0.15). The R-value for group Normal is 0.27 (with a P-value of 0.24) and for group Latency is 0.07 (with a P-value of 0.77). These results indicate that there is no significant correlation between a participant's score and the change ratio of the percentage of unique discoveries in the first and second half of the experiment.

4.7 ANALYZING STRATEGIES: QUALITATIVE APPROACH

This section presents a qualitative approach to analyzing participants. Our focus is on answering why some participants were highly successful in utilizing the tool while others were not. In particular, we will investigate what strategies work and which pitfalls should be avoided. The insights we learn can then help us in designing better HDI systems. These design guidelines are discussed in Chapter 5.

Using discovery markings, we can sort all participants based on their score, i.e., the total number of correct discoveries they made minus the incorrect ones. We then select four extreme cases: the participants with the highest and lowest scores from each group. To look for differentiating patterns, we thoroughly assess the data collected for these selected participants. This data includes their user interaction logs, the discovery markings derived from their recorded videos, their pre and post-test questionnaires, and also notes taken by the experimenter during their sessions.

To better understand how each participant explored the data, the most important aspects of their work is presented in an *exploration timeline*. A sample exploration timeline is shown in Figure 35. In this graph, the horizontal axis represents time (in minutes, from 0 to 60 for the one-hour duration of the experiment), and the vertical axis represents dimensions of data. Rectangles on the graph represent instances where the participant assigns the

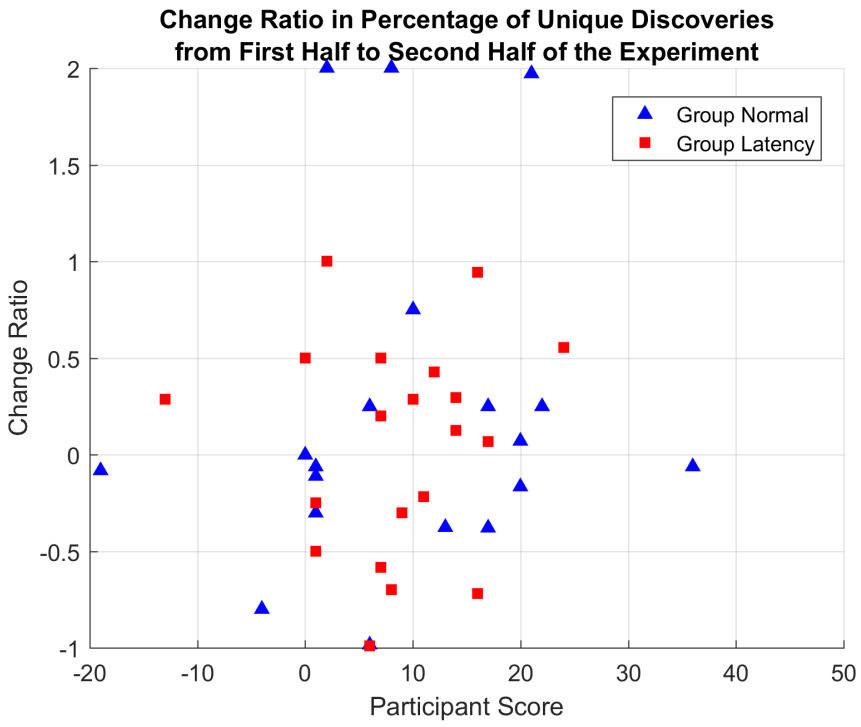


Figure 34: This graph represents the change ratio (from the first half to the second half of the experiment) of the percentage of unique discoveries for all participants.

corresponding data dimension to a visual dimension. The color of rectangles indicates which visual dimension was used for that assignment. The length of rectangles represents the duration for which this assignment was valid. Correct discovery events are represented by solid vertical blue lines, and incorrect discoveries are shown by dashed vertical red lines. Overall, an exploration timeline is a detailed visual representation of a participant's activity and provides a framework for analyzing their strategies.

4.7.1 Lowest Score Participant, Group Latency

The exploration timeline of this participant is shown in Figure 35.

He starts his exploration by building an abstract (i.e., non-map) 5-dimensional view. He has a hard time interpreting that and eventually makes an incorrect discovery. From then on, almost all of his discoveries (which are mostly incorrect) are on complex views with at least 4 dimensions selected. The only correct discovery happens on minute 42, on a 3-dimensional view in which one of the dimensions is Time, essentially making it a 2-dimensional visualization.

There are extended periods of time in which he interacts with a single view without reaching any results (e.g., minutes 51–58). This behavior is in contrast to more successful participants. For example, when the highest score participant of group Latency, shown in Figure 37, confronts a complex abstract 3-dimensional view in minutes 35–36, he spends only two minutes on that view and after not making any progress on it, he abandons it in favor of a different view where he is then more productive at.

In general, the lowest score participant of group Latency has a hard time understanding how the tool works. Many times, what he mentions out loud as his objective is not related to the visualization he creates. This mismatch between what he wants to do and what he actually does persists throughout the experiment. For example, in minute 9, he assigned Longitude to one of the dimensions to create a geospatial view, but he does not assign Latitude as well. This results in an incomplete map view which eventually does not help him in making any correct discoveries. In another instance, he uses the color legend to interpret a view, yet he never assigns any data dimensions to Color, making the interpretations

meaningless.

Many of his incorrect discoveries are due to the *decoupling issue*. The decoupling issue is when a participant incorrectly assumes that various visual dimensions affect each other. For example, if she assigns a data dimension to the X axis, and then another data dimension to the Y axis, then she incorrectly thinks the assignment on the Y axis affects the position of the points on the X axis. Many times participants thought that if they assign a data dimension to an axis, they are filtering for that value. For example, they assumed if they assign the *percentage of jobs with income more than $3K per month* to the X axis, what they see is only neighborhoods with high income. This is in contrast to the way a scatterplot works, which in this case represents all the neighborhoods, but only allocates their position based on the percentage of high-income jobs. The decoupling issue was observed among many participants which may potentially indicate that scatterplots can be easily misinterpreted, even by graduate students who have had training in interpreting such plots.

Another important observation regarding this participant is that he never formed any *discovery patterns*. Discovery patterns are templates that many participants formed to generate discoveries in a similar fashion. For example, a participant may learn how to interpret map views over time. She then assigns various job sectors to the Color dimension and in each case discovers how a certain job category has changed over the years. Discovery patterns provide a repeatable procedure for finding facts and are quite effective. Nevertheless, this participant never formed any such discovery patterns. His timeline is a series of trials and errors in the hope of finding something meaningful in the data without any systematic approach to exploring it. The only time he engages in a discovery pattern is towards the end of the session, in minutes 45–50, where he applies a single pattern to four different topics. In this pattern, he is keeping the X, Y and Time dimensions the same, while assigning different data dimensions to Color. Unfortunately, he makes incorrect interpretations in those instances.

4.7.2 Lowest Score Participant, Group Normal

Exploration timeline of this participant is shown in Figure 36.

