

# Transforming the Reading Experience of Scientific Documents with Polymorphism

by

Damien Masson

A thesis  
presented to the University of Waterloo  
in fulfillment of the  
thesis requirement for the degree of  
Doctor of Philosophy  
in  
Computer Science

Waterloo, Ontario, Canada, 2023

© Damien Masson 2023

## EXAMINING COMMITTEE MEMBERSHIP

---

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner	Niklas Elmqvist Professor, College of Information Studies, University of Maryland
Supervisors	Daniel Vogel Associate Professor, Cheriton School of Computer Science, University of Waterloo
	Edward Lank Professor, Cheriton School of Computer Science, University of Waterloo
	Géry Casiez Professor, Department of Computer Science, Université de Lille
	Sylvain Malacria Research Scientist, Inria, Loki group, Centre de recherche Lille – Nord Europe
Internal Members	George Labahn Professor, Cheriton School of Computer Science, University of Waterloo
	Jian Zhao Assistant Professor, Cheriton School of Computer Science, University of Waterloo
Internal-external Member	Leah Zhang-Kennedy Assistant Professor, Stratford School of Interaction Design and Business, University of Waterloo

## AUTHOR'S DECLARATION

---

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

## STATEMENT OF CONTRIBUTIONS

---

This dissertation includes first-authored peer-reviewed material that has appeared in conference and journal proceedings published by the Association for Computing Machinery (ACM). The ACM’s policy is as follows<sup>1</sup>:

*“Authors can include partial or complete papers of their own (and no fee is expected) in a dissertation as long as citations and DOI pointers to the Versions of Record in the ACM Digital Library are included. Authors can use any portion of their own work in presentations and in the classroom (and no fee is expected).”*

The following list serves as a declaration of the works included in this dissertation. This material is expanded and revised from the original publication.

### **Portions of Chapter 3**

Damien Masson, Sylvain Malacria, Géry Casiez, and Daniel Vogel. 2023. Charagraph: Interactive Generation of Charts for Realtime Annotation of Data-Rich Paragraphs. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (CHI '23). <https://doi.org/10.1145/3544548.3581091>

### **Portions of Chapter 4**

Damien Masson, Sylvain Malacria, Géry Casiez, and Daniel Vogel. 2023. Statlator: Interactive Translation of NHST and Estimation Statistics Reporting Styles in Scientific Documents. In *The 36th Annual ACM Symposium on User Interface Software and Technology* (Conditionally Accepted, UIST '23).

### **Portions of Chapter 5**

Damien Masson, Sylvain Malacria, Daniel Vogel, Edward Lank, and Géry Casiez. 2023. ChartDetective: Easy and Accurate Interactive Data Extraction from Complex Vector Charts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (CHI '23). <https://doi.org/10.1145/3544548.3581113>

### **Portions of Chapter 6**

Damien Masson, Sylvain Malacria, Edward Lank, and Géry Casiez. 2020. Chameleon: Bringing Interactivity to Static Digital Documents. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (CHI '20). <https://doi.org/10.1145/3313831.3376559>

Damien Masson, Sylvain Malacria, Edward Lank, and Géry Casiez. 2020. Bringing Interactivity to Research Papers and Presentations with Chameleon. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems* (CHI EA '20). <https://doi.org/10.1145/3334480.3383173>

<sup>1</sup> <https://authors.acm.org/author-resources/author-rights>

## ABSTRACT

---

Despite the opportunities created by digital reading, documents remain mostly static and mimic paper. Any improvement in the shape or form of documents has to come from authors who contend with current digital formats, workflows, and software and who impose a presentation to readers. Instead, I propose the concept of *polymorphic documents* which are documents that can change in form to offer better representations of the information they contain. I believe that multiple representations of the same information can help readers, and that any document can be made polymorphic, with no intervention from the original author. This thesis presents four projects investigating what information can be obtained from existing documents, how this information can be better represented, and how these representations can be generated using only the source document. To do so, I draw upon theories showing the benefit of presenting information using multiple representations; the design of interactive systems to support morphing representations; and user studies to evaluate system usability and the benefits of the new representations on reader comprehension.

## ACKNOWLEDGMENTS

---

To Daniel Vogel, Edward Lank, Géry Casiez, and Sylvain Malacria, I am forever grateful. As cliché as it may sound, this thesis would have not been possible without you. You have been involved in all key moments of my research path so far, including, for Géry and Sylvain, my very first experience with research. Thank you for the time, guidance, support, and trust you gave me during all these years.

To Niklas Elmqvist, Leah Zhang-Kennedy, George Labahn, and Jian Zhao, thank you for accepting to serve on my committee and carefully reading my PhD thesis. I could not have hoped for a more supportive committee. My defence was an insightful and memorable moment thanks to you.

To my friends, Glauca Melo and Nils Lukas, thank you for making my PhD experience so enjoyable. I am so grateful to have shared the PhD journey with you. I cannot count the number of coffees, drinks, lunches, dinners, walks, and late-night discussions. You have been the most supportive friends one could hope for.

To my friends and colleagues in the HCI lab, thank you for making the lab such a supportive and welcoming environment. As the time I have spent in the lab can attest, I have truly enjoyed my time with you. Thank you for all these discussions, coffee runs, user studies, and pushes for deadlines.

À mes parents, Claudine et Daniel, et à mon frère, Thibaud, merci d'avoir toujours cru en moi, de m'avoir toujours soutenu et de m'avoir toujours fait confiance. Même à travers un écran, nos discussions étaient un soutien précieux durant ma thèse.

To my loving partner, Elyse, I cannot thank you enough for your infallible support throughout. Thank you for listening to my research ideas no matter the time of day. Thank you for turning my disappointments into positives. Thank you for sharing the excitement in my successes.

## TABLE OF CONTENTS

---

	List of Figures . . . . .	ix
	List of Tables . . . . .	xii
	List of Algorithms. . . . .	xiii
1	INTRODUCTION . . . . .	1
1.1	Research Objectives and Overview . . . . .	2
1.2	Contributions . . . . .	4
1.3	Dissertation Outline . . . . .	6
2	RELATED WORK . . . . .	7
2.1	How Readers Consume Documents . . . . .	7
2.2	Benefits of Polymorphic Content . . . . .	8
2.3	Changing Workflows, File Formats, and Software . . . . .	12
2.4	Empowering Readers by Augmenting Existing Documents . . . . .	13
3	ANNOTATING DATA-RICH PARAGRAPHS WITH CHARTS . . . . .	15
3.1	Motivation . . . . .	15
3.2	Background and Related Work . . . . .	16
3.3	Characterization of Charagraph . . . . .	19
3.4	Charagraph Document Viewer . . . . .	22
3.5	User Study . . . . .	28
3.6	Study Part 1: Usability . . . . .	29
3.7	Study Part 2: Utility . . . . .	32
3.8	Discussion . . . . .	36
3.9	Conclusion . . . . .	40
4	TRANSLATING BETWEEN STATISTICS REPORTING STYLES. . . . .	41
4.1	Motivation . . . . .	41
4.2	Background and Related Work . . . . .	43
4.3	What is Reported at CHI? . . . . .	45
4.4	Converting Statistical Reports . . . . .	47
4.5	Conversion Accuracy . . . . .	52
4.6	Statslator PDF Viewer . . . . .	57
4.7	Discussion . . . . .	61
4.8	Conclusion . . . . .	64

Charagraph

Statslator

5	EXTRACTING DATA FROM VECTOR CHARTS . . . . .	65
— ChartDetective —		
5.1	Motivation . . . . .	65
5.2	Background and Related Work . . . . .	67
5.3	Challenges and Design Goals . . . . .	71
5.4	ChartDetective . . . . .	73
5.5	Usability Study . . . . .	80
5.6	Data Quality Study . . . . .	86
5.7	Discussion . . . . .	89
5.8	Conclusion . . . . .	93
6	BRINGING INTERACTIVITY TO STATIC DIGITAL DOCUMENTS. . . . .	94
— Chameleon —		
6.1	Motivation . . . . .	94
6.2	Background and Related Work . . . . .	96
6.3	Chameleon . . . . .	100
6.4	Feature Matching Algorithm Comparison . . . . .	104
6.5	Feasibility Evaluation of Chameleon . . . . .	107
6.6	First-use study . . . . .	109
6.7	Discussion . . . . .	111
6.8	Conclusion and Future Work . . . . .	113
7	CONCLUSION AND FUTURE DIRECTIONS . . . . .	114
7.1	Summary of Contributions . . . . .	114
7.2	Augmenting Documents by Readers, Publishers, or Authors? . . . . .	115
7.3	Limitations . . . . .	116
7.4	Future Directions . . . . .	117
7.5	Final Word . . . . .	120
	Bibliography . . . . .	121
	Appendices	
A	EXTRACTION ALGORITHMS USED IN CHARTDETECTIVE . . . . .	165
A.1	Extraction of Box Plots . . . . .	165
A.2	Extraction of Bars . . . . .	166
A.3	Extraction of Scatter Plots . . . . .	166
A.4	Extraction of Axes . . . . .	166
A.5	Extraction of Lines . . . . .	166
A.6	Splitting Compounded Shapes . . . . .	167
	Glossary . . . . .	168



## LIST OF FIGURES

---

1.1	Design space of complementary representations. Representations should vary in their media types (vertical) and level of abstraction (horizontal). Dashed boxes are examples of representations commonly found in static digital documents. . . . .	2
1.2	Chapters with main research questions ( <i>italic</i> ), main results, and research path connecting each chapter. Highlighted text and arrows show the dependencies and motivation that lead to the next research question.	3
3.1	(a) Charagraphs are in-situ visualizations of numeric data included within text that are dynamically generated (b) by delimiting a selection and (c) selecting a data group. Charagraphs support common data exploration tasks through interactive features such as (d) identifying and (e) comparing values. . . . .	16
3.2	Information that can be leveraged from the text with examples of the design of the Charagraph. . . . .	20
3.3	Level of integration of the chart and text representations. Dashed outline indicates levels specific to Charagraphs. . . . .	21
3.4	Interaction contexts for different data exploration tasks.	22
3.5	Selecting data to visualize: (a) delimiting using traditional text select or marquee selection; (b) selecting one of the data groups suggested by the system. . . .	23
3.6	Changing visualizations. All transitions are animated to keep track of changes. . . . .	24
3.7	Combining information in two Charagraphs: (a) dragging a Charagraph onto a drop zone in another Charagraph combines data in different ways; (b) adding a new series; or (c) adding error bars. . . . .	25
3.8	Text and chart interactions: (a) hovering above numbers in text or chart elements identifies connections; (b) moving the cursor on the Y axis controls a threshold to filter numbers; (c) clicking two values compare them and clicking more than 2 values calculates the mean; (d) right-clicking opens a contextual menu to sort. . . . .	26
3.9	Result summary of the usability part of the study. Mean and standard deviation of participants' answers to the SUS and the statements on a 5-point scale. Average time to create a Charagraph. . . . .	31

3.10	Results to (a) the raw NASA-TLX; (b) the average time to answer; (c) the average percent of incorrect answers; and (d) the 5-point scale statements. Error bars represented standard deviation. Statistical significance represented by a '*' . . . . .	35
4.1	Statslator takes existing statistical reports (a) using NHST or estimation; (b) calculates all possible statistical values using accurate conversion equations; (c) shows the report using graphical and interactive figures configurable by readers. . . . .	42
4.2	Upset plot showing the number of papers that report different sets of CIs, means, standard deviations, and p-values. Lines connecting dots across rows indicate what values are in the intersecting set. . . . .	46
4.3	Statslator user interface after selecting a paragraph with statistical information from WatchWriter [134]. The panel on the right shows the statistical measures extracted and calculated, as well as a configurable graphical report. . . . .	57
5.1	ChartDetective is a system capable of recovering a chart underlying data by leveraging its vector representation. Users select figures from PDF documents and then (A, B) drag-and-drop elements that they wish to extract. (C) The extracted data can be leveraged for downstream tasks such redesigning or interacting with the figure. . . . .	66
5.2	The same chart can be formed with different vector shape arrangements: (a) ONE-TO-ONE when shapes match semantic elements; (b) ONE-TO-MANY when one shape maps to multiple semantic elements; and (c) MANY-TO-ONE when many shapes map to a single semantic element. . . . .	71
5.3	ChartDetective is composed of three views: the Chart View showing the chart being extracted; the Data Table with the tables for X and Y axes and the extracted values; the interactive Reconstructed Chart. . . . .	74
5.4	(a) Shapes in the can be filtered (b) by colour or (c) by shape. (d) Filters can also be combined. . . . .	75
5.5	During drag-and-drop interactions, (A) drop zones compatible with the selection are highlighted to indicate <i>where</i> elements can be dropped. (B) The drop zone turns blue if hovered with a selection of the proper type, and (C) red otherwise. . . . .	78
5.6	Charts that participants had to extract during the usability study. Charts were extracted from the following CHI proceedings (left-to-right, top-to-bottom): [231] [418] [221] [113] [41] [389] [59] [19] [107] [183] [252] [178]	80

5.7 Minimal example of a chart generated by Microsoft Excel and for which the data is imprecisely depicted. Despite the bars representing exact values (10 and 20), they do not consistently line up with corresponding grid lines: the grid line is either below (blue bar) or above (orange bar). . . . . 89

6.1 (1) Static documents such as PDF (left) and presentations (right) can be augmented with Chameleon by (2) locating corresponding interactive figures in the Chameleon database and (3) overlaying them on top of the existing documents. . . . . 95

6.2 *Repeatability vs scales, and Mean Correct Matches vs thresholds of our dataset.* . . . . . 103

6.3 Evolution the true positive rate and false positive rate (dashed lines) depending on the feature-matching algorithm threshold and scale at which a figure is displayed. The y axis uses a logit scale. . . . . 106

7.1 Life-cycle of a document with respect to the actors and resources available. Solid lines indicate the resource is available. Dashed lines indicate the resource is less likely to be available. As such, (c) readers are more likely to have access to the document than the code or data. Inversely, (a) authors have access to all code, system, and data. . . . . 115

7.2 Future directions for exploring the design space of polymorphic content. In grey are the directions that this thesis explored. In blue are future directions I believe to be promising: (a) visuals into text; (b) making algorithms more visual; and (c) making math more concrete. . . . . 118

## LIST OF TABLES

---

4.1	Mean coverage (and standard deviations) of the 95% CIs calculated from different conversion equations and for different <code>DESIGNS</code> , and <code>DECIMALS</code> . The closest the values are to the one obtained from raw data, the better. . . . .	55
4.2	Mean coverage (and standard deviations) of the 95% CIs calculated from p-values adjusted using a different <code>CORRECTION</code> . For uncorrected values, the closest the values to the the one obtained from raw data, the better.	56
4.3	Mean type 1 error rate (and standard deviations) of the p-values calculated from the 95% CI obtained using different <code>CI METHODS</code> and varying <code>DECIMALS</code> precision. The closest the values to the the one obtained from raw data, the better. Coverage of the 95% CI provided for reference. . . . .	56
4.4	Mean power (and standard deviations) of the p-values calculated from the 95% CI obtained using different <code>CI METHODS</code> and varying <code>DECIMALS</code> precision. The closest the values to the the one obtained from raw data, the better. Coverage of the 95% CI provided for reference. . . . .	57
5.1	Breakdown of the success rate when comparing the series extracted by participants to the series of the ground-truth data. All values are percentages. <code>CORRECT</code> (✓), <code>INCORRECT</code> (x), <code>MISSING</code> (-), and <code>UNWANTED</code> (+). . . . .	82
5.2	Average relative error of the values obtained from vector or raster charts. Standard deviation shown between parenthesis. . . . .	88

## LIST OF ALGORITHMS

---

1	Generation of data-group suggestions based on prefixes . . . . .	28
2	Extraction of box plots . . . . .	165
3	Extraction of bars . . . . .	166
4	Extraction of scatters . . . . .	166
5	Extraction of axis . . . . .	166
6	Extraction of lines . . . . .	166
7	Split a compound shape . . . . .	167

## INTRODUCTION

---

Documents have been central to archiving and disseminating knowledge. In the digital era, reading these documents happens primarily on screen instead of paper [204, 234, 380], but digital documents have inherited paper limitations. We rarely print documents [233, 380], yet we use file formats designed for print with fixed layouts, styles, and limited support for dynamic and interactive content. We promote open science and ask for trustworthy and reproducible research [111, 264, 266], yet we bury data in documents difficult to parse and read, even by machines. We externalize our thoughts with many kinds of annotations [3, 279, 308], yet we use software limited to capabilities of pen and paper. We learn best with multiple representations of the same information, such as text complemented by figures [46, 244, 253], yet we limit ourselves to the presentation imposed by document authors.