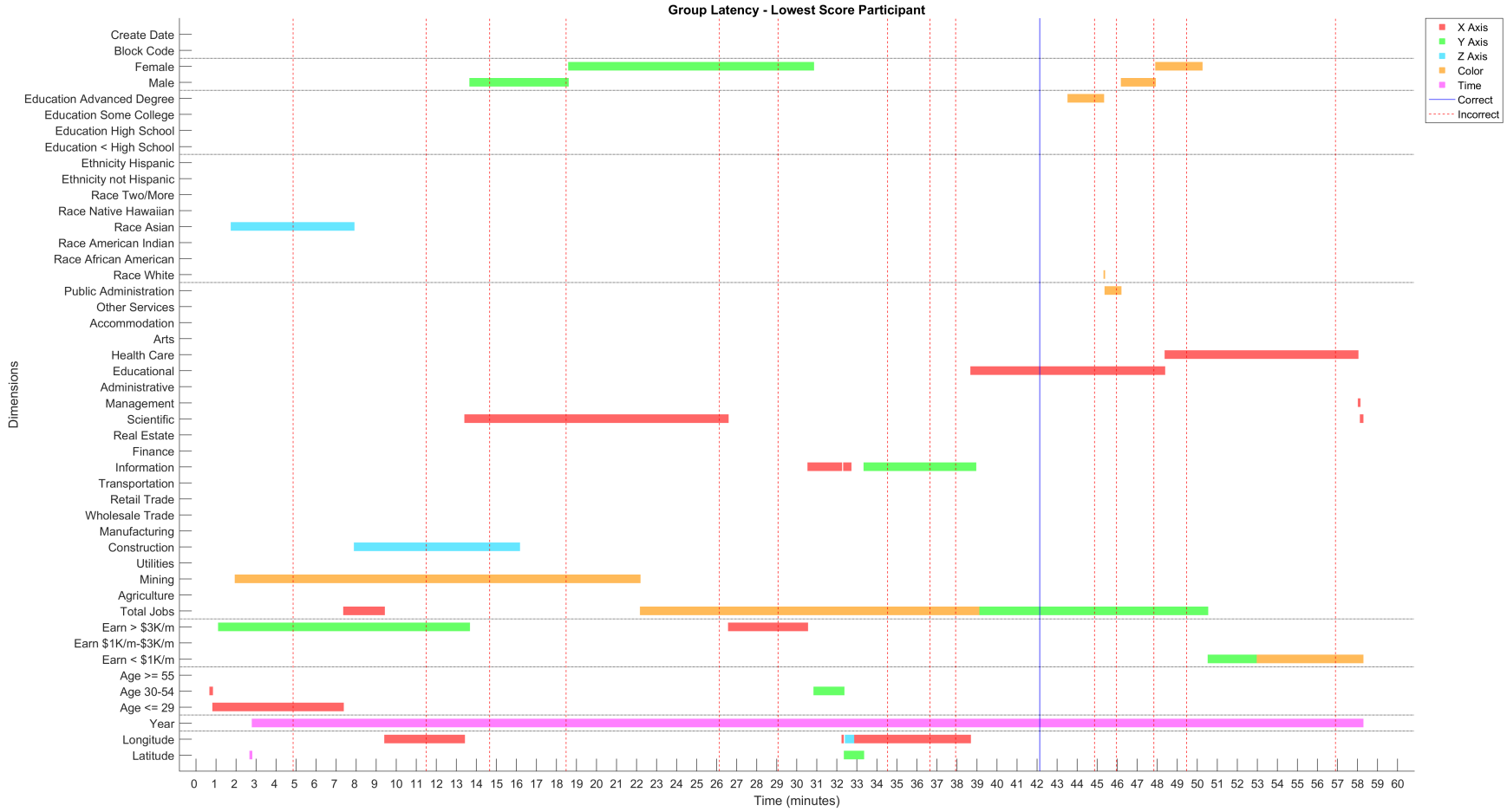


Figure 35: Exploration timeline for the lowest score participant from group Latency.

This participant takes a more methodical approach for exploring the data. He first starts with a 2-dimensional view. He then gradually adds to the complexity and works with 2 or 3-dimensional views for a while. Only when he is working with map-based views, (minutes 20–36), he builds complex 4-dimensional views. Nevertheless, despite his methodical approach and gradual addition of complexity, he has strong issues with decoupling visual dimensions. He incorrectly perceives each visual dimension as a filter that narrows down the exploration space, instead of the correct interpretation in which each visual dimension is a projection line for the entirety of the data. Because of this problem, most of his interpretations are incorrect. It is interesting that he sometimes perceives dimensions as separate entities and sometimes as filters. For example, in one particular view, he interprets data dimensions assigned to X and Y axes as separate dimensions, each showing the percentage of jobs in its corresponding category, but then he uses the Z axis to filter the data. It is worth noting that he never uses the Color dimension for any of his views. Also, in non-map views, he rarely uses the Z dimension as he mentions he finds it confusing.

He is successful in building various discovery patterns. For example, during minutes 20–36 where he is looking at the map view, he quickly learns how to interpret the visualization and then applies that to several different discoveries. Unfortunately, due to the decoupling issue mentioned before, most of those discoveries are incorrect.

He also has problems with interpreting missing data. Although during the training session all participants are explicitly told about the existence of missing data dimensions and work with an example containing missing data, he forgets this fact and is surprised by portions of the data that have missing values. For example, the race and ethnicity attributes are only collected in the last three years of the LEHD dataset. However, he thinks the sudden jump in the population of Asians in those years is due to a large migration of people with Asian origins to Pennsylvania.

In the questionnaire, there is a question on the effectiveness of active participation in exploring a dataset. In particular, participants are asked whether they think watching a video on income inequality would yield a better outcome than actively plotting a visualization on the same topic. This question is asked both before and after the experiment. Interestingly, this participant had different views before and after the experiment on this question. In

his pre-test questionnaire, he says *"I believe watching a video would be more effective than actively plotting a dataset."* But in the post-test questionnaire he says *"I believe more than watching a video, actively plotting a visualization of the same topic would be more beneficial. The data interpretation is more interesting when we start plotting."* This probably shows that even though he formed an incorrect mental model of how the tool works, and although he had issues in working with higher-dimensional views, he nevertheless enjoyed the process and concluded that active participation was showing him important facts that he would have missed if he was just presented with the final discoveries.

4.7.3 Highest Score Participant, Group Latency

The exploration timeline of this participant is show in Figure 37.

He starts the exploration by looking at a map-base view over time. Having a map layer and looking at changes over time makes it easy for him to make his first discovery. He soon repeats that process on multiple similar views, all leading to successful discoveries. On minute 11, he starts building 5-dimensional views, all based on map and time. He makes several other correct discoveries in this mode, although this time his discovery rate is lower. He shows he can find a discovery pattern and then repeatedly use it to find correct facts based on those patterns.

On minutes 26–28, he builds his first non-map view. This is a simple 2-dimensional view. He makes a correct discovery here but then soon returns to the previous map and time-based visualization space. It seems he is more comfortable with map-base discoveries. Interestingly, he gives non-map space another try on minutes 35–36, this time by creating a 3-dimensional view. After spending around 2 minutes on this view, he does not find any discoveries and he removes one of the dimensions to make it simpler. This time, he makes a correct discovery.

It seems that he is willing to try new discovery patterns, but as soon as they become too complicated, he stops exploring them and returns to a simpler discovery pattern. He also explores the data very systematically. He starts with simpler views and only after he fully understands that view (e.g., by making several correct discoveries in that specific projection

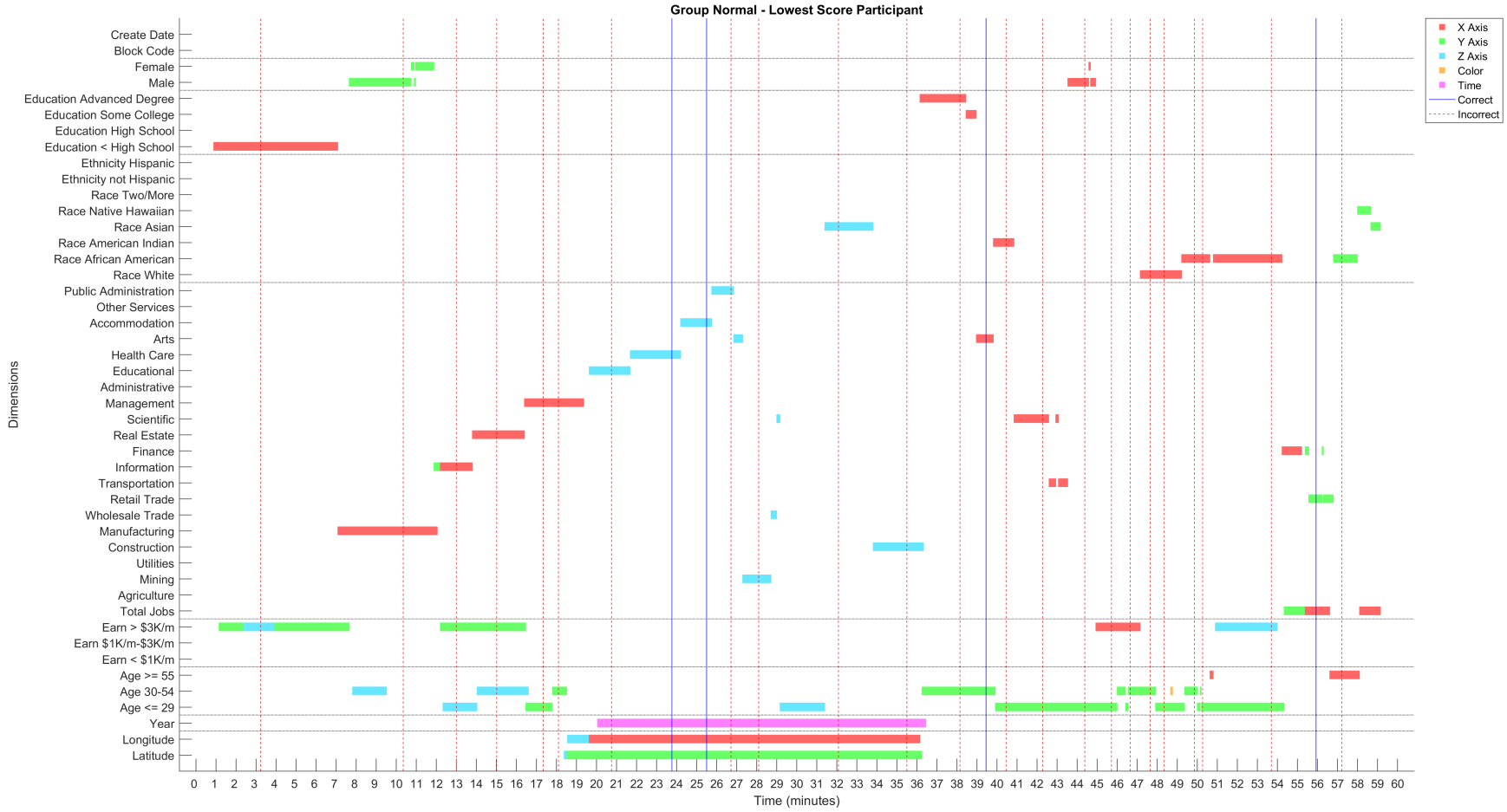


Figure 36: Exploration timeline for the lowest score participant from group Normal.

of the data) he adds more complexity to it.