Many ideas have been explored to make digital documents more accessible, engaging, and understandable [163], but it almost always places the burden on authors. Authors have been asked to write clearly; publish in novel digital formats [60, 85, 281]; use complex software or programming languages to design interactive documents [336, 368, 392]; share their data and all material alongside their publications [111, 264, 266]; craft effective information visualizations [43, 254, 292] combined into an convincing data-driven story [321, 345, 367]; produce accessible figures and documents [53, 107, 307]; and stay informed of best practises in terms of content presentation [57, 88, 120]. Few authors can claim to have all these qualities, and even less can put them into practice because they often clash with existing publishing workflows and expectations. But even if authors could do it all, only new digital documents would be improved whereas the billions of documents already published and disseminated would remain as is.

To address these issues, I introduce the idea of *polymorphic documents*. From the Greek, polymorphic refers to having multiple forms. In biology, species that are polymorphic give rise to different forms, all originating from the same genotype. In programming language theory, a polymorphic function adapts to arguments of different types, all from a single definition. Similarly, I argue that a polymorphic document is a document whose content can be turned into different representations, based only on the content and structure of the original document. Polymorphic documents are freed from the constraints of paper, or digital formats that seek to reproduce paper documents. As a result, readers can explore different representations of the information within documents to gain, extract, and disseminate knowledge.

Polymorphic documents give the opportunity to offer representations that leverage well-known cognitive theories and principles. Specifically, I propose representations that are interactive to promote engagement and autonomy ([media richness theory](#) and [self-determination theory](#)) and that can be person-

*This dissertation document illustrates the difficulties imposed by existing workflows: it does not include interactive content mainly because it had to be submitted as a PDF.*

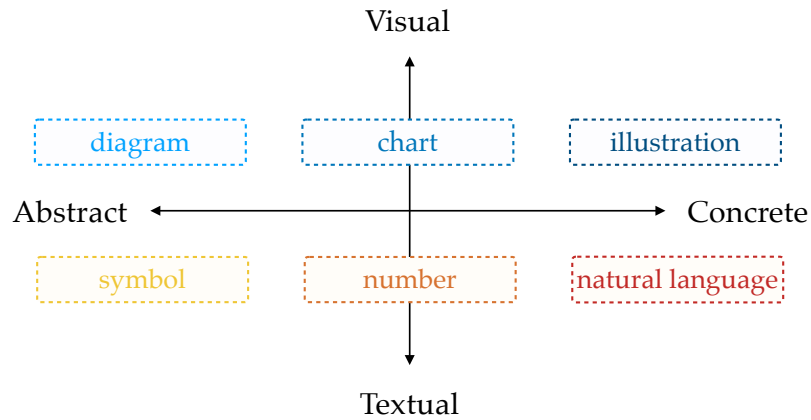


Figure 1.1: Design space of complementary representations. Representations should vary in their media types (vertical) and level of abstraction (horizontal). Dashed boxes are examples of representations commonly found in static digital documents.

alized to tailor the information to readers’ knowledge ([cognitive load theory](#) and [personalized learning](#)). But representations that depict the same information are also most powerful when offering different perspective. Specifically, two different representations are most useful when leveraging different media types such as image and text ([multimedia principle](#) and [dual-coding theory](#)), different density of information ([cognitive load theory](#)), and different levels of abstraction ([ladder of abstraction](#)). When placed on a 2D continuum, this is what I refer to as the “design space of complementary representations” (fig. 1.1). Throughout this thesis, I will place each project within this design space to highlight how the novel representations complement existing ones.

In essence, my research aims to give readers the tools to turn existing digital documents into polymorphic ones. Instead of burdening authors, I believe readers know what they need in documents and can use the tools proposed in this thesis to transform the information in the way they want. Specifically, I propose tools that rely on invariant components of documents such as pixels, text, and features from standardized file formats such as PDF. As a result, these tools are compatible with popular digital formats and preserve the intrinsic qualities of formats, such as being self-contained and ready-to-print. My second objective is to understand how polymorphic documents benefit readers and how readers use them.

### 1.1 RESEARCH OBJECTIVES AND OVERVIEW

Simply put, the high-level research objective of this dissertation is to *improve the reading experience of existing digital documents*. My thesis is that:

*Existing documents can be transformed and enhanced with interactive, visual, and textual representations that help readers understand information.*

*All of the proposed tools are publicly available. Thus, despite the static appearance of this thesis, all figures can be made interactive with Chameleon; all result sections turned into interactive visualizations with Charagraph; all data recovered from the charts using ChartDetective; and all statistics converted into interactive estimation reports with Statslator.*

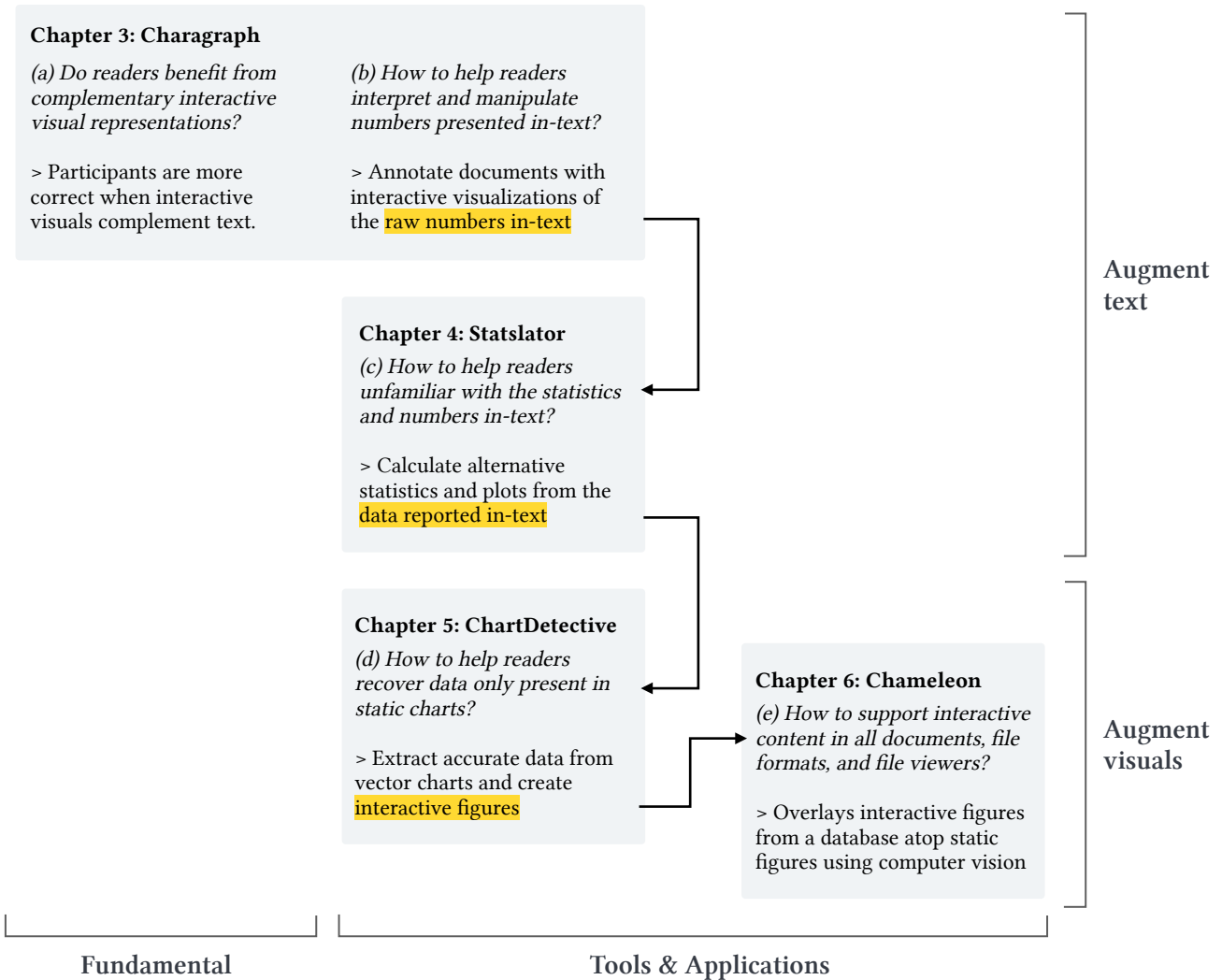


Figure 1.2: Chapters with main research questions (italic), main results, and research path connecting each chapter. Highlighted text and arrows show the dependencies and motivation that lead to the next research question.

To reach this goal, we investigate a set of primary research questions. Note these questions are related and the output of a project often drove the research questions and design decisions of other projects (Figure 1.2). Below are the main research questions investigated.

- (a) *Do readers benefit from complementary interactive visual representations?*
- (b) *How to help readers interpret and manipulate numbers presented in-text?*
- (c) *How to help readers unfamiliar with the statistics and numbers reported in-text?*
- (d) *How to help readers recover data only present in static charts?*
- (e) *How to support interactive content in all documents, file formats, and file viewers?*

*I use 'we' when referring to the projects of this thesis to reflect that this work is the result of collaborative efforts with my supervisors.*



Below, we summarize the steps we took to address these research questions.

1. To answer (a), we conduct a user study in which participants answer comprehension questions after reading a text with and without interactive visualizations. Our results highlight that readers benefit from having interactive visualizations alongside text.
2. To answer (b), we design a tool to morph data-rich paragraphs into interactive visualizations and annotations. These representations leverage the perceptual system to convey numbers and use interactivity to help users explore and manipulate that data. We also conduct a user study to validate that readers can create and customize these interactive visualizations.
3. Results from (a) and (b) apply for numbers that readers are familiar with. To answer (c), we derive a set of equations to convert statistical values into more familiar ones and develop a tool to offer alternative textual and graphical presentations. We validate the equations through simulations using common reporting practices.
4. Results from (a), (b), and (c) depend on the presence of data in-text. To answer (d), we develop a tool to extract data from vector charts. We conduct two studies to verify the usability of the tool with complex charts and the quality of the extracted data.
5. To answer (e), we develop a system that relies on pixels on-screen to detect static figures and overlays interactive content retrieved from an open database. We run simulations to validate the computer vision algorithm. And we conduct a user study to evaluate participants' capability and willingness to use the tool.

## 1.2 CONTRIBUTIONS

*System contributions are often misunderstood. I generally subscribe to James Fogarty's view on the matter [108]*

Four projects explore different aspects of polymorphic documents. Each project introduces a novel system complemented by validations such as simulations and user studies. For each, key results and novel insights are outlined and form our contribution.

### 1.2.1 Annotation of Data-Rich Paragraphs With Charts

We introduce the concept of Charagraphs, which are dynamically generated interactive charts and annotations for in-situ visualization, comparison, and manipulation of numeric data included within paragraphs of text. We contribute a characterization of Charagraphs along three dimensions: the information leveraged, the level of integration, and the interaction contexts.

We contribute a PDF viewer with mechanisms to annotate existing documents with Charagraphs and propose algorithms to perform data extraction.

A user study found participants were more correct with Charagraphs than with text alone when answering comprehension questions about text excerpts.

### 1.2.2 *Translation of Statistics Reporting Styles*

We propose to interactively translate statistical reports in existing documents, from one style such as NHST, to another such as estimation. We report an analysis of CHI papers to demonstrate conversion feasibility and common statistical reporting practices.

A set of conversion equations are derived to perform conversions between p-values, confidence intervals, and various effect sizes. Three Monte-Carlo simulations showed that these conversions stayed accurate even under common reporting practices.

We develop Statslator, a tool to convert between statistical presentations in existing papers. We also contribute methods to extract statistical information using a large language model, error mitigation features, and statistical checking to detect reporting errors in existing scientific documents.

### 1.2.3 *Data Extraction from Complex Vector Charts*

We propose to leverage the benefits of the vector format to extract data underlying charts often embedded in documents and shared online. We highlight several advantages of vector graphics over raster images by reviewing the literature and specification of the file formats.

We develop ChartDetective, a tool to extract underlying data from vector charts. The tool contributes novel interactions to extract data by relying on direct manipulation and drag-and-drop; visual filtering mechanisms; realtime reconstruction of interactive charts; and overlay to verify results.

A user study found participants to be accurate when using the tool even when extracting data from complex charts obtained from scientific documents. A technical validation found the data obtained using our tool to be of higher quality than the data that could be obtained when working with raster images.

### 1.2.4 *Bringing Interactivity to Static Digital Documents*

We propose to use the pixels as invariant features to overlay interactive content atop existing documents, even after their dissemination, and independently of the file format or document viewer used. We review and evaluate the accuracy of various computer vision algorithms that were previously only used to match natural images. Our simulation reports which algorithms worked best for figures commonly found in documents.

We develop Chameleon to augment existing documents with a registration mechanism to add interactive content to an open database that can be shared with others; and an overlaying mechanism with novel detection features of open documents, analysis of pixels currently displayed, and overlaying of HTML views kept in-sync with the document while being browsed.

A first-use user study validated our design choices and the usefulness of the tool compared to existing approaches.

### 1.3 DISSERTATION OUTLINE

The remainder of this thesis is organized as follows (see also fig. 1.2).

*In chapter 2*, we review the literature on current practices and tools used while reading documents. We also review cognitive psychology theories that underpin the design of the representations presented throughout this thesis.

*In chapter 3*, we introduce the concept of charts to annotate data-rich paragraphs, a system to create these annotations, and a user study assessing usability and utility.

*In chapter 4*, we describe a system to translate between statistical reporting styles that we validate with an analysis of scientific documents to assess feasibility, and simulations to confirm accuracy.

*In chapter 5*, we describe a system to extract data from vector charts and two studies to assess usability and accuracy of the system.

*In chapter 6*, we describe a system to overlay interactive content on top of digital documents, simulations to validate the accuracy of the system, and a first-use study to verify usability.

*In chapter 7*, we draw conclusions, summarize limitations, and discuss future opportunities.

Positing that the reading experience can be improved implies that the current reading experience is suboptimal. I believe it to be true because many of the representations used in documents do not leverage our knowledge of cognitive science and how humans read, understand, and learn. There are many reasons for this, most faded away thanks to recent technological progress.

In this section, I review the literature that motivated and informed this thesis. How reading shifted from print to digital devices (section 2.1); that these digital devices can support better representations (section 2.2); that these better representations are currently missing in documents because of file formats and workflows posing challenges (section 2.3); and finally, that these challenges can be addressed through techniques to augment existing documents (section 2.4). Specific related work is reviewed in subsequent chapter.

## 2.1 HOW READERS CONSUME DOCUMENTS

### 2.1.1 *From Paper to Screen*

While recent studies report that the majority of reading happens on-screen [204], in the early days of computers, many experts were convinced that the switch from paper to screen was unlikely to happen [64]. For example, in 2002, Sellen and Harper published the “The Myth of the Paperless Office” in which they argue that offices will remain dependent on paper [346]. At the time, empirical evidence showed that paper was faster to read [84, 135, 206, 268], resulted in higher proofreading accuracy [65, 411], and lower fatigue [75, 407]. Others were more optimistic and claimed that a transition to digital was inevitable [29, 271] and that most of the negative effects of screens on reading will be eliminated with improvements of technology [269].

Today, paper is still present, although the choice between reading on screen or on paper is a matter of preference and context [190] and habits have changed in favour of reading digitally. In 2005, while the majority of students reported an increase in the time devoted to reading electronically, they were still about 89% who preferred printed media versus 3% for electronic media [233]. These results were consistent with a survey in 2006 where students reported frequently reading online 34% of the time [234]. However, this shifted in a 2012 survey where scholars reported that more than half of their reading was on a screen. In fact, only 28% of articles were downloaded to be printed on paper [380]. It is unclear why this transition happened, but possible explanations include easier access to digital resources [214, 234, 381]; a substantial increase of reading done by people making print untenable [233,

*“The shift from print to the computer does not mean the end of literacy itself, but the literacy of print, for electronic technology offers us a new kind of book and new ways to write and read.” – Jay David Bolter (1991)*

*“The question is not can we do everything on screens, but when will we, how will we, and how can we make it great?” – Ted Nelson (1987)*

380, 382]; and the improvement of technologies to make reading at least as efficient as on paper [297, 308, 379].

### 2.1.2 Active Reading Strategies

People adopt different reading strategies depending on their goals [384]. One strategy of particular interest to my work is the idea of “active reading”.

Active reading was first identified with printed documents as “the combination of reading with critical thinking and learning, and involves not just reading per se, but also underlining, highlighting, and commenting.” [3, 308]. Initial efforts to bolster digital active reading focused on bringing the capabilities of pen and paper to screen-based environments [167, 233, 279, 297, 358] and proceeded to extend them through powerful annotation [326, 342], navigation [379], diagramming [370], and note-taking features [160]. Most readers engage in active reading to externalize their thoughts [370], reduce cognitive load [341], improve their understanding [322], emphasize key concepts [3], help memorization [265], aid in later retrieval [342] and share their marks with others [378].

*“An active reader doesn’t passively sponge up information, but uses the author’s argument as a springboard for critical thought and deep understanding.”  
– Bret Victor (2011)*

Since then, the definition of active reading has evolved. Bret Victor introduced Explorable Explanations to “encourage truly active reading” and turn text into “an environment to think in” instead of “information to be consumed” [36]. Victor’s definition of active reading focuses on interactive documents to explore, build intuition, make discoveries, and reinforce or challenge knowledge. This idea has been applied to contexts such as transparent statistical reports [91] and visual exploration [125] (see Hohman et al. [163] for a review).

Both definitions express similar objectives and are used interchangeably in the literature, but differ in the role of the reader. In Victor’s definition, readers engage with interactive elements. In the original definition, readers create their own marks, an act that may be as important, if not more [192, 263, 343], than the marks left on the document [3, 325, 370, 413].

*Throughout this thesis, I refer to active reading as a combination of annotating and interacting with documents, as is the usage in HCI.*

These active reading ideas inspired the design of polymorphic documents presented here. In particular, annotating documents using Charagraphs and Statslator prompts the reader to think critically and learn, following the original definition of active reading. But Victor’s definition is also supported: the created annotations are interactive for readers to explore, make discoveries, and reinforce their knowledge about the information.

## 2.2 BENEFITS OF POLYMORPHIC CONTENT

Since reading with digital devices enables new representations, the question becomes which ones might be most beneficial. Below, I discuss the different ways these representations can vary and benefit readers. For each, I review relevant theories and describe how they are leveraged by the representations proposed in this thesis.

*While I briefly discuss several theories, they are generally supported by substantial empirical evidence and widely accepted within the scientific community.*

### 2.2.1 *Benefits of Interactivity*

I define interactivity as the ability for readers to actively engage with a representation and manipulate it in some capacity. Making a representation interactive may benefit readers in two ways.

*Making a representation interactive can enrich it.* And a richer media may help communication accuracy, speed, and readers' satisfaction. This idea has been introduced by Daft and Robert in their [media richness theory](#) [77]. The theory states that messages of varying complexity require media of varying level of richness. For example, a complex novel message (referred to as "nonroutine" by Daft and Robert) like the ones found in scientific documents will require a richer medium that can convey more information and facilitate rapid feedback to avoid misunderstandings. Interactive representations embedding additional information are, by definition, a richer medium because "*The more learning that can be pumped through a medium, the richer the medium.*" [222].

*Interactive representations can promote engagement.* Interactivity gives readers choices and lets them act autonomously. This touches upon the [self-determination theory](#) that identifies autonomy as a key component of [intrinsic motivation](#) as opposed to [extrinsic motivation](#) driven by external goals or rewards [332]. Intrinsic motivation is generally preferable as it may lead to a state of complete focus, immersion, and enjoyment of the task [66].

In practice, many of these ideas are reflected in the notion of [active reading](#) proposed by Victor (section 2.1.2) and engagement has been a strong motivation to include explorable explanations in documents [163]. But interactivity is also frequently used to offer additional information and let readers go on tangent stories [251, 293].

The polymorphic representations presented in this thesis all include some form of interactivity: Charagraphs are visualizations with interactions to support data exploration tasks and allow readers to create their own data-driven story differing from the canonical one; Statslator generates explorable explanations of statistics; ChartDetective reconstructs interactive versions of static charts; and Chameleon replaces static figures by interactive ones.

### 2.2.2 *Benefits of Adaptation*

I define adaptation as the ability for a representation to change depending on the reader's knowledge, such as using terms, values, and visuals that are familiar to the reader.

*Representations tailored to the reader may reduce extraneous load.* It allows the reader to focus on the complexity of the information itself ([intrinsic load](#)) rather than the complexity induced by its presentation ([extraneous load](#)). This distinction is described by the [cognitive load theory](#) which also explains how [extraneous load](#) impedes learning and why it should be reduced as much as possible [48]. This idea powers concepts such as [personalized learning](#) that provide students with a learning experience tailored to their strengths, needs, and interests [40].

In practice, adaptation is sometimes used by news articles to provide reliable information such as checking the temperature difference in the reader’s hometown [305] and checking if the readers’ job will be automatized [404]. It has also been used to provide readers with familiar analogies when a document mentions physical measurements such as distances [171, 197]

In this thesis, Charagraphs use adaptation to let readers customize the appearance of charts. Similarly, Statslator converts statistical values and provides readers with informationally equivalent options to choose from.

### 2.2.3 Benefits of Varying the Type of Media

Two representations can present the same information (informational equivalence) yet differ in how easy that information is to extract (if no computational equivalence) [357]. This is typically the case for representations that differ in the type of media. For example, even though both representations convey the same information, it is much easier to identify a trend using a line chart rather than a table. Below, I outline two main benefits of presenting information both verbally and using imagery.

*Text and images may reduce extraneous cognitive load and help comprehension.* Specifically, the [dual-coding theory](#) suggests that people mentally represent information verbally and visually through two interconnected but distinct systems [289]. Thus, presenting both text and images allows activating both systems and drawing connections between them to enhance understanding.

*Text and images may improve recall.* The [dual-coding theory](#) explains that both text and image may facilitate memorization because the information is then encoded twice, with both the verbal and visual system [289]. Specifically, experiments testing the “[multimedia principle](#)” observed that information is better learned when conveyed through text and related images [253].

One case of polymorphic content that has been particularly well documented concerns numerical values. Experts have been debating for a century about the best representation of numeric information: inline with sentence text, in a table, or in some form of graphical visualization [63, 402]. It has long been suggested that these representations are not equivalent and should be chosen carefully [14, 101, 120, 176, 180, 213, 235, 242]. Data visualizations are often the best choice because they offload cognitive work to the perceptual system [213, 267]. But an even better choice is to not choose and embed multiple representations, given that space, time, and expertise permit such a report [57, 120, 176, 187].

In this thesis, the proposed representations enhance existing content by offering a different type of media: Charagraph turns text into data visualizations; Statslator turns written statistics into plots; and ChartDetective turns charts into numbers.

*“Words and pictures  
belong together.  
Viewers need the help  
that words can  
provide.” – Edward  
Tufte (1986)*

### 2.2.4 *Benefits of Varying the Level of Abstraction*

Another way a representation may vary is in its use of abstract and concrete information. Samuel I. Hayakawa proposed to use a metaphorical ladder to discuss these different abstraction levels [152]. At the bottom of the ladder are concrete details and specific examples that typically include sensory information. As we move up the ladder, the information becomes more abstract, losing sensory information but becoming more general and applying to a broader range of examples. This was initially discussed for words but is trivially extended to images. For example, a photograph is more concrete than a diagram. Offering multiple level of abstractions has two main benefits.

*Abstract and concrete information may help memorization.* The [dual-coding theory](#) suggests it as another way the mind dually encodes information [289]: abstract information will use the verbal system, whereas concrete information may be easier to represent using mental imagery and be encoded using the visual system. Experimentally, the theory has been supported through several studies that found people can remember concrete words better than abstract ones [290, 291].

*Abstract and concrete information may improve understanding.* Specifically, the action of moving up and down the [ladder of abstraction](#) with multiple representations may benefit readers. Specific and concrete information is typically easier to understand. Thus, making it more abstract may help clarify the underlying concept. Inversely, an abstract idea may be easier to understand using concrete examples. A possible explanation is the reduced [intrinsic load](#) that concrete representations incur compared to abstract ones that might use complex terms.

Many systems have leveraged this idea of [ladder of abstraction](#). For example, Victor proposed to analyze a system with emergent behaviours by starting with the concrete example of a car simulation and then moving up the ladder by abstracting parameters of the simulation such as time [391]. Others have considered moving down the [ladder of abstraction](#) to help with abstract and notoriously difficult concepts such as mathematical notations [286, 351]. In “Kill Math”, Victor argues for an alternative to symbolic math that is more concrete and intuitive [394]. Victor presents four demonstrations using visual representation such as plots, concrete objects instead of symbolic variables, and interactive widgets. Similarly, ARMath is an application to help children learn and discover mathematical concepts using everyday objects [186]. But Noyon [334] might be the system that most embraced the idea pushed forward by Victor. Noyon is a system to represent mathematical expressions through sketches and allows users to go up and down the [ladder of abstraction](#) from symbolic (math notation) to iconic (sketches).

The representations proposed in this thesis allow readers to move between levels of abstraction: Charagraph allows moving up the ladder by abstracting text into a chart and Statslator allows moving down the ladder by converting statistics into concrete values that readers are more likely to understand.

*“I believe that a person should not have to imagine the interpretation of abstract symbols. Instead, dynamic graphs, diagrams, visual models, and visual effects should provide visceral representations.” – Bret Victor (2011)*



## 2.3 CHANGING WORKFLOWS, FILE FORMATS, AND SOFTWARE

Despite the many ways representations can be made more powerful, several issues prevent their adoption. Many have discussed these issues and the opportunities created by better formats, from the field of Astronomy [131] to Biology [15, 298], including news articles [133, 323], and of course, Computer Science [139]. We categorize the main issues according to their root cause and discuss the solutions that different communities have proposed.

*“PDF is skeuomorphic, intended to carry the character of an old entity into a new one.*

*It is designed to produce an exact replica of a printed document. Great if you want printed documents, terrible if you don’t.” – Rupert Goodwins (2022)*

*Digital Formats Designed For Print:* Digital formats were traditionally designed to emulate paper. In fact, PDF, the most popular digital document format, was created to replace FAX machines, and its purpose remains to print identically across computers and printers [401]. Because these formats focus on rendering identically across machines while remaining lightweight, they tend to have limited features and to preserve little structure and meta-data, making them hard to use for anything other than rendering and printing. This is in contradiction with the desire of many communities to link data and code with the document [131]; offer audio, video, and interactive content [139]; and make documents “digestible” by machines [298].

*Limited Software Support:* Even when file formats support more dynamic and interactive content, the software prevent their presentation. Again, the PDF is a perfect illustration. In 2008, Adobe added support for Flash content within PDF as part of ISO 32000-1:2008 and immediately released a version of Adobe Acrobat Reader with the capabilities to read such content. While the process to include Flash content was laborious for authors [139], the bigger issue is that despite PDF being a universal format, only Adobe Acrobat Reader supported the feature. Thus, tablets and in-browser PDF readers could not read Flash content. Worse, Flash was discontinued in 2020 and the new ISO 32000-2:2020 removed the feature. The few documents integrating Flash (e.g., [121, 189]) are now permanently static, even with Adobe Acrobat Reader.

*“There continues to be a market for PDF-based document workflow systems that demand PDF inputs and produce PDF outputs, no matter how much the humans at either end grind their teeth in frustration” – Rupert Goodwins (2022)*

*Incompatible Workflows:* Perhaps the most pernicious issue is not technical but behavioural. Historically, publishers printed all manuscripts to be available in physical libraries. The publishing workflow was designed to create this paper final product and in turn participated in the success of PDF over other formats such as HTML [139]. Today, virtually all publishers have stopped printing proceedings, yet they still use the same workflows. Some publishers still require figures to be greyscale or to be included separately (e.g., Brill [37]), and a PDF version of the paper is often the only accepted format. As a result, people and publishers tailored their pipelines, editing technologies, authoring software, and document viewers to these workflows and are unlikely to change them.

## 2.3.1 Tentative Solutions to Support Augmented Documents

Most solutions to have documents freed from the limitations of paper rely on the adoption of a new file format. Consider HTML documents, a solution seldom adopted by the Computer Science community [60, 85]. One reason

that PDF persists as a format for document distribution is that it supports the flexible dissemination of documents by combining content and visual formatting in a compact format. HTML rarely achieves this simplicity of presentation: new devices and browsers frequently render aspects of a document idiosyncratically, which can make reading an uncomfortable experience. However, even if we advocate moving to a new format, this means that our extensive archive of existing documents must either be translated and re-imagined with interactive content, or must be maintained in the original non-interactive format.

The other solution is to create new kinds of documents viewers [17]. For example, Utopia Documents [15] is a novel PDF viewer integrating visualization and data-analysis tools. By inspecting the content and structure of PDF files, this viewer is able to integrate dynamic objects and annotations in documents, such as visualizing the 3D structure of protein sequences. Of note, the Biochemical Journal started using Utopia Document in 2009 [298]. However, the website hosting the document viewer is unavailable today. This solution is also limited to specific file formats and forces readers to change their habits, adopt a novel document reader, and potentially lose features.

These issues have been central in the design of the solution proposed in this thesis. For example, Chameleon can incorporate dynamic content within the confines of existing static documents and without any modification of existing document or viewing applications. And the other systems we propose are all readily available and all working with PDF documents.

#### 2.4 EMPOWERING READERS BY AUGMENTING EXISTING DOCUMENTS

Data-rich documents are notoriously challenging to write [51] and visualizations are difficult to create [57, 120, 187]. Systems have been proposed to help authors write better documents [51] and create better data visualizations [68, 169, 238, 353, 362, 410]. However, the created documents remain static which limits the exploration possibilities for readers [36], and forces a split of attention between text and figures [16]. Other systems have targeted the design of interactive documents by offering ways to author them [60] and help craft data-driven stories [217, 251, 374], but these require a substantial amount of work for authors and are often incompatible with PDF-based workflows such as academic publishing.

An alternate approach adopted by the systems proposed in this thesis is to augment existing documents using semi-automatic systems. Previous systems have been used to direct the reader's attention to specific sentences [52, 109, 403, 414], provide definitions of unknown terms and symbols [153], integrate relevant information and commentary from follow-up work [315], and generate summaries [369]. Some systems help readers interpret and manipulate the results by enhancing the visualizations already present in the document. For example, redesigning charts to be more useful [340], adding overlays [201], and introducing interactive features [237]. Or the text can be leveraged to automatically annotate existing charts [170, 209] and interactive

links that connect text and charts [300]. However, all of these approaches assume the author originally included visualizations in the document.

Another idea explored by Elastic Documents is to generate new chart visualizations from tables in a preprocessing data extraction step performed on the entire document [17]. Data in tables are extracted and used to generate different static chart visualizations. Readers can filter and browse the generated charts or select a sentence in the document and let the system show the most relevant charts. The relevance is calculated by matching the keywords from the sentence and the table headers.

The projects presented in this thesis all strive to unburden authors by giving readers the tools to change the representation of the knowledge within documents. In particular, Charagraph and Statslator are similar to Elastic Documents in that they are new visualizations complementing existing documents. However, these are annotations dynamically generated by readers and for information contained within text or math equations. This makes these annotations highly customized by the reader for their needs and tightly coupled with the text or equation through bidirectional interactive features.

# 3

## INTERACTIVE GENERATION OF CHARTS FOR REALTIME ANNOTATION OF DATA-RICH PARAGRAPHS

---

In this chapter, we propose to leverage the benefits of varying the levels of abstraction ([ladder of abstraction](#)) and the types of media to help readers interpret numbers in-text ([multimedia principle](#)).

### 3.1 MOTIVATION

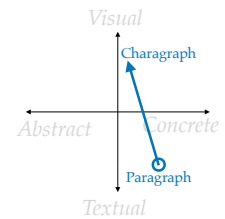
Documents and scientific reports are often filled with numerical values. These regularly interrupt the flow of sentences with quantities like means, medians, confidence intervals, and p-values. While these are important to support claims, they can make it hard to compare and interpret patterns in numeric data due to the linear structure of sentences.

In fact, the use of sentences to present numbers has long been criticized [101, 198, 402]. Edward Tufte [63] notes that: “*The conventional sentence is a poor way to show more than two numbers because it prevents comparisons within the data*”.

While experts almost always recommend visualizations such as charts to present numeric data [7, 63, 122, 191, 402], they may not always be included or useful. Authors may omit a visualization because it takes time to create [187], occupies space in the document [7, 142, 176, 191], requires substantial effort and skill [57, 120, 187], or seems unnecessary to present a few numbers [93, 122, 191]. Worse, even when authors include visualizations, readers might ignore them if they are poorly designed or not aligned with the reader’s goal [267], overblown [385], or not accessible [194, 412]. And readers might struggle to connect text and figures because the layout of the document often separates them [16, 63].