It is also worth noting that he uses time dimension very well to filter missing data. He has no issues with decoupling various dimensions of a visualization. In his questionnaire, he mentions he has seen EVA before, probably at a presentation, but he has never used it. He also mentions that exploring the data instead of just watching a video helps him better concentrate: *"active plotting guarantees I pay attention."*

4.7.4 Highest Score Participant, Group Normal

The exploration timeline of this participant is shown in Figure 38.

He starts exploring the data and understanding the tool very methodically. He first starts by building a 2-dimensional view. After making a correct discovery there, he adds another dimension to the same view, making it 3-dimensional. Again, after he makes a correct discovery there, he creates a more complex 4-dimensional view by keeping the previous view and adding one more dimension to it. He then makes another correct discovery on that view. He then builds a map-based view and assigns one of the data dimensions to Color. After making a discovery there, he keeps the same view but adds another dimension to it by assigning another data dimension, this time to the Z axis. He then makes another discovery, which is built on top of the previous one. He repeats this pattern in minutes 10–11 as well. In minutes 14–15, he goes outside of the map view but keeps one of the dimensions similar to his previous discovery. He then makes another correct discovery in a 2-dimensional abstract view. He then keeps the dimensions the same, but adds another axis to it, making a 3-dimensional view which again leads to a correct discovery.

This pattern of related discoveries happens frequently. He often looks at various views, finding different facts in the data until he decides to investigate one of those facts in more depth. He then looks at that dimension from several different views, making related discoveries all looking at the same concept but from different perspectives. For example, in minutes 18–22 he looks at various facts all based on a map view. At minute 23, one of those facts becomes interesting to him, and he then starts exploring that fact from different perspectives. For example, he looks at the total number of jobs in the retail sector. He then

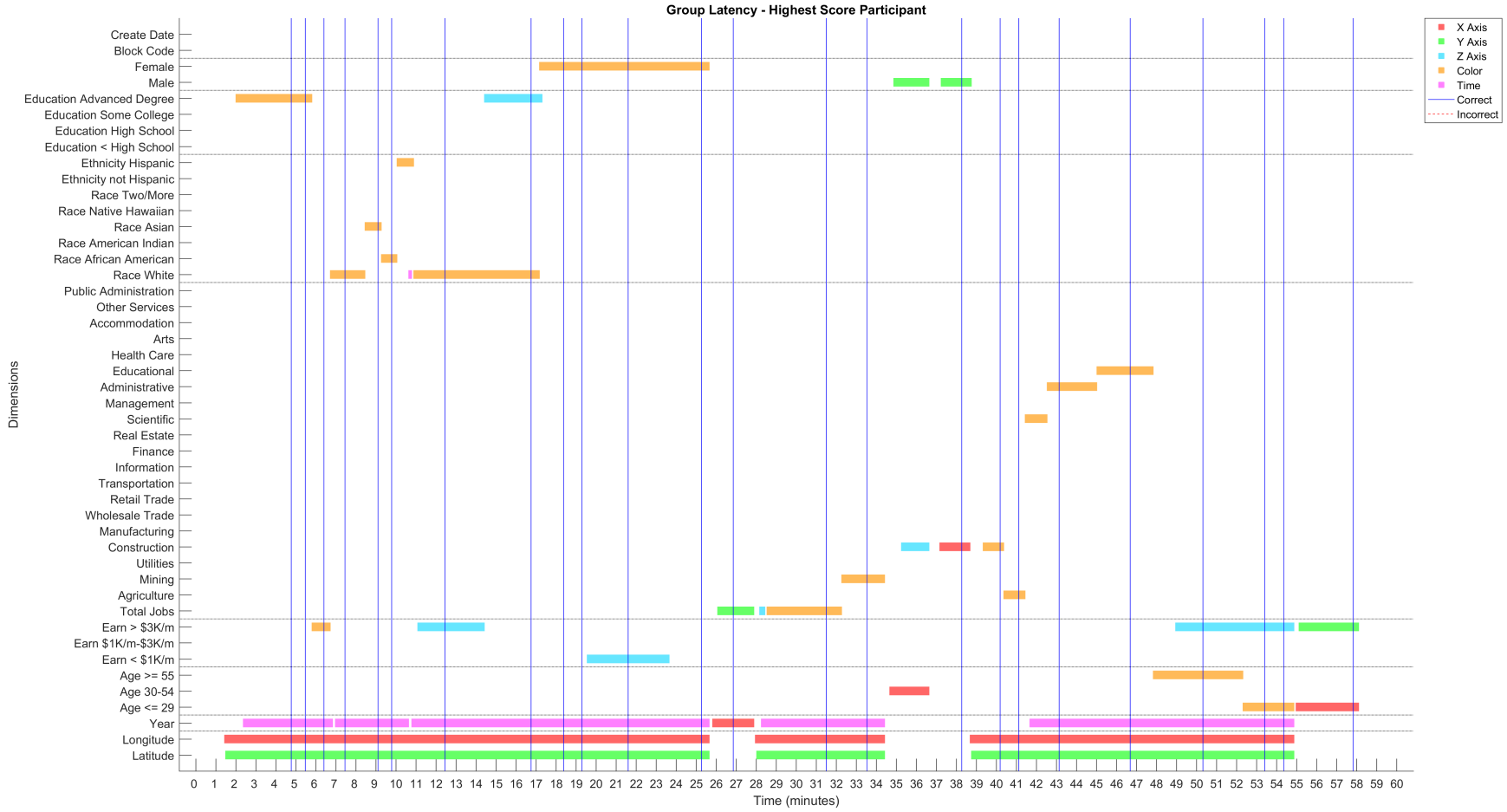


Figure 37: Exploration timeline for the highest score participant from group Latency.

investigates how it correlates with the number of males in a neighborhood. He then explores whether education has any correlation with those jobs and later on he looks at income and how it correlates with jobs in the retail sector. Overall, from minute 25 to 33, he is looking at a series of interconnected discoveries, all looking at different aspects of the same story.

This pattern of storytelling (i.e., building a series of interconnected discoveries, all looking at different aspects of the same shared subject) is repeated multiple times, e.g., in minutes 33–34, 35–37, 38–47, 47–60. In each episode, he makes several connected discoveries. Some of the views are abstract (often 3-dimensional), and some are map-based, showing his ability to slice and dice the data in a wide variety of ways, all to help him get a deeper understanding of some topic by discovering various facts about it. The overall result is a detailed and multidimensional story on his topic of interest.

Another observation is that he tries to reach the same conclusion from different approaches. This means that he sometimes repeats a discovery several times. This helps him to make sure he understands the tool correctly, and he is correctly mapping the questions in his mind to the visual space and vice versa. He is also capable in double-checking his discoveries. For example, once he reached an incorrect conclusion, but then upon further examination, he quickly finds his mistake and corrects it.

He is one of the few participants who has no trouble understanding complex non-map views and shows no signs of the decoupling issue. For example, in minute 32, he makes a complex view on race, gender, education level and income and interprets it correctly.

In his questionnaire, he mentions he has never seen EVA before. He also thinks both video and data exploration are valuable in their own regard: *"Exploratory data analysis can help lead to new questions and discoveries, whereas videos are best for knowledge already gained."* Also, some participants ran out of ideas for exploring the data around the end of their experiment, but he had still many more questions to ask: *"I'd like to look more at the interactions between race and types of jobs (manufacturing, healthcare, etc.) I thought about this only at the end."*