One response to issues when data is included in text has been to provide interactive connections between text and existing visualizations in the document. Because archival documents are not easily modified once disseminated [247], most proposed solutions use some form of automatic linking [170, 207, 209, 257, 300] or an authoring system to create dynamic interactive documents from the beginning [203, 215–217, 374]. However, these approaches remain dependent on the author’s willingness to embed figures. Another strategy proposed by Elastic Documents [17] is to preprocess a document to generate new visualizations from the tables and then include ways to filter them while reading. But this requires inferring what readers need a priori and assumes that the data is presented in structured tables. What if, as is often the case, the data exists only in sentences?

We introduce the concept of *Charagraphs*, which are dynamically generated interactive charts and annotations for in-situ visualization, comparison, and manipulation of numeric data included within paragraphs of text. We explore the design of Charagraphs and contribute a system that allows readers to



In our design space fig. 1.1, Charagraph transforms concrete textual paragraphs into visual charts and annotations

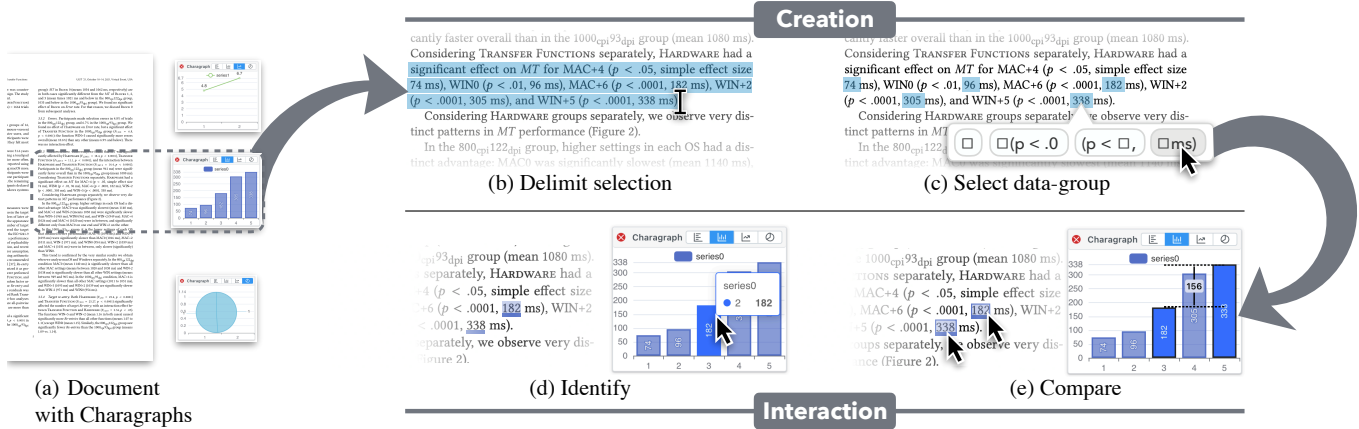


Figure 3.1: (a) Charagraphs are in-situ visualizations of numeric data included within text that are dynamically generated (b) by delimiting a selection and (c) selecting a data group. Charagraphs support common data exploration tasks through interactive features such as (d) identifying and (e) comparing values.

quickly generate them from existing documents. Instead of expecting authors to embed well-designed visualizations, our system empowers readers to quickly craft their own visualizations, customized to their needs, from the data presented in texts. Although our system can also extract data from document tables, we focus on the problem of extracting data from sentences. The created Charagraphs use data visualizations to offload cognitive work to the perceptual system [213, 267] alongside interactive features to compare and manipulate the data. This can help when data is presented only in-text, or when the existing static visualizations are ineffective and readers would prefer a different representation. In a user study, participants created their own Charagraphs to annotate documents and answer questions. All participants found the system easy to use and were able to create the visualization they had in mind. Furthermore, all participants preferred to have Charagraphs available; had lower mental demand, effort, and frustration; and were more correct in completing comprehension, comparison, and interpretation tasks when they had access to Charagraphs (98.61%) compared to text alone (92.59%).

### 3.2 BACKGROUND AND RELATED WORK

We first examine what numeric data can be expected in text and then review the different ways of presenting data and their advantages and drawbacks depending on the context. We end by discussing previous work turning text into visualizations.

#### 3.2.1 What Values Are Reported In-Text?

Most documents report numerical values in text to support claims because “a stronger quantitative result is more informative than a weaker, qualitative result that

*subsumes it*" [104]. For example, statistical results are often reported alongside means or mean differences [103, 104]. Practises vary across domains and publishers.

The American Psychological Association (APA) style instructs authors to *"emphasize particular data in the text when they aid in the interpretation of the findings"* and when reporting inferential statistics, the data in text *"should allow readers to confirm the basic reported analyses (e.g., cell means, standard deviations, sample sizes, correlations)"* [14]. In fact, the APA style normalizes this process by presenting a list of statistical abbreviations and symbols to be used in text (e.g., "M" for "mean", "Mdn" for "median").

Within the HCI community, recommendations for transparent statistics call for quantitative results to provide effect sizes (including the direction of the difference, the unit, and the type of estimate and uncertainty) [142] and interval estimates [88]. These values are easier to interpret and compare graphically [88, 104, 235], but textual reports are commonly chosen as an alternative because *"space may not always permit a graphical report"* [142].

As a result, scientific documents often report numerical data in a semi-structured way using specific symbols and patterns.

### 3.2.2 How to Best Present Numerical Values?

There are three ways to include numerical values in documents: inline with sentence text, in a table, and in some form of graphical visualization [63, 402]. It has long been suggested that these representations are not equivalent and should be chosen carefully [14, 101, 120, 176, 180, 213, 235, 242]. We review two relevant situations.

*When there are few values:* While visualization experts generally agree that *"under no circumstance should text be used by itself to convey important statistical data if more than a very few facts are to be presented"* [101] (also [63, 402]), they are divided on which representation to use when faced with small datasets. Some state that graphical visualizations with little content indicate a lack of purpose [93], take up a lot of space [7, 176, 191], add to readers' cognitive load [14, 122], and are costly to create [187]. Yet, experimental evidence suggests a general advantage for visualizations such as bar charts, even when presenting only two values [89, 195, 359].

*When presenting the same information:* Two different representations of the same information are often beneficial [213, 344, 357]. For example, Ottley et al. found that visualizations make it easy to identify critical information, but the precise numerical value was easier to extract from text [287]. Early on, Tufte advocated for the combination of words, numbers, and graphics and deplored the segregation between figures and their associated text in scientific manuscripts [63]. Experimentally, the [multimedia principle](#) shows that people learn better when an image complements text [253] and strong evidence suggests that illustrated texts improve reading comprehension and recall [92, 124]. The general recommendation is that authors should use text reinforced by graphics [101].

*Charts are a good empirically tested example of the "benefits of varying of the type of media".*

### 3.2.3 Issues With In-Text Data and Rationale for Reader-Generated Interactive Charts

*Here, we focus on the issues related to including static charts. But, as previously mentioned, there are additional issues when including interactive content such as Charagraphs.*

Data is usually reported in text, and at best presented visually in static charts, which remains far from ideal for readers.

First, *textual data is difficult to interpret*. Numeric values make it hard to perform analytic tasks such as identifying trends [259] and are presented linearly in sentences broken at arbitrary points depending on document layout [385]. Despite the advantages of data visualizations, a document might present data only in text or tables. In contrast, visualizations help understand the information [259, 360], discover facts and insights about the data [198, 267, 274] and perform general interpretive operations such as locating, deriving, interpolating, and comparing values [47, 69, 101, 259]. As such, allowing readers to generate visualizations would be beneficial.

Second, *data is segregated in documents* and it is often tedious for readers to combine and visualize the results reported in different documents or different parts of the same document [278], because documents segregate the information and the data, even when related [385]. If readers can generate charts from the text, then they could do it by combining two distinct parts of a document.

Third, *textual report of results might be misinterpreted*. Instead of using dichotomous statistical testing such as thresholds on p-values, it has been suggested that reporting interval estimates such as confidence intervals could reduce misinterpretations and improve understanding of results [88, 142]. Yet, these estimates are often best conveyed graphically [156, 235]. For example, readers will have difficulties deciding if two confidence intervals reported in text overlap, and by how much (e.g., how different is 95% CI [-0.08, 0.5] from 95% CI [0.1, 0.4]?).

We posit that the above mentioned issues could be alleviated by supporting the dynamic generation of interactive charts and annotations generated by the reader, in order to offer in-situ visualization, comparison, and manipulation of numeric data included within paragraphs of text. In the remainder of this document, we refer to such interactive charts generated by the reader as Charagraphs.

### 3.2.4 Generating Visualizations from Text

Several tools propose methods to transform high-level textual descriptions into data visualizations [117, 239, 270, 347] and infographics [67, 310] (see Shen et al. [349] for a survey). However, these approaches differ from Charagraphs in that they are essentially visualization authoring tools with natural language as the input modality. Thus, they require a curated dataset provided by the author and use short queries specifically crafted to generate a visualization from the dataset by referencing metadata like column headers (e.g., “How much do various cars weigh?” or “draw a line chart of daily sales forecasts” [363]).

Closer to our approach are methods to augment documents with visualizations. Essentially the content of the document forms a query to generate a visualization using an associated dataset. For example, Kim et al. help readers relate to distances and areas mentioned in text by re-expressing them on a map and using familiar landmarks [197]. Hullman et al. generalized this idea to help readers understand physical measurements such as weights, lengths, and volumes with different strategies to re-express them in a relatable way such as comparing them with familiar objects [171]. An optimization method and a large dataset of familiar objects is used to generate re-expressions using both text and images. Similarly, Contextifier [170] and NewsViews [118] generate contextual visualizations when reading news articles. Both systems pre-process the document text and search for relevant data in existing external databases. NewsViews generates interactive annotated maps using commonly available data like unemployment and education statistics. Contextifier generates stock charts annotated with news article content using commonly available stock data.

In contrast, Charagraphs are generated by readers using only data contained within the text of a document. Using an external database is not necessary, and may not even be possible since text data within scientific documents may not be easily accessed or even available in common databases. In addition, Charagraphs support many forms of chart visualizations, and are designed to be interactively customized with strong linkages with the text data.

### 3.3 CHARACTERIZATION OF CHARAGRAPH

A Charagraph is a visualization that augments in-text data: this visualization comprises text annotations overlaying the document and a chart. Both the text annotations and the chart are interactive and in-sync: modifications and interactions done on one representation are reflected on the other.

We characterize a Charagraph along three dimensions: 1) what in-text *information* can be leveraged; 2) how to *present* the textual and graphical representation; and 3) what *interactions* can be performed with both representations.

#### 3.3.1 Information Leveraged

Designing an effective visualization often requires understanding the data, its provenance, and its meaning [267]. Fortunately, paragraphs are semantically rich. Beyond numbers, the structure and formatting of sentences, the words, the style, and the suffix or prefix used with numbers provide valuable information about the data. We consider the different information that can be extracted from the text and their impact on the graphical representation encodings (marks) [26], embellishment (visual channels) [20], and meta-data (titles, labels, and legend).

- **Data** corresponds to the values reported in text that are of interest to the reader (Figure 3.2a). These are often numerical, but could be categorical



such as country names. Without additional information, meaning can be inferred. For example, numbers adding up to a 100 are likely proportions.

- **Statistical terms** explain *how* the data was obtained. For example, how the data was aggregated or what transformation was applied to it. These terms generally precede values (e.g.,  $M=12$ ,  $SD=1.2$  or 95% CI [1, 2]), and can inform the choice of encodings. For example, error bars for standard deviation (Figure 3.2b).
- **Units** give general information on *what* the data is. They generally follow the values (e.g., 25%, 33 years, 8°C) and might inform the choice of encodings. For example, a percent suggests a pie chart whereas a timeline may be best for years. Units can also suggest embellishment, such as representing temperature data as bars styled as thermometer (Figure 3.2c).
- **Labels** describe the context of the data in the document (Figure 3.2d). Their placement relative to the data depends on the sentence structure (e.g., the first day of May [...] was 15°C). It can serve as meta-data to label data points, legends, and axes and also change embellishment. For example, colouring a series blue if nearby text refers to an ocean.
- **Semantics** correspond to *why* the data was included in the document (e.g., it dropped from [data] to [data]). Semantic-aware Charagraphs might change the encodings depending on the reason; a line chart better represents a sentence describing a trend, whereas a bar chart better represents a comparison (Figure 3.2d).

### 3.3.2 Level of Integration

The presentation of the textual and graphical forms offers one dimension to control the narration. Mandl and Levin note that “*the medium processed in the first place can elicit an attitude, which determines the processing of the other medium*” [244]. The prominence of each representation can be manipulated to orient readers toward a more controlled (text-driven or author-driven) or exploratory (visualization-driven or reader-driven) narrative [345]. Figure 3-3 lists the different levels of integration of the graphical and textual information, going from only text to only charts.

- **Chart in Text** puts the emphasis on the textual representation and follows the narration of the document. A Charagraph can achieve this by

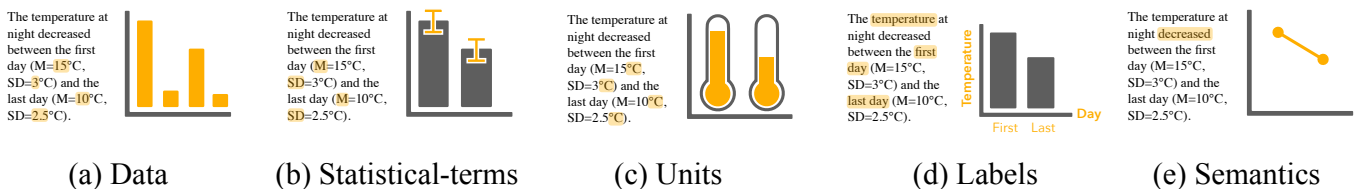


Figure 3.2: Information that can be leveraged from the text with examples of the design of the Charagraph.

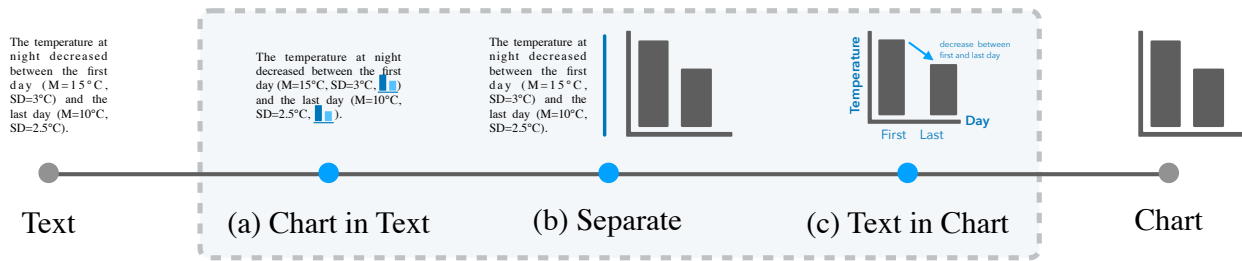


Figure 3.3: Level of integration of the chart and text representations. Dashed outline indicates levels specific to Charagraphs.

using word-scale visualizations [126, 128, 385] (also called micro visualizations [294] or word-sized graphics [24, 385]) that are known to help readers retain information while reading [127] and also support interaction [129, 130, 215] (Figure 3.3a).

- **Separate** lets readers decide which representation best suits their needs. A Charagraph achieves this by having both representations equally visible, shown side-by-side in their own separate views (Figure 3.3b).
- **Text in Chart** commonly referred to as an “annotated chart” puts the emphasis on the graphical representation. Readers are less guided than with a strict sentence order (Figure 3.3c). Annotations are a common device to support data-driven story-telling [321]. Similarly, a Charagraph achieves this by presenting the chart visualization with textual annotations to emphasize a few elements (see Ren et al. [318] for a review of annotation types).

### 3.3.3 Interaction Contexts

A Charagraph acts as a static visualization when not being interacted with. Otherwise, it has two interaction contexts based on the position of the pointer: in-text and in-chart. Previous work on reading behaviours with interactive documents [61, 125, 420] suggests that readers use in-text interactions to follow the document narrative and in-chart interactions to explore and engage with the data. To connect the representations, feedback is synchronized between text and visualizations.

- **In-Text** interaction is most likely during close-reading, when sentences are read carefully one after the other. As such, the interaction should be limited to connecting text to a visualization to support the narrative rather than distract from it. For example, the pointer position can serve as proxy for reading position so only values included in the current sentence under the pointer are highlighted with values before or after the pointer position de-emphasized or hidden. Figure 3.4a illustrates the minimal feedback in text as a result of the synchronization with the chart.

- **In-Chart** interaction indicates that readers are going beyond the document narrative. As such, the interaction should apply to all the data from the Charagraph, and can support common data exploration tasks such as comparing, filtering, sorting, and summarizing [10] (Figure 3.4b).

### 3.4 CHARAGRAPH DOCUMENT VIEWER

We built a system to generate Charagraphs inside a document viewer. We focus on giving readers tools to create and customize Charagraphs instead of fully automating their creation. We argue that, akin to **active reading**, the creation process is part of sensemaking. By enabling readers to select and decide on the best visualization, they actively participate in understanding the text and the data. From our characterization, we focus on the “Separate” level of integration as it offers the maximum bandwidth of information for both representation and let readers decide on the representation to use. Similarly, the system only leverages information in “Data”, “Statistical-terms” and “Units” and gives the tools to customize the Charagraph further. A live version of our system is accessible online: <http://ns.inria.fr/loki/charagraph>

#### 3.4.1 Select In-Text Data

A reader who wants to visualize part of data included in text first needs to delimit where it is located in the document and then specify the data of interest through a “data-group menu”.

##### 3.4.1.1 Delimit a Selection

There are two mechanisms to let readers delimit a selection before prompting the data-group menu.

- *Traditional Text Selection* use standard text segmentation methods such as dragging to have a selection snapping to letters and triple-clicking to select a whole sentence.

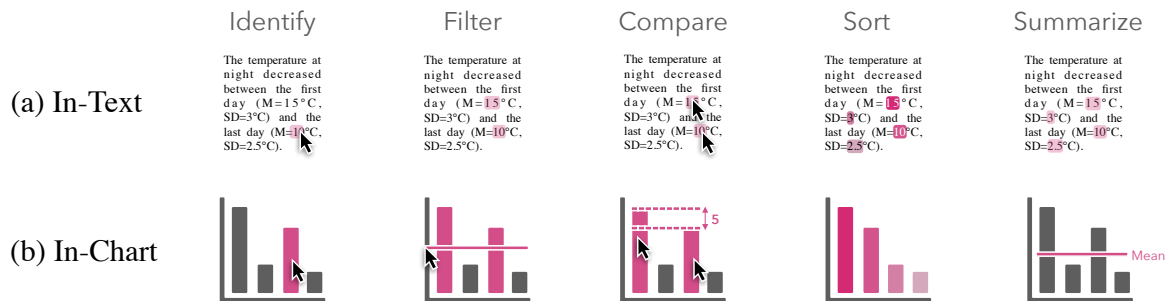


Figure 3.4: Interaction contexts for different data exploration tasks.

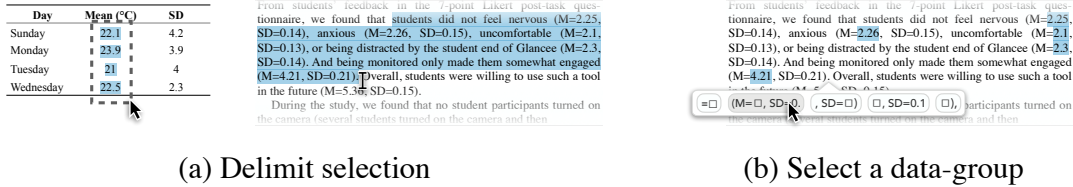


Figure 3.5: Selecting data to visualize: (a) delimiting using traditional text select or marquee selection; (b) selecting one of the data groups suggested by the system.

- *Rectangular Marquee Selection* allows readers to select all text contained within a rectangle formed by the selection. This tries to address the limitations of traditional text selection caused by small motor errors [22], snapping mechanisms [49], and unrelated elements that are selected in documents that do not preserve sentence-structures such as PDF. For example, selecting the column of a table in a PDF often selects other columns or entire rows because the selection is designed for sentences.

To distinguish between selection methods, a marquee selection must be started outside the text. In both cases, readers only need to select a text portion containing the data of interest to guide the system and reduce the number of data-group suggestions.

#### 3.4.1.2 Choose a Data-Group

A data-group is a group of values deemed semantically similar by the system. Because data included in text can take various forms and may be intertwined with other information, readers can refine their selections using system-generated suggestions of data-groups.

When a portion of text containing data is selected, a data-group menu is displayed below the selected portion (Figure 3.5b). The groups are sorted based on the number of values they contain. The first group always contains all the values in the selected portion, whereas the other groups are formed based on the text preceding and following the values. This allows to form groups based on the “statistical-terms” and “units” as per our characterization (Section 3.3) and potentially also works when the sentence use a similar structure to introduce numbers. For example, text reporting mean and standard deviation in APA-style will have at least two data groups for values preceded by “M=” and “SD=” (Figure 3.5). Similarly, text reporting the evolution of a statistic over time will often use prepositions such as “from” and “to” that would distinguish them.

Options in the menu are named after their preceding and following text, and the position of the value is represented by a square. Readers can preview the selection by hovering over each option. When the cursor is placed above an option, the corresponding numeric values in the text are highlighted (Figure 3.5b). Clicking an option immediately creates a Charagraph from the selected values.

## CHARAGRAPH

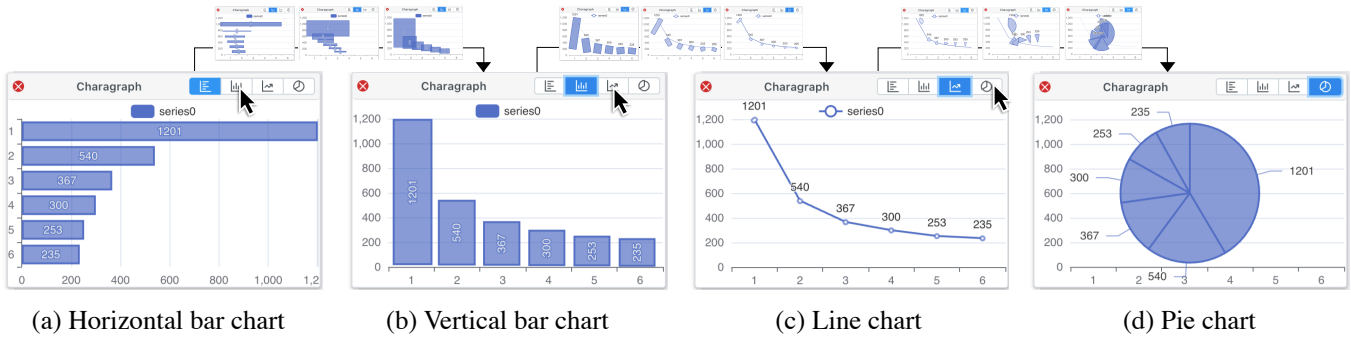


Figure 3.6: Changing visualizations. All transitions are animated to keep track of changes.

### 3.4.1.3 Edit the Selection

After a Charagraph is created, readers can add or remove values to further refine the selection. A right-click on one of the underlined values in the text opens a context menu with the option to remove the value. Right-clicking a data point in the chart opens the same context menu. A right-click on a value not currently included opens a context menu with the option to add the value to the current Charagraph.

### 3.4.2 Customize the Visualization

By default, newly created Charagraphs appear in the margin, show values with vertical bars, and have numbered tick and legend labels. Readers can change those defaults to customize the visualization. Similarly, all Charagraphs can be resized and moved using direct dragging manipulations.

#### 3.4.2.1 Change Visualization

*Allowing readers to customize the visualization relates to the benefits of adaptation and may help reduce extraneous load.*

Readers can choose visualizations more suited to their task [333] or the data [313]. The system supports popular chart types [21] and related visualizations that can be generated from data typically found in-text: horizontal and vertical bar charts for aggregated quantities, such as means, and for comparison between data series; line charts to visualize the evolution of a measurement; and pie charts to show proportions (shown to have a slight advantage over bar charts for more complex data [360]). Readers may change the visualization using buttons above the Charagraph (Figure 3.6). To help readers track the data and understand the new visualization, the change is animated. The current representation “morphs” into the new one, following animated transition recommendations [155]. For example, each bar in a bar chart progressively morphs into each marker counterpart in a line chart (transition Figure 3.6b-c).

### 3.4.2.2 Rename Elements

By default, the axis ticks and legends are labelled using a whole number index based on their order of appearance in the text. The index numbers and any other text element in the Charagraph can be renamed by direct manipulation. Double clicking any text element enables readers to change the text associated with that element such as tick labels or legends.

### 3.4.3 Combine Information

Readers can combine data from different parts of a document by merging Charagraphs through a drag-and-drop interaction (Figure 3.7). Those compound Charagraphs support multiple series to compare different results and identify trends, and using other series as error bars to visualize the variability and uncertainty of the data.

#### 3.4.3.1 Add Series

Readers can create a Charagraph with two or more series by creating a Charagraph for each series separately and then merging them through a drag and drop interaction. To reveal this feature to users, drop zones overlay all other Charagraphs whenever a Charagraph is dragged (Figure 3.7a). If the Charagraph is dropped over the “Add Series” zone of another Charagraph, it is modified to include the new series and the dragged Charagraph is removed. If the Charagraph is a bar chart, it becomes a grouped bar chart (Figure 3.7b).

#### 3.4.3.2 Add Error Bars

Adding error bars follows the same interaction as adding series, except the Charagraph with the error bar values needs to be dropped in the zone called “Add Error Bars” (Figure 3.7a). The dragged Charagraph is turned into error bars that overlay the Charagraph (Figure 3.7c).

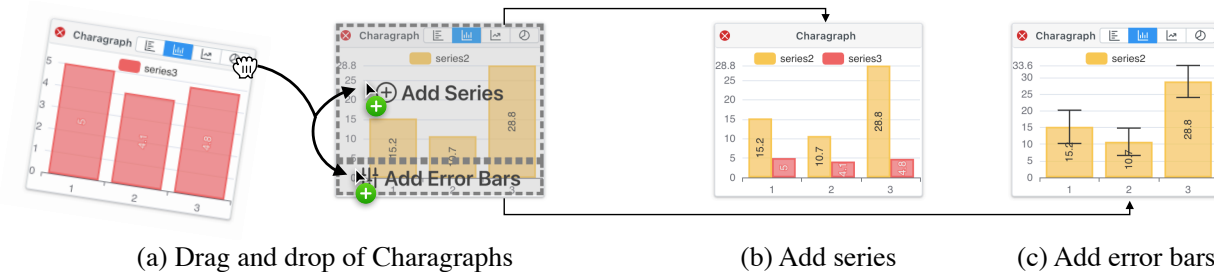


Figure 3.7: Combining information in two Charagraphs: (a) dragging a Charagraph onto a drop zone in another Charagraph combines data in different ways; (b) adding a new series; or (c) adding error bars.

## CHARAGRAPH

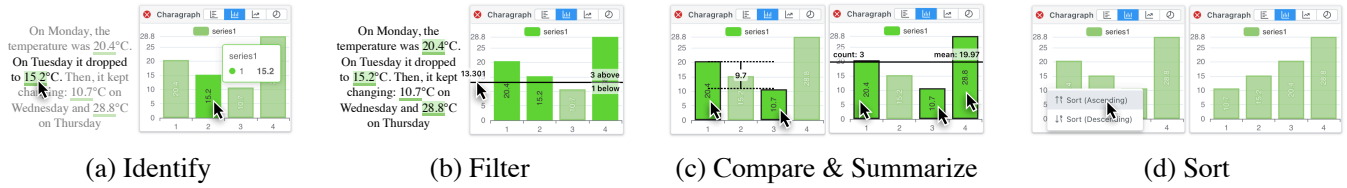


Figure 3.8: Text and chart interactions: (a) hovering above numbers in text or chart elements identifies connections; (b) moving the cursor on the Y axis controls a threshold to filter numbers; (c) clicking two values compare them and clicking more than 2 values calculates the mean; (d) right-clicking opens a contextual menu to sort.

### 3.4.4 Interact with Charagraphs

*We use interactivity to make the representation “richer” by allowing the extraction of more information and with immediate feedback.*

Charagraphs support interactivity to perform data exploration tasks [417], facilitate the connection between text and chart [209, 300, 420], and boost engagement [163]. The goal is to enable the most common visual data exploration tasks where simple visual inspection is not enough. Following Amar et al.’s taxonomy of analytic tasks, this includes filtering to find data cases matching certain conditions, computing derived values such as means or counts, and sorting to rank data cases [10]. Annotations in the text are synchronized with the corresponding chart and both text and chart representations are interactive, following recommendations from our characterization (Section 3.3).

#### 3.4.4.1 Identify

Readers can hover over a value in the text or a data point in the chart to see the value highlighted in the text and the chart. The entire document is dimmed except for the sentence containing the value to highlight the context in which the value was reported (Figure 3.8a). This highlighting interaction helps connect the values across the text and chart [216]. For example, even with unlabelled charts, the reader can determine the highest value by locating the highest bar in the chart, then pointing at it to locate it in text and find the corresponding label (Figure 3.8a).

#### 3.4.4.2 Filter

Finding values above or below a threshold is done in the chart by moving the cursor over the Y-axis (X-axis for horizontal bars). A horizontal line is shown at the location of the cursor indicating how many values are above or below. Values above the threshold are highlighted in both the text and the chart (Figure 3.8b).

#### 3.4.4.3 Compare and Summarize

Values in the text and in the chart can be clicked to remain selected. Derived values are calculated from the selection: if two values are selected, their difference is shown in the chart (Figure 3.8c left); if three or more values are

selected, the mean is calculated with a bar displayed at the corresponding position (Figure 3.8c right).

#### 3.4.4.4 *Sort*

Values follow the order of appearance in the text by default, but can be sorted by ascending and descending order by right-clicking the chart and selecting the corresponding option in a context menu (Figure 3.8d). The sorting operation is animated to help readers understand and locate elements [155].

#### 3.4.5 *Implementation*

We implemented the system using TypeScript and React [317] for the interface, PrimeReact [309] for the library of graphical components, and Apache ECharts [11, 226] for the visualizations. Everything is implemented inside the PDF.js [288] PDF viewer, the default PDF reader used by Mozilla Firefox. Our system works with all PDF documents and can be used as a general PDF viewer. The full source code is hosted online: <http://ns.inria.fr/loki/charagraph>.

In this section, we provide details for the two main components required to create a Charagraph: the selection of text and the generation and extraction of data from text.

##### 3.4.5.1 *Structural Text Selection with PDFs*

Because the PDF format is focused on visual display and small file size, the format stores text at a character-level layout with no formal representation of sentences and paragraphs<sup>1</sup>. PDF viewers traditionally reconstruct sentences based on the order of characters and their position on the page: two characters may be merged based on arbitrary distance thresholds. In PDF.js, the reconstructed sentences are invisibly overlaid on the PDF using a collection of `<span>` tags. The web browser then handles text selection using the invisible text layer. This approach has two main drawbacks: (1) Sentence reconstruction is imperfect and words may be improperly merged while others may be merged when they should not; and (2) the web browser maps user input to a list of `span` tags with no structural knowledge of the PDF. This causes unexpected and spurious selections.

Text selection was re-implemented inside the system to be consistent and support marquee selection. The algorithm relies on a geometric and semantic representation of the document: The geometric representation is obtained by modifying PDF.js to return the precise location and size of the text on the page. The semantic representation is generated from the text reconstructed by PDF.js. The representations are aligned so that a shape maps to a precise index in the text, and vice versa. This allows the marquee selection to function because individual characters are precisely located and selected only if included

<sup>1</sup> PDF can technically be more structured using tags, often for accessibility purposes (see ISO 32000-1:2008§14.8). However, very few PDFs contain those tags in practise.

*Feel free to try the tool on this thesis document. All results sections can be turned into Charagraphs.*



within a rectangle. The traditional text selection is based on the semantic representation and reproduces typical behaviour.

---

**Algorithm 1:** Generation of data-group suggestions based on prefixes

---

**Input:** A text portion  $t$  and the maximum length  $m$  of the prefixes

**Output:** A list of data-group suggestions

$numbers \leftarrow$  substrings within  $t$  matching

$/[-+.]?[0-9]+[.]?[0-9]*([eE][-+]?[0-9]+)?/$

$groups \leftarrow$  [group formed from  $numbers$ ]

**foreach**  $n \in numbers$  **do**

**for**  $l \leftarrow 0$  **to**  $m$  **do**

$pg \leftarrow \{ \}$

$p \leftarrow$  substring of length  $l$  preceding  $d$

**if**  $p$  does not match  $/\s*[\.,\.\(:\s)\s*/$  **then**

**foreach**  $c \in numbers$  **do**

**if**  $c$  is prefixed by  $p$  **then**  $pg.add(c)$

**if**  $pg \notin groups$  **then**  $groups.add(pg)$

**return**  $groups$  sorted by size

---

#### 3.4.5.2 Data-Group Suggestion

The suggestion algorithm works by identifying numbers in text with a regular expression and then grouping these values based on their suffix and prefix. Because the algorithm forms all possible groups, the list is pruned. Groups that are identical in the values that they select are merged. And groups with prefix and suffix that are empty spaces or punctuation marks are excluded. The resulting list is sorted from largest to smallest group and used as suggestions. The pseudocode of the algorithm to form groups based on prefixes is provided in Algorithm 1 (the algorithm for suffixes is identical except it matches the text following the numbers). In the system, we set the maximum length of prefixes and suffixes to 20.