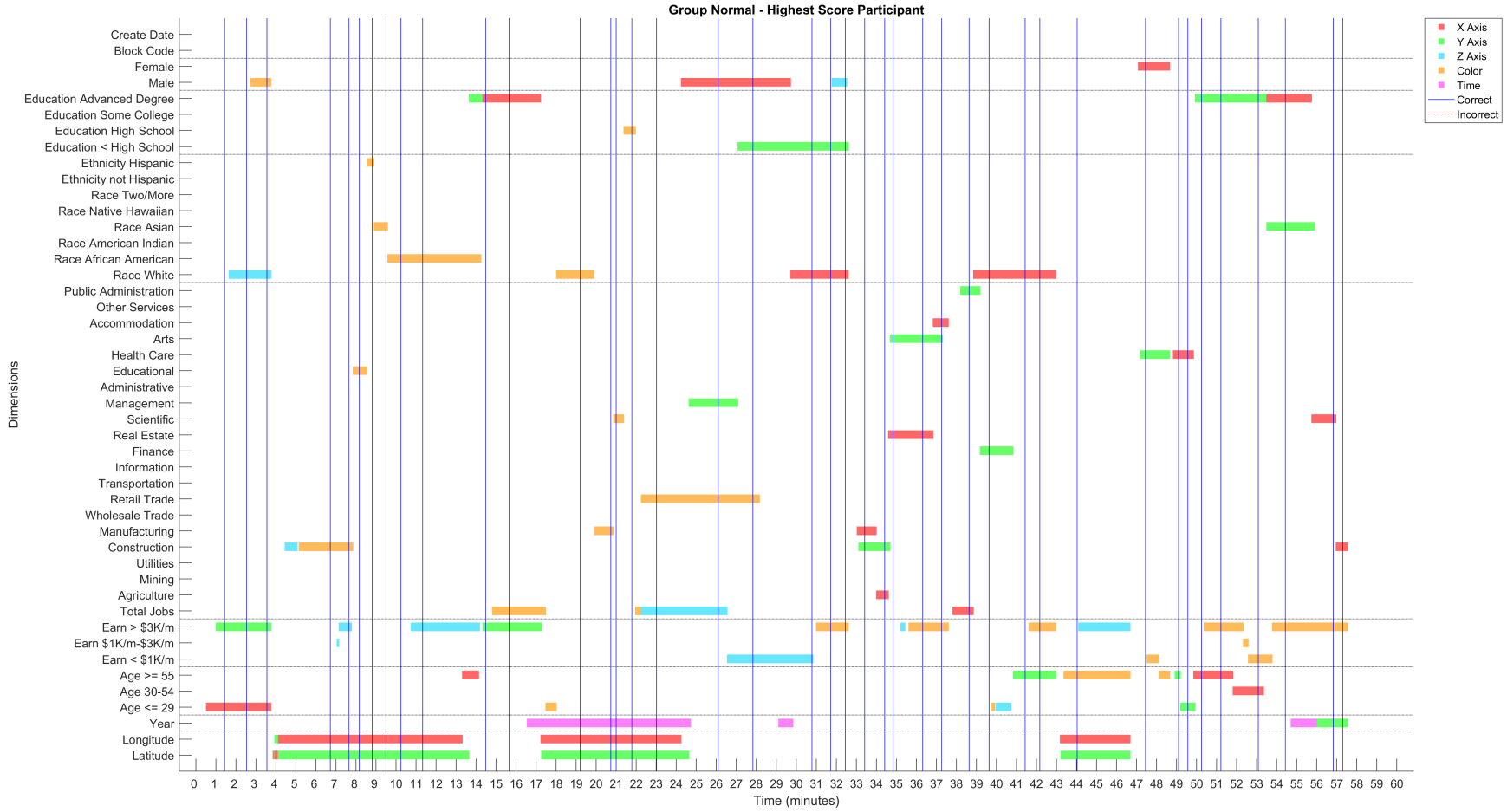


Figure 38: Exploration timeline for the highest score participant from group Normal.

4.8 SUMMARY OF FINDINGS

The goal of our analysis is to answer two main questions, (1) whether incremental latency decreases the quantity and quality of knowledge discovery in EVA, and (2) if there are distinct observable differences between the high performer and low performer participants.

Our findings indicate that incremental latency results in a decrease in the quantity of discoveries, but the quality of discoveries remains intact. In particular, for every view a user generates, the chance of making a discovery is not statistically different between the two groups. Furthermore, for each discovery a user makes, the chance of that discovery being correct is also similar between the two groups. Also, both groups generate 60% unique views, indicating that incremental latency does not lead to a decrease in exploring new ideas. The only differences between the Latency and Normal group are related to the depth of discoveries where participants of group Latency benefit more from the map layer (i.e., they have more correct discoveries in those views) and also tend to have more correct discoveries in moderately complex views while producing more mistakes in highly complex views.

In regards to the second question, our investigation indicates that successful participants utilize strategies that lead to formation of a correct mental model of the tool. This mental model defines how a participant can translate the question in her mind into a visual configuration inside EVA and then how can she translate a visualization back to the intended answer. High performers examine the tool methodically, starting from simple and familiar concepts, and then gradually adding to the complexity of the views or moving to more abstract subsets of the data. Furthermore, successful participants test their mental model repeatedly to make sure their understanding of the tool and data is indeed matching their contextual knowledge and hence they can rely on their interpretations on their new discoveries. Additionally, high performers use their mental model to build discovery patterns. These patterns act as a template which can be used on several projections of the data, each leading to different discoveries. Therefore, discovery patterns provide a blueprint on how a participant can start from a novel view, explore it in a particular way, and look for specific patterns that can then lead to discoveries.

These results are discussed in more details in Chapter 5.

5.0 DISCUSSION

This thesis was shaped by the research questions pursued in the emerging field of Human-Data Interaction which strives to understand how people make sense of data. Consequently, one of the initial contributions of this thesis was the development of EVA, a visual analytics tool for exploring large and high-dimensional datasets. EVA acted both as a prototype for data-driven knowledge discovery and also as a platform for experimenting and developing next-generation HDI systems. During our preliminary case studies, two questions became prominent which eventually led to the other main contributions of this thesis. The first objective was to understand how an HDI system can scale well with big data while remaining effective as a knowledge discovery tool. This question eventually boiled down to the investigation of incremental latency and its effects on the quality and quantity of discoveries. In Section 5.1, we summarize our findings and provide suggestions for a scalable HDI system architecture. Additionally, our user studies provided us with an opportunity to investigate how the strategies utilized by some participants resulted in a superior performance. Consequently, as our second objective, we propose design guidelines for big data HDI systems which aim to improve the performance of all users and facilitate knowledge discovery. These guidelines are discussed in Section 5.2.

5.1 EFFECTS OF LATENCY ON KNOWLEDGE DISCOVERY

Results presented in Chapter 4 suggest that although without latency participants generated more views (1.4 times more in our experiments), however, in both groups half of those views led to a discovery. Additionally, in both groups, 80% of those discoveries were correct.

These findings indicate that incremental latency (as described in Section 2.5.3) leads to a slower rate of discovery generation, however, it does not change one's chance of making a discovery per effort (i.e., generating a view), and does not change one's chance of deducing a correct fact per discovery. Moreover, incremental latency does not change one's chance of making a unique discovery as in both groups, 60% of discoveries were unique. Although participants were more likely to make an incorrect discovery in a unique view (2 times more in group Normal and 8.5 times more in group Latency), however, a direct comparison of the two groups showed no statistically significant difference, indicating that uniqueness leads to more mistakes, but latency does not change this rate.

Despite the similarities between the performance of participants in both groups, a deeper analysis showed some differences in the types of discoveries each group excelled at. For example, half of the views in each group had a map layer (a contextual helper). Participants in group Normal were 1.8 times more likely to make a correct discovery in a map view while group Latency participants were 6.4 times more likely to do so, suggesting that overall, having a map layer decreased participants' chance of making a mistake. Nevertheless, a direct comparison between the groups showed that having a map layer in group Latency resulted in a 2.5 times higher chance of making a correct discovery than in group Normal. These findings suggest that although having a map layer increases one's chance of making a discovery, such contextual layers play a more prominent role in the presence of latency.

Additionally, the complexity of views and discoveries were different between the groups. Without latency, there was no correlation between the number of dimensions assigned to a view and the chance of making a discovery, or the chance of that discovery being correct. On the other hand, with latency, participants had a higher chance of making a correct discovery when the view was higher dimensional. Interestingly, if the view became too complicated, the chance of making a mistake increased as well. A direct comparison between the groups showed that incorrect discoveries in group Latency had a higher number of dimensions than incorrect discoveries in group Normal. Overall, latency pushed participants toward generating more complex discoveries, but it also made those discoveries more error prone.

While any scalable HDI system should ultimately deal with latency in some form, our findings suggest that where we allow the latency to happen has a profound effect on the

resulting knowledge discovery process. For example, Liu and Heer [49] demonstrate that even introducing a half a second delay to interactive tasks (such as rotating the camera) results in a significant decrease in user activity and discovery rates. However, as it has been recently shown by Zgraggen et al. [90], immediately showing low accuracy results and then gradually improving them over time has no measurable effect on performance. Based on their experiments, participants who faced such an *incremental latency* were as productive as the ones who experienced no latency, even though the results took up to 12 seconds to fully materialize. Our research confirms these findings and extends these results in two aspects. First, we have increased the size of data and the number of visual elements. While the experiments by Zgraggen et al. rely on a $10,000$ row dataset and provide simple aggregated visualizations such as bar charts, in our case, we increased the data size to 2.8 million rows and used a point cloud visualization that represents every single row of the data as a visual element. Second, we have increased the duration of the incremental latency to a minute for a zoomed out view. Nevertheless, even with such large datasets and demanding high-resolution visualizations, and even with such long latency times, still the performance measures of "chance of discovery per view" and "chance of correctness per discovery" remained similar to the no-latency setup.