### 3.5 USER STUDY

It is generally accepted that charts are better at communicating trends [259], proportions [360], uncertainty, and effect sizes [88, 104, 142] all while making it easier to interpolate [47], predict [259], and perform general interpretive operations such as locating, deriving, or comparing values [69, 101]. In contrast, text is preferred for compact representations and to extract exact numerical values.

However, these findings were obtained with charts carefully designed. It is unclear whether these findings apply to interactive charts generated from and read alongside data-rich texts in the wild. And it is unclear whether the

benefits of showing the same information in two different forms outweigh the cognitive costs.

We conducted a two-part user study: part one evaluates the *usability* of the system and its interaction model (*Can people use our system to create Charagraphs?*); part two evaluates the *utility* of Charagraphs (*Do people benefit from having Charagraphs compared to only text?*). Both parts isolate the aspect being evaluated. For the sake of clarity, the two parts are presented in different sections, although participants did both in the same session.

### 3.5.1 Participants

We recruited 12 participants (23 to 36 age range, mean = 28.9, 7 identified as female, 5 as male) from a local university. All participants were graduate students or researchers from the fields of Computer Science or Medical Science. On a 5-point scale, they all reported frequently reading documents that include in-text numbers such as scientific or news articles (Mdn=5, SD=0.8) and that they were familiar with common descriptive statistics such as standard deviations and p-values (Mdn=4, SD=0.6). In appreciation for their time, participants received a compensation of \$15 CAD.

### 3.5.2 Apparatus

Participants took part in the study remotely from their personal computer. The experiment was hosted online and participants shared their screen with the experimenter. The interaction of the participants with the website (e.g., clicks, movements, time, answers to questions), a screen capture, and the microphone were recorded.

## 3.6 STUDY PART 1: USABILITY

Participants were asked to create Charagraphs to help answer four questions about data presented in a text excerpt. Since we are examining usability of the Charagraph system, the questions serve as goals to motivate the use Charagraphs. The four questions represented two common tasks (two questions per task) based on taxonomies of tasks related to information visualizations [10, 35]. These questions can be solved easily and quickly, but require an understanding of the visualization. Each question is multiple-choice with up to eight choices. For example, a ‘Filter’ task was expressed in a question like “*How many are above/below [number]?*” and a ‘Find Extremum’ task expressed like “*What is the maximum/minimum*” or “*What is the [number]th largest/smallest?*”.

### 3.6.1 Procedure

*Introduction and Tutorial (10 mins)* – After completing a demographic questionnaire, participants were informed that they would “*use a system to visualize*

*the data presented in-text*". Participants went through a fragmented tutorial in which they reviewed a video-tutorial and then immediately practised what they just saw, but on a different text. There were 4 steps to the tutorial: 1) selecting and extracting numbers from the text to create a Charagraph; 2) adding error bars; 3) adding series; and 4) interacting with the Charagraph. The text used in the video was created for the study and presented the life expectancy for different continents in 2000 (in a paragraph) and in 2015 (in a table). Participants practised on a similar fictitious paragraph and table presenting the evolution of the temperature at two different periods. The experimenter could intervene if needed.

*Usability Task (15 mins)* – Participants were presented a text excerpt and asked to "Create a visualization to help you answer questions about [some data presented in the text]". The experimenter did not intervene and participants were free to create any Charagraph. The questions were presented after the participants created at least one Charagraph and pressed "Next". Both the paragraph and the created Charagraphs remained visible during question-answering. This phase was repeated for the three different text excerpts, always in the same order. At the end, participants completed a System Usability Scale (SUS) [38] and rated statements on a 5-point scale.

### 3.6.2 Text Excerpts

We extracted three text excerpts from different sources. All excerpts did not have an associated visualization in their original document and are therefore concrete use cases for Charagraphs. The excerpts were given to participants unchanged and also included other irrelevant data to the task. It is the participants' responsibility to extract only the information of interest to create a Charagraph.

- The SURVEY excerpt is a paragraph extracted from a report published by the United Nations Educational, Scientific and Cultural Organization (UNESCO) about the most dominant languages in films produced in 2009 [2]. It is comprised of 97 words and reports 13 different numbers, of which only 5 (the number of films for different languages) are of interest to complete the task.
- The SCIENTIFIC excerpt is a paragraph extracted from a paper published at the ACM CHI 2022 conference [220] reporting the response to a Likert questionnaire. Both means and standard deviations are reported in APA style. The paragraph is 74 words long and contains 12 numbers, of which 10 are of interest (5 means and their corresponding standard deviations).
- The WEEKLY REPORTS excerpt was formed by merging two consecutive weekly reports about the number of cases of COVID published by the World Health Organization [284, 285]. The excerpt consists of 218 words and 31 numbers. Only six numbers are relevant to the task (the number of cases for three countries in the first week, and their corresponding number of cases in the following week).

### 3.6 STUDY PART 1: USABILITY

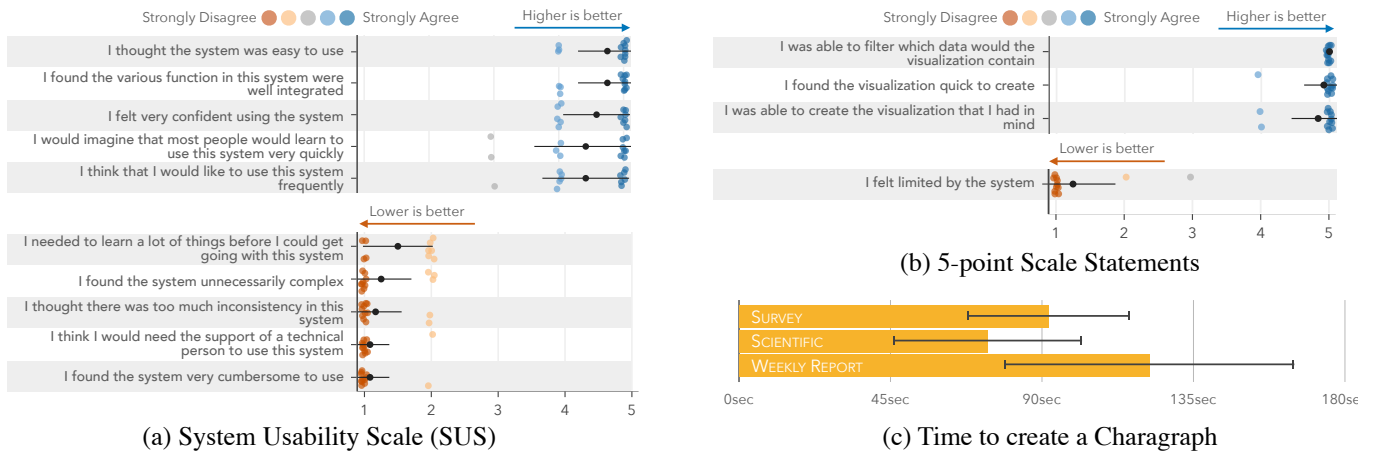


Figure 3.9: Result summary of the usability part of the study. Mean and standard deviation of participants' answers to the SUS and the statements on a 5-point scale. Average time to create a Charagraph.

The three excerpts were chosen to cover potential use cases for Charagraphs: show a distribution and compare different values (SURVEY), visualize means and uncertainty or variability (SCIENTIFIC), and visualize the evolution or trend (WEEKLY REPORTS).

#### 3.6.3 Results

All participants were able to use our system to create Charagraphs. Additionally, they correctly answered all the stimuli questions about the text excerpts. Figure 3.9 summarizes the results.

##### 3.6.3.1 Time

The time it took participants to read the text and create Charagraphs ranged from 36s to 3min 13s (M=1min 36s, SD=38s). Participants were the fastest with SCIENTIFIC (M=1min 14s, SD=27s), followed by SURVEY (M=1min 32s, SD=26s) and WEEKLY REPORTS (M=2min 2s, SD=42s).

##### 3.6.3.2 Questionnaire

The average SUS usability score was  $81^2$  (Mdn=81.2, SD=5.2) and the participants felt that they could use the system quickly and successfully: on a 5-point scale (1-strongly disagree, 5-strongly agree), they rated that they were able to create the visualization that they had in mind (Mdn=5, SD=0.4, ████), they did not feel limited by the system (Mdn=1, SD=0.6, ███), they were able to filter which data to extract and visualize (Mdn=5, SD=0, ████), and they found the visualization quick to create (Mdn=5, SD=0.3, ████).

<sup>2</sup> A SUS score above 80 is considered 'Good' [18]

### 3.6.3.3 *How Charagraphs Were Created*

Participants took 55s on average to read the text before initiating the creation of their first Charagraph. Regular text selection was used 72% of the time, and all participants relied on the automatic suggestions rather than selecting the values manually. Participants were also mostly right the first time: they deleted only 9% of the Charagraphs they created. In terms of type of visualization selected, most participants stuck to the default bar chart, adding error bars for SURVEY (11/12) and another series for WEEKLY REPORTS (11/12). Only one participant used a line chart for WEEKLY REPORTS, and another participant used a pie chart for SURVEY although they switched to a bar chart after seeing the questions.

## 3.7 STUDY PART 2: UTILITY

In this second part of the study, we focused on the utility of Charagraph. Because, we are interested by the quality of the answers given by participants, we use a broader range of tasks and questions that are more challenging to answer than in the usability task. We eliminate the usability component to keep the study focused on utility. All results are compared against a baseline text-only condition. Note that this part immediately followed the usability part.

### 3.7.1 *Text Excerpts*

We extracted three types of excerpts covering typical use cases for Charagraphs. For each type, we extracted two excerpts from different sources resulting in six excerpts in total.

- PROPORTION excerpts were obtained from two sources: 1) a news website reporting on the proportion of White Americans eligible voters per state during the 2020 presidential election [136]; and 2) the U.S. Bureau of Labor Statistics reporting on the unemployment rate among the major worker groups [208]. Both excerpts report seven different percentages.
- UNCERTAIN paragraphs were obtained from two papers published at CHI [240, 375]. They report the results for 6 and 7 items to a questionnaire using strict APA and APA-like styles resulting in 12 and 14 numbers for both mean and standard deviation.
- EVOLUTION paragraphs were obtained from two articles published by the UNESCO. One reported the evolution of the adult literacy rate between 1990 and 2016 for 5 regions, resulting in 10 numbers [364]. The other reported on the evolution of the production of films between 2005 and 2009 for five countries, resulting in 10 numbers [2].

The excerpts were given to participants unaltered (retaining style, wording, and spacing).

*Charagraph is the only project to have been evaluated on news articles and reports published by specialized agencies. More generally, the focus of this thesis is scientific documents.*

### 3.7.2 Conditions

This part of the study had two conditions:

- **BASELINE:** the text excerpts are loaded in a traditional PDF reader with typical features such as text selection, zooming, and keyword search (CTRL-F).
- **CHARAGRAPH:** the condition is identical to **BASELINE** except the PDF reader allows the creation of Charagraphs (using the system described in Section 3.4). Because this part of the study does not evaluate the usability of our system, and in order to guarantee that the Charagraphs would be useful to answer questions, the Charagraphs were designed prior to the study and participants were instructed to replicate them. We still chose to make participants create the Charagraph instead of just automatically generating it; we argue that the act of *creating* the Charagraph is part of the understanding. Additionally, we make participants use Charagraphs without labels. Our motivation is twofold: first, we believe that labels are optional because the information can be obtained through the interactivity and participants explicitly selected the data to plot, thus they already know what it represents. Second, with labels, most of the questions can be answered solely using the chart. While this is one of the motivations for Charagraphs, several studies have already shown the superiority of graphical representations over text [7, 63, 122, 191, 402]. By removing labels, we ensure that participants will go back-and-forth between graphical and textual representation, which is a more challenging and realistic scenario.

### 3.7.3 Test Questions

For each text excerpt, we designed six questions that could be answered only by reading the text. The questions were chosen to represent the most common data analysis activities [10, 35, 313]:

- **RETRIEVE:** “What is the value for [item]?”
- **FILTER:** “How many [items] are above/below [threshold]?”
- **DERIVE:** “What is the difference between [item] and [item]?”
- **EXTREMUM:** “What is the [i]th largest/smallest?”
- Specific questions depending on the type of excerpts.
  - **SIMILARITY** questions for **PROPORTION** excerpts: “What are the two most similar/dissimilar [items]?”
  - **VARIABILITY** questions for **UNCERTAIN** excerpts: “What is the [item] with the largest/smallest variability?”.
  - **TREND** questions for **EVOLUTION** excerpts: “What [item] had the largest/smallest increase in [value] between [year] and [year]?”.

All questions were pilot-tested before running the experiment to make sure they were clear and understandable. For each question, the participants were shown all possible answers or a list of eight possible answers (for DERIVE and FILTER) in a random order.

#### 3.7.4 Procedure

*Utility Task (25 mins)* – Participants were presented with a text excerpt and, after pressing “Next”, had to answer six questions about some of the data presented in-text. The questions were shown below the PDF reader and participants were given no time limit to read and answer the questions. Participants went through both conditions and all six excerpts, although in different orders. After each condition, participants completed a raw NASA-TLX and rated statements on a 5-point scale (1-strongly disagree, 5-strongly agree).

- In the BASELINE condition, when the text was first presented (and before answering questions) participants were prompted to “*Read the paragraph and press ‘Next’*”. The text remained visible during the question-answering.
- In the CHARAGRAPH condition, participants had to follow instructions to create a specific Charagraph: The instructions were only shown when the text was first presented (before answering questions) and the experimenter made sure that the participant created the expected Charagraph. To help participants understand what the visualization represented, the instructions were preceded by a brief explanation such as “*Visualize the number of films produced in 2009*”. The Charagraph, along with the text, remained visible during the question-answering and participants were free to interact with the Charagraph or not.

*Semi-Structured Interview (10 mins)* – At the end of the session, the experimenter conducted a semi-structured interview. Participants were given the opportunity to comment on any aspect of the study or the system. The experimenter also initiated discussions about the participants’ preferred condition, the aspects of the system they found most useful, their strategies, if they would use such a system, in what context, and on what kinds of documents.

#### 3.7.5 Ordering

Text excerpts and their associated questions were presented with an order following a balanced Latin square. We ensured that 1) each participant saw each text excerpt exactly once; 2) each condition had exactly one text excerpt of each type; and 3) a text excerpt appeared exactly once at every possible position every six participants. As such, half of the participants saw the same text excerpt in the BASELINE condition, while the other half saw it in the CHARAGRAPH condition. Additionally, we balanced the order of the conditions: half of the participants started with the BASELINE condition while the other half started with the CHARAGRAPH condition.

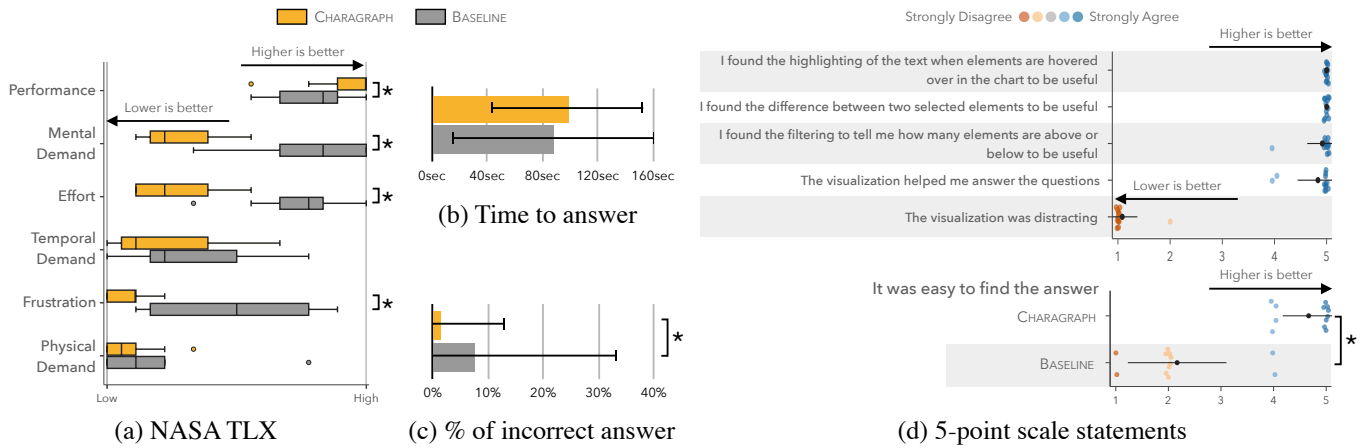


Figure 3.10: Results to (a) the raw NASA-TLX; (b) the average time to answer; (c) the average percent of incorrect answers; and (d) the 5-point scale statements. Error bars represented standard deviation. Statistical significance represented by a ‘\*’

### 3.7.6 Results

We calculate p-values using Student’s t-test when the values are normally distributed (time) and a Wilcoxon Signed-Rank test otherwise (correctness, workload, and 5-point scales) using the stats module from scipy [396].

#### 3.7.6.1 Correctness

Wilcoxon Signed-Rank found a significant effect of the condition on correctness ( $p=.003$ ,  $z=-2.98$ ). Participants were more correct at answering questions with CHARAGRAPH ( $M=98.61$ ,  $SD=11.7$ ) than BASELINE ( $M=92.59$ ,  $SD=26.19$ ). In fact, aside from three participants who correctly answered all questions regardless of the condition, the other nine participants had consistently higher accuracy with CHARAGRAPH than BASELINE.

#### 3.7.6.2 Time





The t-test did not find a significant effect of the condition on time ( $p=.15$ ,  $t=1.46$ ). Participants spent a similar amount of time to answer questions with CHARAGRAPH ( $M=1\text{min } 38\text{s}$ ,  $SD=55\text{s}$ ) than BASELINE ( $M=1\text{min } 28\text{s}$ ,  $SD=1\text{min } 13\text{s}$ ).

#### 3.7.6.3 Workload (NASA-TLX)

Wilcoxon Signed-Rank found a significant effect of condition on mental demand ( $p<.003$ ,  $z=-2.95$ ), frustration ( $p=.007$ ,  $z=-2.67$ ), effort ( $p=.002$ ,  $z=-3.09$ ) and performance ( $p=.03$ ,  $z=-2.14$ ). No significant effect for temporal ( $p=.17$ ,  $z=-1.38$ ) and physical demand ( $p=.07$ ,  $z=-1.84$ ). Compared to BASELINE, CHARAGRAPH had lower mental demand ( $M=7$  vs.  $M=16.2$ ), frustration ( $M=3.7$  vs.  $M=10.8$ ), effort ( $M=6.7$  vs.  $M=15.5$ ), and higher performance ( $M=18.5$  vs.  $M=16.5$ ).



#### 3.7.6.4 Questionnaire

All 12 participants preferred CHARAGRAPH over BASELINE. They found the answers easier to find with CHARAGRAPH (Mdn=5, SD=0.5, ) compared to BASELINE (Mdn=2, SD=0.9, ) and the difference was significant ( $p=.003$ ,  $z=-2.97$ ). Importantly, participants did not find Charagraphs to be distracting (Mdn=1, SD=0.3, ) and instead all agreed that it helped them answer the questions (Mdn=5, SD=0.4, ).

#### 3.7.6.5 Interactions Used

On average, participants hovered over the chart while answering 83% of the questions, and over the text in 44% of the questions. Counting the number of interactions in each context, clicking a value in the chart was the top interaction when answering EXTREMUM (78%), SIMILARITY (59%), VARIABILITY (65%), and TREND (75%) questions. In contrast, clicking values in-text was preferred for DERIVE questions (64%). The filter interaction was the top interaction only for FILTER questions (100%). RETRIEVE questions were most often answered without interacting with the text (3%) nor chart (11%).

### 3.8 DISCUSSION

We discuss the results by first answering our two research questions and then open the discussion to other themes that stem from observations and participants' comments.

*People can use our system to create Charagraphs.* All participants were able to create Charagraphs and use their main features to answer questions. Participants even customized the Charagraphs to their preferences: some participants kept visualizations separate because they disliked error bars, changed their style, and renamed ticks and legends.

*People benefit from having Charagraphs compared to only text.* Participants answered the questions more correctly with Charagraphs (98.61%) compared to text only (92.59%) and used and found useful all the main features of Charagraphs. All of them preferred Charagraphs and would like to use the system again. This was also reflected in the questionnaires. Participants had lower mental demand, effort, and frustration with Charagraph than when using text alone. Although harder to measure, participants' reactions suggest that Charagraphs benefit other areas. For example, despite having already read the text, P6 commented "Wow! It's them that produce the most [films]? It's mind-blowing" right after creating a Charagraph. This suggests the potential of Charagraphs to discover facts that would have been missed if only the text had been read.

### 3.8.1 Observations and Participants' Comments

*Charagraphs might change reading behaviours.* Two participants mentioned that knowing that they could create Charagraphs changed how they read the text: P9 “When I know I can create a visualization [...] it’s kind of like I already have an imagined figure in my mind when I read a paragraph” and P4 “It was me thinking about oh, this information would be useful for my graph [...] so I did this kind of filtering while I was reading, so I was thinking a lot more about what I was reading”. Researchers have suggested that the benefit of [active reading](#) might not be in the marks left on paper but on the process of creating those marks [3, 192, 325, 370, 413]. These comments suggest a similar effect with Charagraphs.

*Labels are not necessary but would have made answering questions easier and faster.* When designing the utility part of the study, we decided against having participants add labels to Charagraphs. The idea was to simulate a perhaps more realistic scenario in which a user would not have to bother adding labels because they are not absolutely necessary. We also thought that the alternative (with labels) was too powerful considering that the questions become answerable just from the Charagraph, without looking at the text. Participants commented that not having labels was not an issue because of the interactive highlight linking text and chart. P3 “The [missing] labels... I don’t think it bothered me. Because the highlighting will just link the data with the text. Because the data is very close to the text”. However, some participants also commented that they would have preferred having labels: P6 “If I had labelled the legends and ticks, I’d have been even faster”; P8 “[labels] would have been easier than referring to the text”; P9 “if I can directly find the answer in the visualization that would be better”.

*Charagraphs might help even when data is already visualized.* Readers may have preferences that might not be met by canonical visualizations [219, 267]. Even in our controlled scenarios during the usability task, participants expressed different preferences: 2 participants preferred having two separate visualizations for means and standard deviation instead of a combined representation with error bars. P9 explained “for me the bar chart is easier to read than error bars [because error bars] do not start at 0”. If not preferences, a participant also mentioned that existing visualizations might be overblown, P12 “Even if they [documents] do have graphs, they don’t have the specific information that you are looking for... so if you want, like, focus on a specific thing like the mean for example”. Similarly, an existing static visualization might be deceptive because it has truncated axes, or because the aspect ratio exaggerates a trend [254, 292]. Or the visualization has accessibility issues because of its colour scheme and style [194, 412]. In these cases, as long as the data is presented in text, a Charagraph can recreate the static visualization, but with the possibility of tweaking its scale and style. The resulting “reader-created” visualizations can be used to compare or replace the static canonical visualizations already present in the document.

### 3.8.2 Limitations

*This limitation has been partially solved by using ChatGPT like in Statslator, presented in the next chapter.*

*Some in-text data might be difficult or impossible to extract.* The suggestion algorithm forms groups based on the text before and after numbers. This requires numbers to share prefixes (e.g., M=, SD=) or suffixes (e.g., %, ms, cm) or sentences to follow similar structures with identical prepositions before numbers. When this assumption fails, users have to resort to adding missing values manually. Additionally, data other than numbers and certain sentence constructions are currently unsupported. For example, a sentence might refer to a quantity mentioned previously, such as “half of the participants”. Lastly, combining two Charagraphs or adding error bars relies on the order of the data points. As such, this requires the data to be presented in text in the same order or to manually re-organize the data in the Charagraphs so that they match.

*The instructions in the utility study might have made participants less engaged with the text.* In the utility part of the study, we gave participants instructions to follow to create a Charagraph. This was done to remove the usability component (which had already been tested in the previous part). Several participants mentioned that this made them less engaged with the text: P4 “I definitely noticed, with the instruction I was not reading the text [...] I think I actually preferred the sort of exploratory mode [usability part] where I could just make the graphs I wanted without the instructions” and P5 “I didn’t like having the instructions [...] I think, in the last condition [baseline with only text] and in the condition without instructions [usability part] I read the text more”.

*Participants were well-versed in data literacy.* All the participants in our study were graduate students and researchers trained in reading scientific documents and interpreting numeric values. Thus, participants could understand the content with text only. Given findings that charts help casual readers understand numeric data [89, 306], it is plausible that people who are less versed in reading scientific documents might benefit from Charagraphs as long as they are familiar with data visualizations.

### 3.8.3 Future Work

*Exploring the design space of Charagraphs.* In our characterization (Section 3.3), we mentioned two other level of integration that are currently not supported by our system: chart in text using word-scale visualizations [126, 385] and text in chart using annotated charts [318]. Our study focused on separate representations that let readers choose the one that is most appropriate. It would be interesting to observe the impact of the two other modes on reader’s behaviours. Similarly, we believe there is an opportunity to extract the information contained within the text using NLP to guide the generation of Charagraphs and give better defaults. From our characterization (Section 3.3), statistical-terms and units could be extracted to design embellished charts that are more memorable [20, 32]; labels could be recovered to annotate charts

and add names to legends, ticks and axes; and the semantic could decide the visualization.

*Support for advanced Natural Language Processing (NLP).* Adding NLP has the potential of making the creation of Charagraphs easier and quicker. First, the suggestion algorithm could rely on named entity recognition (NER) to filter numbers that are not data points such as references to sections, dates and figures. Semantically related groups of numbers could be formed by analyzing the sentence structure and content which might result in better suggestions. Second, the extraction of data points could be made more robust by understanding common natural language structures. For example, to understand “half of the participants”, a system would need to keep track of the mentioned values (here, the number of participants), understand the described arithmetic (half of), and calculate the final number. Third, NLP could extract meta-information about the data, such as p-values to automatically decorate the visualization with the paired significance level. Last, NLP could automatize the extraction of labels by parsing the relation between the number and the label. Such a system could draw from approaches to parse textual analogies [100, 212]. However, these remain challenging even with modern techniques (see the difficulties faced by approaches to automatically connect text and chart [209, 300]).

*Support for other visualizations and tasks.* In addition to the common bar, line, and pie charts already supported, the system could be extended to include scatter plots, choropleth maps, box plots, and radar charts, which are relatively common [21]. The challenge is whether required data can be mined or calculated from the text. For example, a scatter plot is designed to show raw data points, but text typically reports on aggregated values representing raw data (e.g., means and standard deviations). A scatter plot also requires data in two-dimensions (e.g. x and y), which may not be reported in-text. Regarding other tasks, a Charagraph often represents a subset of data, or a combination of some data of interest. This suggests other tasks for a reader, such as to characterize the distribution of the selected data, find correlations, clusters, or anomalies. These tasks could be made easier by automatically identifying data facts and remarkable features [377] of the generated Charagraphs, and offer suggestions or annotations [209, 362].

*Evaluating the impact of Charagraphs on information retention and engagement.* Other benefits of graphical representations over text might transfer to Charagraphs. For example, previous work has shown better recall of information [33, 161] and greater reader engagement [253] when using data visualization compared to text alone. A user study that specifically evaluates these aspects could further inform us on the benefits of Charagraphs.

*Charagraphs with printed text and physical documents.* The system currently works with digital documents but could be extended to the physical world. Using a smartphone, readers could take a photo of a document containing number in the text. The system would process this photo using Optical Character Recognition (OCR), offer similar selection features to extract the values,

and generate a Charagraph on the smartphone. Future work could investigate the design and implementation of such a system or a more advanced version that would overlay Charagraphs on the physical document, similar to DuallyNoted that uses augmented reality on a smartphone to annotate paper documents [311].

### 3.9 CONCLUSION

Documents seldom include visualizations despite the advantages of graphics compared to textual representations. To help readers when visualizations are missing or poorly designed, we introduced Charagraphs, dynamically generated interactive charts and annotations for in-situ visualization, comparison, and manipulation of data included within text. We also presented a document viewer to create Charagraphs in existing PDF documents. In a user study, participants could quickly create their own imagined visualizations from data in text and were more correct when answering questions using Charagraphs compared to only reading text. Charagraphs provide an immediate solution for readers wanting to visualize and manipulate numeric data in existing documents, and explore the concept of reader-generated visualizations.

# 4

## INTERACTIVE TRANSLATION OF NHST AND ESTIMATION STATISTICS REPORTING STYLES IN SCIENTIFIC DOCUMENTS

---

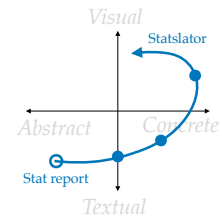
In the previous chapter, Charagraphs were found to help readers understand and manipulate the numbers in paragraphs of text by producing visualizations (benefits of varying the type of media). However, this assumed readers were familiar with the numbers in-text. In this chapter, we investigate ways to adapt the content of a document to help readers understand unfamiliar numbers and statistics (benefits of adaptation to reduce [extraneous load](#)).

### 4.1 MOTIVATION

When studying a population, a common approach is to collect data from a random sample of that population (e.g., participants) and use inferential statistics to generalize the findings. A detailed report of this analysis in scientific documents enables readers to evaluate the strength of the findings. However, the presentation of these reports as imposed by authors might be hard to understand [71, 205], incomplete [373], even misleading [25, 132, 218].

Within the HCI community and other empirical fields, statistical reports commonly include *p-values* obtained from *null hypothesis significance testing* (NHST). However, *p-values* and their associated reporting language might mislead even trained scientists [218, 255], and reporting *p-values* alone is insufficient to draw meaningful and nuanced conclusions, capture uncertainty of results, and answer quantitative questions about the effect [71, 88, 132, 211]. For these reasons, some recommend reporting additional information such as effect sizes and interval estimates [14, 373], while others advocate for avoiding *p-values* and NHST altogether [45, 70]. This debate is particularly visible in HCI where all positions cohabit. *P-values* are prevalent in publications [27, 397] and more robust NHST approaches are being developed [99, 408]. Yet, workshops and articles are also promoting estimation [45, 88, 142] and bayesian approaches [188, 299].

While much of the ongoing debate is about *what* to report, it also extends to *how* inferential statistics should be reported. For example, what symbols to use [14], what words to communicate findings [28, 88], and what numeric precision to use [30]. How authors report statistics continuously evolves as the field matures: for example, the practise of reporting *p-values* using inequalities has noticeably decreased since 2010 [27]. New reporting styles like graphical presentations of effect sizes can effectively convey the information [45, 142, 156], but such figures are seldom found in documents because they can be difficult to create [57, 187] and take additional space [7, 142]. Even graphical representations can be misunderstood, for example, error bars may depict standard deviations, standard errors, and confidence intervals [72]. At



In our design space fig. 1.1, Statslator transforms abstract statistical reports into familiar numbers, textual, visual and interactive reports.

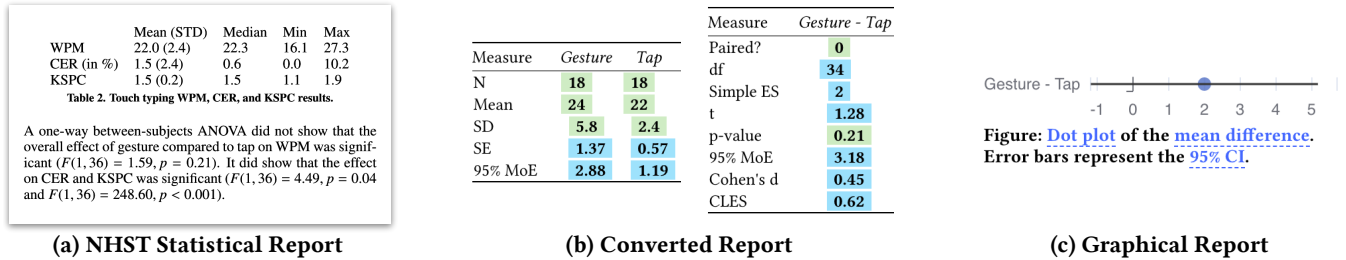


Figure 4.1: Statslator takes existing statistical reports (a) using NHST or estimation; (b) calculates all possible statistical values using accurate conversion equations; (c) shows the report using graphical and interactive figures configurable by readers.

best, the meaning of error bars is clear and a reader can recover a quantity of interest by “eyeballing it” [71]. At worst, a reader can form the wrong conclusion about the depicted results [205].