In summary, we recommend the incremental visualization approach as an effective architecture for scaling big data HDI systems. Although due to their inherent latency, the rate of discovery generation would be lower in comparison to a non-latency system, nevertheless, a user would achieve similar performance metrics based on the rate of discoveries per view and the correctness of those discoveries. Moreover, incremental systems require cheaper hardware and scale better with the size and complexity of the data. They also impose fewer assumption on the type of queries one can ask of the data, hence expanding the domain of questions addressable in the system. It is worth noting that despite these benefits, incremental systems work best when the visualization is accompanied by contextual information (such as a map layer). Also, as users may generate more mistakes on complex visualizations, incremental systems should limit the degree of complexity one can incorporate in each view to improve the accuracy of discoveries.

Additionally, we like to encourage the design and development of incremental database

systems. Most visualization systems rely on a database backend to ingest and analyze the data. However, databases are traditionally built for precise and accurate answers and hence are not suitable for the incremental and inaccurate nature of big data visual analytics. Development of a database system that answers a single query with a stream of answers, each accompanied by its confidence interval, and each answer improving the accuracy of the previous result, would facilitate the adoption of incremental visualization systems and have a meaningful impact on the design and utilization of HDI tools. BlinkDB is an early example of such a database that has been proposed by Agarwal et al. [5]. Their system implements an approximate query engine which runs queries on samples of the data and then accompanies its results with confidence intervals. It can achieve a 100 times speedup in comparison to other databases, as the authors demonstrate that it can answer queries under 2 seconds on a 17TB dataset. Although BlinkDB is not an entirely incremental database, it is a significant development in the direction of *fast and eventually accurate* data processing engines necessary for supporting scalable HDI systems.

5.2 DESIGN GUIDELINES FOR BIG DATA HDI SYSTEMS

From our quantitative analysis presented in Section 4.6, we can conclude that strategies utilized by participants do not significantly change during the experiment, nevertheless, the qualitative analysis of Section 4.7 demonstrated that the selected participants took different approaches in exploring the data and consequently achieved widely different success rates. This section provides an overview of our findings based on the four selected participants (highest and lowest scores from each group), the summary of which is shown in Figure 39.

The initial aim of a user is to build a *mental model* of the tool. This model provides a framework for translating one's questions into their corresponding visual configurations within the system. It also provides a guideline for interpreting those visualizations and translating them back into meaningful answers. Two factors affect the successful formation of a mental model: grounding, and systematic exploration. First, the user should start from a familiar concept which she already knows and build its corresponding view in the tool.

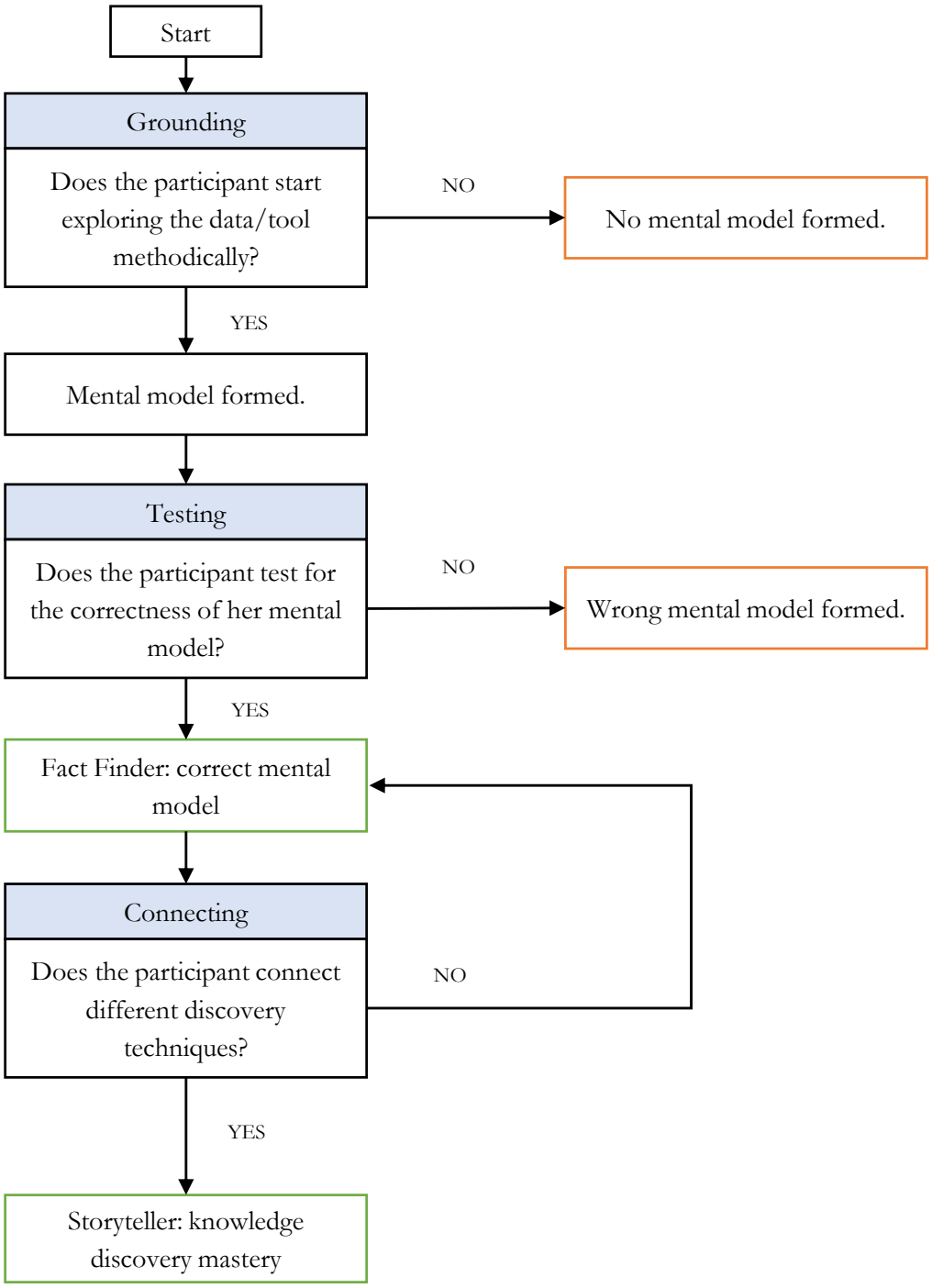


Figure 39: Phases of knowledge discovery.

This *grounding phase* provides a shared common ground of knowledge (as proposed by the grounding theory in communication [22]) which then allows the user to build a bridge between a concept she already knows and its corresponding pattern in the tool. For example, during the training session, participants initially built a map-view of the population distribution. As this concept is familiar to most participants and they have a reasonable understanding of the geography of the state and its population distribution, such a view acts as a grounding point for shaping a mental model of the tool. For some participants, the examples provided during the training session were sufficient. However, some participants decided to start their main session by further extending the grounding phase and looking for more familiar visualizations to better grasp how the system works and how they should interpret each view.

The second aspect that affects mental model creation is the methodical exploration of the data, where a participant starts with a simple visualization (often 2-dimensional), and after thoroughly understanding that view, adds more complexity to it (e.g., adding color, or making it 3-dimensional). From the four analyzed participants, only the lowest score participant of group Latency did not take a methodical approach to exploring the data. Instead, he started his session with forming complex and abstract visualizations, and although he had a hard time understanding them, he did not reduce the complexity of those views. Consequently, he never formed a mental model of the tool as it was evident in the mismatch between the questions he had in mind and the corresponding views he generated to answer those questions.

Although forming a mental model is a crucial step in utilizing a knowledge discovery tool, it must be fully *tested* before one can rely on it. The user must make sure how she thinks the system works does match how it actually works; otherwise one may form an incorrect mental model and eventually only produce false discoveries. Both high score participants dedicated a considerable amount of time into testing their mental models. For example, they used their contextual knowledge of the socio-economic status of Pennsylvania to check whether their findings matched with facts they already knew. They also used the map layer to gain additional information on what they were looking at and again make sure their interpretations were indeed correct. This was especially helpful for the group Latency participant as most of the time he had less visual information available. However, contrary

to this behavior, the lowest score participant of group Normal did not test his mental model. After a few visualizations, he quickly formed an incorrect mental model and then used that model throughout the experiment, resulting in many incorrect discoveries.

A common mistake by many participants that also affected the mental model of the lowest score participant of group Latency was the *decoupling issue*, mainly seen in interpreting scatter plots. In a scatter plot, each axis acts independently—the position of a point on the X axis is not determined by the variable selected for the Y axis. Nevertheless, many participants incorrectly assumed that one axis could filter the results on the other axis, even though during the training session they were explicitly told that EVA does not filter data and that axes of a scatter plot act independently. For example, one would assign "the percentage of jobs with low income" to the X axis and "the total number of jobs" to the Y axis. He would then assume that the visualization is representing the total number of jobs specifically for the low-income neighborhoods. This difficulty in decoupling dimensions led to many incorrect discoveries. It also indicates that sometimes even simple visualizations such as a scatter plot can be misinterpreted, even by graduate students who have had prior training in reading such charts.

Another source of mistakes was missing data. The LEHD variables related to gender, education, and ethnicity were only recorded for the last three years while the rest of the variables were collected for ten years. During the training session, all participants were explicitly told about the existence of missing variables and had worked with an example containing missing values to observe how it is represented in EVA. However, some participants forgot this fact and made incorrect observations when dealing with missing data. For example, the lowest score participant of group Normal created a visualization of jobs for the Asian race and explored it over time. He then noticed a sudden jump in the percentage of Asians who have a job from the year 2008 to 2009 (which is when census started collecting the data) and incorrectly assumed the reason for this increase is that many Asians have moved to Pennsylvania in that year.

The two successful participants had similar strategies in building their mental models. They both started with familiar concepts, explored the tool methodically by initially exploring simpler visualizations and then gradually adding complexity to it, and they both

thoroughly tested their assumptions to make sure their interpretations are correct. At this stage, a participant becomes a productive *fact finder* where she can successfully explore the data and make correct discoveries. An interesting observation was that fact finders often utilized a few *disocvery patterns*. A discovery pattern is a blueprint for finding several facts, all based on a similar set of operations. For example, once a participant learns how to use the map view to study the distribution of jobs in agriculture, she can then quickly apply the same pattern to other job categories and study those as well. Hence, a discovery pattern acts as a repeatable strategy for knowledge discovery. In the case of the high score participant of group Latency, he learned that by assigning a job sector to the Color dimension and then viewing the data over a map and across time, he can investigate if a job category is becoming more or less popular and on which regions of the state that change is happening. He then applied this pattern to multiple job categories to find several facts regarding the job trends. Once in a while, he also tried to expand his collection of discovery patterns by adding more complexity to it or even by trying a new pattern (e.g., a 2-dimensional non-map scatter plot). Overall, participants who formed correct mental models of the tool were able to form discovery patterns and apply them successfully to make several discoveries. However, the most successful participants took even a further step and went beyond just finding facts; they became *storytellers*.

The two high score participants of group Normal and Latency formed multiple discovery patterns. However, the one from group Normal was able to connect those discovery patterns further together to create a coherent narrative for several of his discoveries. As a fact finder, the participant from group Latency was able to find several correct discoveries; however, those discoveries were disconnected facts. For example, he discovered the trend of several job categories, yet each trend was mentioned as a self-contained discovery, unrelated to the previous discoveries. As a storyteller though, the participant from group Normal was able to connect consecutive facts to form a data-driven story, spanning across several aspects of a phenomenon. For example, when he found a trend in a job sector, he then investigated what caused that trend, how it correlates with education, and how all those facts affect the overall income level of that occupation. To achieve this capability, one needs to connect multiple discovery patterns and learn how each pattern can illuminate a particular aspect of the story.

For example, a map view over time may be useful for investigating trends, but studying the effect of education on that job category may require an abstract scatter plot. To form these connections, the high score participant of group Normal initially applied several discovery patterns to find the same discovery. Although *rediscovering* a fact may seem unnecessary, in practice it provided a *grounding* platform for connecting disjoint abstract discovery patterns into a single sophisticated data exploration apparatus. Such a deep understanding of the tool and ability to tackle complex representations allowed him to thoroughly examine the data and produce meaningful and interesting stories.

In summary, our analysis indicates that a productive knowledge discovery strategy is based on acquiring a correct mental model of how the data exploration tool works, employing that model to construct several discovery patterns, and ultimately connecting those discovery templates to form a powerful mechanism for producing coherent and data-driven stories. When an HDI tool facilitates this process, it can turn into an invaluable medium for knowledge discovery, as not only it expedites the process of finding relevant answers to the user's questions, but it also acts as an engine for hypothesis generation. This is particularly important in analyzing large and high-dimensional data, as even finding the proper set of questions can sometimes be challenging. For example, when during a hands-on session with the creators of the LEHD dataset, they were exposed to the example mentioned in Section 3.2.2, the researchers mentioned they did not think asking such a question was even possible from their data. In this regard, a big data HDI system should be both a tool for expanding one's domain of questions and also providing mechanisms for answering those questions. Consequently, we propose our design guidelines for an HDI system acting on large and high-dimensional data as follows:

1. **Incremental Latency:** An HDI system should be interactive, fast, and responsive. Interaction delays should be kept under 100ms and initial query-response times should not exceed a few seconds. However, the initial response can be inaccurate. Instead, the system can gradually improve its answers and increase the accuracy of its results over time. Such an incremental latency is acceptable for knowledge discovery purposes. Moreover, it does not demand expensive hardware or optimization requirements and most notably can scale well with large data sizes. Such incremental systems should also

support non-episodic interaction, where a new query from the user would terminate the ongoing process, freeing up computational resources for incremental processing of the new query. The system should also provide clear indicators of its inaccuracy level, such as confidence intervals or a progress bar specifying what percentage of the total data has been used to calculate the current answer.

2. **Ease of Navigation:** Considering the large size of big datasets, an HDI system should provide intuitive and interactive means of navigation within the data. Such mechanisms should facilitate navigation across scales (e.g., zooming in and out), within a scale (e.g., pan, or rotation for 3-dimensional views), and across projections. The ability to dissect the data from any desired projection is especially vital in analyzing high-dimensional data. Furthermore, to simplify navigating to regions already explored by the user, the tool should provide bookmarking capabilities, hence allowing the user to jump from one spatiotemporal point to another.
3. **Contextual Layers:** Contextual information can enhance one's data exploration experience and provide additional data that is often necessary for finding the real cause of a phenomenon under investigation. For example, a map layer can augment any geospatial dataset and even hint at the corresponding socio-economic metrics that may affect the variable under study. Providing such layers is even more imperative for incremental systems as they reduce the impact that inaccurate and partial results may inflict on the knowledge discovery process.
4. **Controlling the Degrees of Freedom:** Allowing the user to generate complex views can result in a higher number of unintended mistakes. An HDI system should carefully balance how much freedom it grants and if necessary, limit the number of dimensions a user can study at the same time. For example, sometimes 3-dimensional scatter plots are more confusing than helpful.
5. **Learning Tutorials:** A vital step in utilizing an HDI tool is to assist the user in building a correct mental model of how the tool works and how she can translate her questions into its visual space and then interpret those patterns. To achieve this goal, an HDI system should provide tutorials and train the user. Such tutorials should follow a systematic approach, where they should start from simpler visualizations and then

gradually add complexity. Furthermore, the system should start with concepts that are already known to the user to increase the chance of finding a common ground of communication. Additionally, the system should provide extensive feedback, verifying that user's mental model is indeed correct. Also, the training material should provide examples of the types of questions that cannot be answered by the tool to help the user understand what the limitations of the HDI system are.

6. **Continuous Assistance:** Besides the introductory tutorials, an HDI system can increase one's chance of making correct discoveries by continuously providing feedback on what the user is seeing. For example, the system can provide automatically generated texts, presenting a superficial interpretation of the current view. Such feedbacks can act as a scaffold [68], creating a semi-guided space in which the user is less prone to mistakes such as the decoupling issue. Furthermore, missing data should always be represented distinctly (e.g., a separate color reserved only for missing values) or its existence should be explicitly revealed to the user to avoid making incorrect discoveries.
7. **Facilitating the Formation of Discovery Patterns:** After a user learns how to use the system, the system can provide occasional suggestions on similar visualizations to one the user is currently viewing. Such visualizations can keep all but one of their variable similar, hence creating a window for investigating a different variable via a visualization template already known to the user. This approach can facilitate the formation of discovery patterns and extend the repertoire of knowledge acquisition techniques known to the user. For example, if a user is looking at a job distribution over a map and is using color to represent the percentage of jobs in that category for each neighborhood, the system can propose similar visualizations that look at a different job category on a colored map view. This can help the user understand such a pattern can be applied to other variables as well and result in similar discoveries.
8. **Facilitating the Connection of Discovery Patterns:** For an advanced user, the system can also suggest dissimilar views that have one or two variables in common. For example, if a user is looking at a job category over a map, the system can suggest a view of the same job category vs. education level, presented in a scatter plot. Such dissimilar but related views can provide opportunities for the user to connect different discovery

patterns together and ultimately become a storyteller.

BIBLIOGRAPHY

[1] CREATE Lab. http://www.cmucreatelab.org/, 2017.

[2] CREATE's Explorables. http://explorables.cmucreatelab.org/, 2017.

[3] Longitudinal Employer-Household Dynamics. http://lehd.ces.census.gov/, 2017.

[4] three.js, JavaScript 3D Library. http://threejs.org/, 2017.

[5] Sameer Agarwal, Barzan Mozafari, Aurojit Panda, Henry Milner, Samuel Madden, and Ion Stoica. BlinkDB: Queries with Bounded Errors and Bounded Response Times on Very Large Data. In *Proceedings of the 8th ACM European Conference on Computer Systems*, EuroSys '13, pages 29–42, New York, NY, USA, 2013. ACM.

[6] Saman Amirpour Amraii. Explorable Visual Analytics. http://eva.cmucreatelab.org/, 2017.

[7] Peter Bakkum and Kevin Skadron. Accelerating SQL Database Operations on a GPU with CUDA. In *Proceedings of the 3rd Workshop on General-Purpose Computation on Graphics Processing Units*, GPGPU '10, pages 94–103, New York, NY, USA, 2010. ACM.

[8] Mike Barnett, Badrish Chandramouli, Robert DeLine, Steven Drucker, Danyel Fisher, Jonathan Goldstein, Patrick Morrison, and John Platt. Stat!: An Interactive Analytics Environment for Big Data. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, SIGMOD '13, pages 1013–1016, New York, NY, USA, 2013. ACM.

[9] E. Wes Bethel, Hank Childs, and Charles Hansen. *High Performance Visualization: Enabling Extreme-Scale Scientific Insight*. CRC Press, October 2012.

[10] Johanna Beyer, Ali Al-Awami, Narayanan Kasthuri, Jeff W Lichtman, Hanspeter Pfister, and Markus Hadwiger. ConnectomeExplorer: Query-guided visual analysis of large volumetric neuroscience data. *IEEE transactions on visualization and computer graphics*, 19(12):2868–2877, December 2013.

[11] Johanna Beyer, Markus Hadwiger, and Hanspeter Pfister. A Survey of GPU-Based Large-Scale Volume Visualization. In *Eurographics Conference on Visualization (EuroVis)*, page to appear, Swansea, UK, 2014.

[12] Johanna Beyer, Markus Hadwiger, and Hanspeter Pfister. State-of-the-Art in GPU-Based Large-Scale Volume Visualization. *Computer Graphics Forum*, 34(8):13–37, December 2015.

[13] T. Blascheck, M. John, K. Kurzhals, S. Koch, and T. Ertl. VA2: A Visual Analytics Approach for Evaluating Visual Analytics Applications. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):61–70, January 2016.

[14] M. Bostock, V. Ogievetsky, and J. Heer. D3 Data-Driven Documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, December 2011.

[15] Alberto Cairo. *The Functional Art: An Introduction to Information Graphics and Visualization*. New Riders, Berkeley, California, 1 edition edition, September 2012.

[16] Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.

[17] Stephen M. Casner. Task-analytic Approach to the Automated Design of Graphic Presentations. *ACM Trans. Graph.*, 10(2):111–151, April 1991.

[18] Joel Chan, Katherine Fu, Christian Schunn, Jonathan Cagan, Kristin Wood, and Kenneth Kotovsky. On the Benefits and Pitfalls of Analogies for Innovative Design: Ideation Performance Based on Analogical Distance, Commonness, and Modality of Examples. *Journal of Mechanical Design*, 133(8):081004–081004–11, August 2011.

[19] R. Chang, C. Ziemkiewicz, T. M. Green, and W. Ribarsky. Defining Insight for Visual Analytics. *IEEE Computer Graphics and Applications*, 29(2):14–17, March 2009.

[20] Hsiang-Ting Chen, Tovi Grossman, Li-Yi Wei, Ryan Schmidt, Björn Hartmann, George Fitzmaurice, and Maneesh Agrawala. History Assisted View Authoring for 3D Models. CHI, 2014.

[21] Yang Chen, Jing Yang, and W. Ribarsky. Toward effective insight management in visual analytics systems. In *2009 IEEE Pacific Visualization Symposium*, pages 49–56, April 2009.

[22] Herbert H Clark and Susan E Brennan. Grounding in communication. *Perspectives on socially shared cognition*, 13(1991):127–149, 1991.

[23] William S. Cleveland and Robert McGill. Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods. *Journal of the American Statistical Association*, 79(387):531–554, 1984.

[24] Jeffrey Cohen, Brian Dolan, Mark Dunlap, Joseph M. Hellerstein, and Caleb Welton. MAD Skills: New Analysis Practices for Big Data. *Proc. VLDB Endow.*, 2(2):1481–1492, August 2009.

[25] Gintautas Dzemyda, Olga Kurasova, and Julius Žilinskas. *Multidimensional Data Visualization: Methods and Applications*. Springer, November 2012.

[26] N Elmqvist. Embodied Human-Data Interaction. In *ACM CHI 2011 Workshop "Embodied Interaction: Theory and Practice in HCI*, pages 104–107, 2011.

[27] N. Elmqvist, P. Dragicevic, and J. Fekete. Rolling the Dice: Multidimensional Visual Exploration using Scatterplot Matrix Navigation. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1539–1148, November 2008.

[28] J.-D. Fekete. Visual Analytics Infrastructures: From Data Management to Exploration. *Computer*, 46(7):22–29, July 2013.

[29] Yu Feng, Rupert A. C. Croft, Tiziana Di Matteo, Nishikanta Khandai, Randy Sargent, Illah Nourbakhsh, Paul Dille, Chris Bartley, Volker Springel, Anirban Jana, and Jeffrey Gardner. Terapixel Imaging of Cosmological Simulations. *The Astrophysical Journal Supplement Series*, 197(2):18, December 2011.

[30] Danyel Fisher, Igor Popov, Steven Drucker, and m.c. schraefel. Trust Me, I'M Partially Right: Incremental Visualization Lets Analysts Explore Large Datasets Faster. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 1673–1682, New York, NY, USA, 2012. ACM.

[31] Benjamin Jotham Fry. *Computational Information Design*. Thesis, Massachusetts Institute of Technology, 2004. Thesis (Ph. D.)–Massachusetts Institute of Technology, School of Architecture and Planning, Program in Media Arts and Sciences, 2004.

[32] Gartheeban Ganeshapillai, Joel Brooks, and John Guttag. Rapid Data Exploration and Visual Data Mining on Relational Data. In *Proc. of the KDD 2014 Workshop on Interactive Data Exploration and Analytics*, 2014.

[33] S. Gratzl, A. Lex, N. Gehlenborg, H. Pfister, and M. Streit. LineUp: Visual Analysis of Multi-Attribute Rankings. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2277–2286, December 2013.

[34] Jim Gray, Surajit Chaudhuri, Adam Bosworth, Andrew Layman, Don Reichart, Murali Venkatrao, Frank Pellow, and Hamid Pirahesh. Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals. *Data Mining and Knowledge Discovery*, 1(1):29–53, March 1997.

[35] H. Guo, S. R. Gomez, C. Ziemkiewicz, and D. H. Laidlaw. A Case Study Using Visualization Interaction Logs and Insight Metrics to Understand How Analysts Arrive

at Insights. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):51–60, January 2016.

[36] I. Herman, G. Melancon, and M.S. Marshall. Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):24–43, January 2000.

[37] Anthony JG Hey, Stewart Tansley, Kristin Michele Tolle, and others. The fourth paradigm: Data-intensive scientific discovery. 2009.

[38] S. Ingram, T. Munzner, V. Irvine, M. Tory, S. Bergner, and T. Möller. DimStiller: Workflows for dimensional analysis and reduction. In *2010 IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 3–10, October 2010.

[39] Stephen Ingram, Tamara Munzner, and Marc Olano. Glimmer: Multilevel MDS on the GPU. *Visualization and Computer Graphics, IEEE Transactions on*, 15(2):249–261, 2009.

[40] T. Isenberg, P. Isenberg, Jian Chen, M. Sedlmair, and T. Moller. A Systematic Review on the Practice of Evaluating Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2818–2827, December 2013.

[41] Daniel Keim, Gennady Andrienko, Jean-Daniel Fekete, Carsten Görg, Jörn Kohlhammer, and Guy Melançon. *Visual Analytics: Definition, Process, and Challenges*. Springer, 2008.

[42] Daniel A Keim, Jörn Kohlhammer, Geoffrey Ellis, and Florian Mansmann. *Mastering The Information Age-Solving Problems with Visual Analytics*. Florian Mansmann, 2010.

[43] Brian A. Kidd, Lauren A. Peters, Eric E. Schadt, and Joel T. Dudley. Unifying immunology with informatics and multiscale biology. *Nature Immunology*, 15(2):118–127, February 2014.

[44] Albert Kim, Eric Blais, Aditya Parameswaran, Piotr Indyk, Sam Madden, and Ronitt Rubinfeld. Rapid Sampling for Visualizations with Ordering Guarantees. *Proc. VLDB Endow.*, 8(5):521–532, January 2015.

[45] David Klahr and Kevin Dunbar. Dual Space Search During Scientific Reasoning. *Cognitive Science*, 12(1):1–48, January 1988.

[46] R. Kosara and J. Mackinlay. Storytelling: The Next Step for Visualization. *Computer*, 46(5):44–50, May 2013.

[47] D. J. Lehmann and H. Theisel. Optimal Sets of Projections of High-Dimensional Data. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):609–618, January 2016.

[48] L. Lins, J.T. Klosowski, and C. Scheidegger. Nanocubes for Real-Time Exploration of Spatiotemporal Datasets. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2456–2465, December 2013.

[49] Z. Liu and J. Heer. The Effects of Interactive Latency on Exploratory Visual Analysis. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2122–2131, December 2014.

[50] Zhicheng Liu, Biye Jiang, and Jeffrey Heer. imMens: Real-time Visual Querying of Big Data. *Computer Graphics Forum*, 32(3pt4):421–430, June 2013.

[51] Jock Mackinlay. Automating the Design of Graphical Presentations of Relational Information. *ACM Trans. Graph.*, 5(2):110–141, April 1986.

[52] Sergey Melnik, Andrey Gubarev, Jing Jing Long, Geoffrey Romer, Shiva Shivakumar, Matt Tolton, and Theo Vassilakis. Dremel: Interactive Analysis of Web-Scale Datasets. In *Proc. of the 36th Int'l Conf on Very Large Data Bases*, pages 330–339, 2010.

[53] F. Miranda, L. Lins, J. Klosowski, and C. Silva. TopKube: A Rank-Aware Data Cube for Real-Time Exploration of Spatiotemporal Data. *IEEE Transactions on Visualization and Computer Graphics*, PP(99):1–1, 2017.

[54] Richard Mortier, Hamed Haddadi, Tristan Henderson, Derek McAuley, and Jon Crowcroft. Challenges & opportunities in human-data interaction. *DE2013: Open Digital, MediaCityUK, Salford, UK*, 2013.

[55] Richard Mortier, Hamed Haddadi, Tristan Henderson, Derek McAuley, and Jon Crowcroft. Human-Data Interaction: The Human Face of the Data-Driven Society. SSRN Scholarly Paper ID 2508051, Social Science Research Network, Rochester, NY, October 2014.

[56] P. H. Nguyen, K. Xu, A. Wheat, B. L. W. Wong, S. Attfield, and B. Fields. SensePath: Understanding the Sensemaking Process Through Analytic Provenance. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):41–50, January 2016.

[57] Mary H. Nichols, George B. Ruyle, and Illah R. Nourbakhsh. Very-High-Resolution Panoramic Photography to Improve Conventional Rangeland Monitoring. *Rangeland Ecology & Management*, 62(6):579–582, November 2009.

[58] Kai A. Olsen, Robert R. Korfhage, Kenneth M. Sochats, Michael B. Spring, and James G. Williams. Visualization of a document collection: The vibe system. *Information Processing & Management*, 29(1):69–81, January 1993.

[59] C. A. L. Pahins, S. A. Stephens, C. Scheidegger, and J. L. D. Comba. Hashedcubes: Simple, Low Memory, Real-Time Visual Exploration of Big Data. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):671–680, January 2017.

[60] Y. Park, M. Cafarella, and B. Mozafari. Visualization-aware sampling for very large databases. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, pages 755–766, May 2016.

[61] Alessandro Perina, Dongwoo Kim, Andrzej Truski, and Nebojsa Jojic. Skim-reading thousands of documents in one minute: Data indexing and visualization for multifarious search. In *Workshop on Interactive Data Exploration and Analytics (IDEA'14) at KDD*, 2014.

[62] Yulin Qin and Herbert A. Simon. Laboratory Replication of Scientific Discovery Processes. *Cognitive Science*, 14(2):281–312, March 1990.

[63] David N. Reshef, Yakir A. Reshef, Hilary K. Finucane, Sharon R. Grossman, Gilean McVean, Peter J. Turnbaugh, Eric S. Lander, Michael Mitzenmacher, and Pardis C. Sabeti. Detecting Novel Associations in Large Data Sets. *Science*, 334(6062):1518–1524, December 2011.

[64] Hans Rosling, Rönnlund A Rosling, and Ola Rosling. New software brings statistics beyond the eye. *Statistics, Knowledge and Policy: Key Indicators to Inform Decision Making. Paris, France: OECD Publishing*, pages 522–530, 2005.

[65] P. Ruchikachorn and K. Mueller. Learning Visualizations by Analogy: Promoting Visual Literacy through Visualization Morphing. *IEEE Transactions on Visualization and Computer Graphics*, 21(9):1028–1044, September 2015.

[66] Saman Amirpour Amraii. Interactive Exploration of LEHD, A Case Study in Knowledge Discovery. LED Partnership Workshop 2014, 2014.

[67] Randy Sargent, Chris Bartley, Paul Dille, Jeff Keller, Illah Nourbakhsh, and Rich LeGrand. Timelapse GigaPan: Capturing, Sharing, and Exploring Timelapse Gigapixel Imagery. *Fine International Conference on Gigapixel Imaging for Science*, November 2010.

[68] R. Keith Sawyer. *The Cambridge Handbook of the Learning Sciences*. Cambridge University Press, Cambridge; New York, 2006.

[69] J. Schneider and P. Rautek. A Versatile and Efficient GPU Data Structure for Spatial Indexing. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):911–920, January 2017.

[70] H. J. Schulz, M. Angelini, G. Santucci, and H. Schumann. An Enhanced Visualization Process Model for Incremental Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 22(7):1830–1842, July 2016.

[71] Christian D Schunn and David Klahr. A 4-space model of scientific discovery. In *Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society*, pages 106–111, 1995.

[72] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *, IEEE Symposium on Visual Languages, 1996. Proceedings*, pages 336–343, September 1996.

[73] Ben Shneiderman. Extreme Visualization: Squeezing a Billion Records into a Million Pixels. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 3–12, New York, NY, USA, 2008. ACM.

[74] Ben Shneiderman. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Addison-Wesley, Boston, 5th ed edition, 2010.

[75] Ben Shneiderman and Catherine Plaisant. Strategies for Evaluating Information Visualization Tools: Multi-dimensional In-depth Long-term Case Studies. In *Proceedings of the 2006 AVI Workshop on BEyond Time and Errors: Novel Evaluation Methods for Information Visualization*, BELIV '06, pages 1–7, New York, NY, USA, 2006. ACM.

[76] A. Silberschatz and A. Tuzhilin. What makes patterns interesting in knowledge discovery systems. *IEEE Transactions on Knowledge and Data Engineering*, 8(6):970–974, December 1996.

[77] Herbert Alexander Simon, David Klahr, and Kenneth Kotovsky. *Complex Information Processing: The Impact of Herbert A. Simon*. Psychology Press, 1989. Google-Books-ID: D7LePkUFn7gC.

[78] C. D. Stolper, A. Perer, and D. Gotz. Progressive Visual Analytics: User-Driven Visual Exploration of In-Progress Analytics. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1653–1662, December 2014.

[79] C. Stolte, D. Tang, and P. Hanrahan. Multiscale visualization using data cubes. *IEEE Transactions on Visualization and Computer Graphics*, 9(2):176–187, April 2003.

[80] Martin Theus. High-dimensional Data Visualization. In *Handbook of Data Visualization*, Springer Handbooks Comp.Statistics, pages 151–178. Springer Berlin Heidelberg, January 2008.

[81] C. Turkay, E. Kaya, S. Balcisoy, and H. Hauser. Designing Progressive and Interactive Analytics Processes for High-Dimensional Data Analysis. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):131–140, January 2017.

[82] Wouter R. van Joolingen and Ton De Jong. Supporting hypothesis generation by learners exploring an interactive computer simulation. *Instructional Science*, 20(5-6):389–404, September 1991.

[83] Z. Wang, N. Ferreira, Y. Wei, A. S. Bhaskar, and C. Scheidegger. Gaussian Cubes: Real-Time Modeling for Visual Exploration of Large Multidimensional Datasets. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):681–690, January 2017.

[84] Colin Ware. *Information Visualization, Third Edition: Perception for Design*. Morgan Kaufmann, Waltham, MA, 3 edition edition, June 2012.

[85] C. Kenneth Waters. The Nature and Context of Exploratory Experimentation: An Introduction to Three Case Studies of Exploratory Research. *History and Philosophy of the Life Sciences*, 29(3):275–284, 2007.

[86] Leland Wilkinson. The Grammar of Graphics. In James E. Gentle, Wolfgang Karl Härdle, and Yuichi Mori, editors, *Handbook of Computational Statistics*, Springer Handbooks of Computational Statistics, pages 375–414. Springer Berlin Heidelberg, January 2012.

[87] Leland Wilkinson, D. Wills, D. Rope, A. Norton, and R. Dubbs. *The Grammar of Graphics*. Springer, New York, 2nd edition edition, August 2005.

[88] M. Williams and T. Munzner. Steerable, Progressive Multidimensional Scaling. In *IEEE Symposium on Information Visualization, 2004. INFOVIS 2004*, pages 57–64, 2004.

[89] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager: Exploratory Analysis via Faceted Browsing of Visualization Recommendations. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):649–658, January 2016.

[90] E. Zgraggen, A. Galakatos, A. Crotty, J. D. Fekete, and T. Kraska. How Progressive Visualizations Affect Exploratory Analysis. *IEEE Transactions on Visualization and Computer Graphics*, 23(8):1977–1987, August 2017.