It is virtually impossible to please all readers with one style of report for inferential analysis. For this reason, some recommend including multiple reporting styles [14] which can be extended to “multiverse analysis” [145, 365] and even made interactive to give readers control over the presentation [91]. However, they are difficult to create [143, 232, 335] and often incompatible with publishing workflows that impose page limits and digital formats with little or no support for interactivity [247]. Even in the unlikely case that all authors adopt reporting styles akin to interactive multiverse analyses, it is unclear how previously published documents could be supported, especially when their raw data is unavailable.

We argue for reader-centred statistics: readers should have the final word on statistical inference presentation because what matters is correct interpretation, be it through estimation or NHST<sup>1</sup>, textual or graphical content, or static or interactive documents. Moreover, this should be possible with old and new documents without substantial efforts. Our key insight is that most scientific articles report enough information about at least one type of inferential statistical analysis which can be translated into a different reporting style. For example, a confidence interval (CI) can be calculated given a p-value and means. The difficulty lies in three aspects: (1) obtaining the statistical information—we analyzed statistical reports at CHI and propose a semi-automatic pipeline to extract common statistical reports; (2) converting the information into a target statistic—we demonstrate how to do these conversions and thoroughly evaluate their accuracy given common reporting practices; and (3) presenting the converted information back to readers in their preferred style—we design a document reader that allows readers to customize textual, graphical, and interactive statistical reports. We implement our solutions to these aspects into Statslator, a tool for readers to retro-actively encode best practises in terms of statistical reporting with no author involvement. This effectively supports documents that were already disseminated

*While the primary motivation is to tailor the statistical report to the reader (benefits of adaptation), the representations also benefit from being interactive and varying the type of media.*

<sup>1</sup> Some argue p-values cannot be properly interpreted even when their pitfalls are understood [45]. We believe readers should still decide if they wish to use p-values.

and bypasses limitations due to outdated publishing practises that discourage the use of figures because of page limits and prevent interactive documents due to reliance on the PDF format or inflexible publishing workflows.

## 4.2 BACKGROUND AND RELATED WORK

We first review the challenges associated with different ways of reporting statistics and then detail recommendations made by the community to tackle these challenges. We then detail how previous systems allowed readers to personalize documents without involving authors.

### 4.2.1 Challenges with Statistical Reporting

Prevalent practices in scientific documents can make statistical reports hard to understand.

First, *there is much confusion about the meaning of statistical values*. For example, the potential for misinterpreting p-values [71, 132]: a p-value might lead a reader to believe the result is more certain than it is (dichotomous thinking); that it conveys effect size (ambiguous use of “significant”); and that there is no difference when  $p > .05$  or that p is the probability that the null hypothesis is true (inverse probability fallacy). But misinterpretations are not limited to p-values. Any statistical report not fully understood by readers might be misinterpreted, including standard errors, confidence intervals [25], and effect sizes [196]. Thus, it is important to give readers access to statistics they are familiar with, or default to ones that are easily understood.

Second, *there are many equivalent ways to report the same statistical values, all requiring a different interpretation*. Consider two groups  $M_1 = 5, SD = 1$  and  $M_2 = 7, SD_2 = 3$ . Now consider the multitude of valid ways to convey the effect size (calculated value in parenthesis): the mean difference (2), Cohen’s d (.89), Glass’  $\delta$  (.67), the rank-biserial correlation (.41), odds ratio (5.1),  $\eta^2$  (.17), Cohen’s f (.45), and the common language effect size (.74). Worse, there is a lack of consensus on how to calculate some specific effect sizes such as Cohen’s d and all variations are found in documents, with no indication of what formula was used (like we just did) [210]. Similarly, a confidence interval can be reported at various confidence level, and for different estimates [71]. These inconsistencies defeat most of the purpose of standardized effect sizes to convey an effect size that is comparable across studies.

Third, *there are many ways to report and represent statistical values*. In textual reports, values are often introduced with letters and Greek symbols [14]. Besides being confusing, these symbols are inconsistent across documents and sometimes collide. For example,  $d$  can refer to any of the four ways of calculating Cohen’s d [210]. Similarly,  $r$  is the symbol for both the rank-biserial correlation and the Pearson correlation coefficient (which are equivalent only in specific situations). The reported values themselves can be rounded with arbitrary precision, or they can be given as inequalities, often the case for p-values [27, 30].



Similar inconsistencies are in graphical reports of statistics. There is a wide diversity of chart types, and much has been written about how charts can deceive or confuse an inattentive reader [43, 292]. Additionally, standard graphical marks may not have consistent meanings. For example, the visual style of an “error bar” is easily recognizable, but whether they represent standard error, standard deviation or a confidence interval may be misinterpreted or unknown [25, 72, 162, 205]. Even the simple bars or points could be representing separate means or mean differences [70]. With adequate detail in a caption, these kinds of issues can be mitigated and some recommend graphical reports instead of text [88, 120, 142, 156]. But, authors are reluctant to include many graphs since they can be difficult and time consuming to create and take up additional space [7, 57, 187], which is unwanted for publications that limit page size or for peer review guidelines that equate contribution size to paper length.

Statistical reports are already difficult to read and understand for readers, and the diversity and inconsistency in how statistics are reported only makes it more challenging. Our approach enables a customizable presentation of statistical results controlled by readers, with embedded context, consistent calculations, and connections across reporting styles.

#### 4.2.2 Recommendations to Report Statistics

The primary response to challenges with statistical reporting is to encourage authors to present results using a consistent, detailed, and clear style. For example, the American Psychological Association (APA) states that “*complete reporting of all tested hypotheses and estimates of appropriate effect sizes and confidence intervals are the minimum expectations for all APA journals*” [14]. To make these reports consistent, APA also recommends specific phrasings and symbols. Within the HCI community, similar recommendations have been proposed [88, 142], and tools have been created to assist authors in the process [181, 399]. But these guidelines have yet to become standard practice [123]: as of 2018, about 15% of CHI papers included confidence interval whereas 50% reported p-values [27].

In parallel, these recommendations keep evolving. For example, the APA started recommending the use of CIs in its fifth edition following a push from the community [146, 199]. Alternative presentations of statistics have been proposed and are often supported by experimental evidence. For example, figures showing effect size and confidence intervals can help readers [88, 142, 156]. Hypothetical outcome plots [172] (HOPs) rely on animations and help convey uncertainty [185]. Multiverse analysis reports [145, 365] that might be explorable [91], can highlight how fragile or strong the results are. Analogies and some more natural effect sizes such as the common language effect size are often better understood [196].

While recommendations exist, the bottleneck seems to be in their implementation, either because of slow adoption, difficulty of creation, or publishing

format limitations. And of course, even if all these issues were to be solved, the problem would remain for existing documents.

#### 4.2.3 *Personalized Reading Experience*

When authors' adoption of new guidelines is slow or unlikely to happen, a possible solution is to offer tools for readers. Several systems have been proposed that take as input a document and augment it in various ways.

For example, reported measurements such as distances might be difficult to interpret if readers cannot relate to them. Thus, systems have been proposed to automatically generate analogies and relatable explanations and visualizations of the measurements reported in a document [171, 197]. Similarly, there is a wealth of research on how to generate visualizations to accompany documents, either to give more contexts while reading by leveraging external databases of relevant information [118, 170], to generate visualization in-context for data that is already in the document but scattered in textual tables [17], or simply to give readers a way to annotate documents using charts so that they can make sense of numbers in-text [246].

These approaches have the advantage of being immediately applicable and to all documents, including those already disseminated. In this work, we adapt this approach to the context of statistical reports, and adjust the generated presentations to fit the numerous recommendations made by the community.

### 4.3 WHAT IS REPORTED AT CHI?

We analyzed the proceedings of ACM CHI conferences to better understand what inferential statistics are reported and how they are presented. The goal is to gauge the feasibility of generating different statistical representations from the data reported in text. We examine CHI papers because HCI is a multidisciplinary field and CHI is very large and diverse. Statistical practises likely vary among CHI authors depending on their field, background, and exposure to inferential statistics.

Our analysis focuses on text, not values included within figures or tables, and our approach does not consider complex sentence structures. Therefore, our results should be viewed as lower bounds rather than absolute proportions.

#### 4.3.1 *Corpus of Papers and Analysis*

We collected 9,611 PDFs from 1982 to 2022<sup>2</sup>. For each paper, text was extracted using a Python port<sup>3</sup> of MuPDF<sup>4</sup> and a set of case-insensitive regular expressions as identified statistical reports:

<sup>2</sup> Note CHI was not held in 1984

<sup>3</sup> <https://github.com/pymupdf/PyMuPDF>

<sup>4</sup> <https://mupdf.com/>

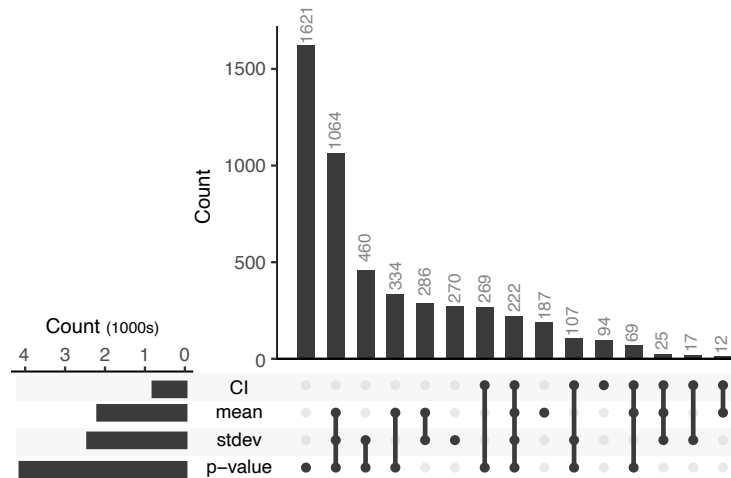


Figure 4.2: Upset plot showing the number of papers that report different sets of CIs, means, standard deviations, and p-values. Lines connecting dots across rows indicate what values are in the intersecting set.

- One way to identify reported values is to look for numbers preceded by the relationship symbol '=', '<', '>', '≤', '≥', '<<', or '>>'. When found, the number and preceding word or symbol were extracted.
- Another way to identify values reported in text without a relationship symbol is to look for names of common descriptive and inferential statistics such as 'mean', 'median', 'standard deviation' and variations such as 'M', 'Mdn', and 'μ'. We only extracted those followed by the verb 'be' in any form such as 'is', 'was', and 'were', and then followed by a number.
- To count the mention of confidence intervals, we looked for the terms '%CI', '% CI', and 'confidence intervals'.
- To detect statistical tests, we checked for the name of a specific test among a list of 15 popular ones, including 't-test', 'ANOVA' and possible variations such as 'ANCOVA' and 'MANOVA'.

The results were harmonized by reviewing the top candidates and grouping the ones referring to identical values. For example, 'mean', 'average', 'M', and 'μ' were all grouped as 'mean'. Statistical tests known under different names were also grouped, such as 'Wilcoxon Rank Sum Test' and 'Mann-Whitney U Test'.

#### 4.3.2 Results

As a sanity check, we compared our results to Besançon and Dragicevic [27] who also used regular expressions to examine CHI proceedings although their analysis was limited to p-values and CIs from 2010 and 2018. We found 5.9% to 14.2% of CI for 2010-2018, they report "from 6% to 15%". We found 48.4% of p-values, they report "around 50%".

#### 4.3.2.1 *Reported Values*

6,266 papers (65%) mentioned at least one value. Of these papers, p-values (66%) were most common, followed by standard deviations (39%), means (35%), F-values (28%), t-scores (19%), and confidence intervals (13%). Standard errors were found in only 2% of papers. Standardized effect sizes were also seldom found, such as Pearson's correlation coefficient (11%) and Cohen's  $d$  (5%).

Figure 4.2 shows the number of times a CI, mean, standard deviation, or p-value, or combination of these values, are reported in a paper. Of particular interest are papers reporting both p-values and means; means and standard deviations; and CI and means. For most of these 2,012 papers, we will show translating to a different reporting style is possible (section 4.4). Although papers reporting only p-values (1,612) cannot be translated, standardized effect sizes can still be calculated to complete the report. For the rest, additional values will be needed. Overall, a large proportion of CHI papers have useful statistical reports, and recall that our conservative analysis likely underestimate actual occurrences and it is possible that values like means can be estimated from figures [182, 248]

#### 4.3.2.2 *Number of Decimals*

Consistent with APA recommendations [14], standard deviations, means, F-values, t-scores, and CIs were reported with a median of two decimals. The exception was for p-values, which were reported with a median of three decimals. However, 17% of these p-values were reported as inequalities with values '0.05' or '0.01'.

#### 4.3.2.3 *Statistical Tests*

4,800 papers (50%) mentioned at least one statistical test. Among these papers, t-tests were mentioned in most (61%), followed by ANOVA (49%), Wilcoxon signed-rank test (10%), Mann-Whitney U test (9%), and Chi-squared test (9%). Other tests, such as Friedman, were found in less than 5% of these papers.

### 4.4 CONVERTING STATISTICAL REPORTS

Based on the information reported in papers, we present a set of equations to perform bidirectional conversions between NHST-based reports and estimation-based reports. To our knowledge, these derivations were not or superficially covered, especially in the context of practices common in HCI studies. For example, some readers might have been taught the "conversion rule" that 95% CI is about twice as large as the standard error; the "overlap rule" that if two independent 95% CIs on the separate mean just touch,  $p$  is about 0.01, and no overlap means  $p < 0.01$ ; and the "difference rule" that if  $p > .05$  then the 95% CI on the effect size will extend slightly past 0 [71]. What is sometimes omitted, however, is how these methods were derived,

how precise conversions can be obtained, and when they do not work. For example, the overlap rule for a CI fails when the study follows a within-subject design [71], and the other two assume that the sample size is large. Similarly, a widely cited article in the medical science community described how to convert p-values into CIs [8], but it assumes more than 60 participants: if applied to smaller studies typical in HCI [42], the converted estimations would be overoptimistic leading to false conclusions.

For all conversions, we presume the size of the groups  $N_1$  and  $N_2$  and the study design are known (i.e., *within* or *between-subject*) since this information would be reported in any rigorous scientific report. By extension, the degrees of freedom  $df$  for a t-test can be calculated if not already reported. For within designs  $df = N - 1$ , otherwise  $df = N_1 + N_2 - 2$ . Unless specified otherwise, all equations are valid for all variations of t-test for both within and between subject, groups of various sizes, and with equal or unequal variances.

#### 4.4.1 Converting to Confidence Intervals

When comparing two groups, the CI of interest is the CI for the effect size, where the effect size is usually the difference between the two group means. This CI can replace a p-value as it conveys the estimate of effect size and the uncertainty around it.

The CI is calculated as  $[\Delta M - MoE, \Delta M + MoE]$  where  $\Delta M = M_2 - M_1$  is the difference of the group means, or “unstandardized” effect size, and  $MoE$  is the margin of error (corresponding to half the CI). Below, we show different ways to calculate  $MoE$  needed to obtain the CI.

##### 4.4.1.1 From means and t-score

Cumming [70] explains that the calculation of the  $MoE$  depends on the t component  $C_t = t_\alpha(df)$ , the variability component  $C_v$ , and the sample size component  $C_s$ . Adopting this terminology, the calculation of a t-score can be expressed as follows.

$$tscore = \frac{M_2 - M_1}{C_v \times C_s} \quad (4.1)$$

Note this formulation is an abstraction since the calculation of  $C_v$  and  $C_s$  depends on the choice of t-test and the study design. But by referring to these three components, our equations are compatible with all standard t-tests. As such, the equation above can be rearranged to recover  $C_v \times C_s$  given  $M_1$  and  $M_2$  and a t-score. The remaining  $C_t$  term is calculated using the t-distribution for a given degree of freedom (noted  $t(df)$ ). As a result, given a t-score  $tscore$ , the means of both groups  $M_1$  and  $M_2$ , the  $MoE$  at confidence level  $\alpha$  is calculated as follows

$$MoE_\alpha = t_\alpha(df) \times \frac{M_2 - M_1}{tscore} \quad (4.2)$$

A simplification that is often made is to use the normal distribution instead of the t-distribution; with a large sample size of at least 30, the t-distribution will approximate a normal distribution, so knowing  $df$  can be relaxed. For

example, for a large  $df$  and a 95% confidence level,  $t_{.95} \approx z_{.95} = 1.96$  which simplifies to the equation presented by Altman and Bland [8]. However, in many HCI studies, the sample size is small, frequently around 12 participants [42]. In these cases, this assumption leads to narrower “overoptimistic” CIs and should be avoided.

#### 4.4.1.2 From means and p-value

The p-value of a t-test is measured by the area under the curve of the t-distribution corresponding to the t-score. It refers to the probability of obtaining the t-score or a more extreme one when assuming the null hypothesis is true. Thus, to recover the t-score from a two-tailed p-value, we can use the inverse cumulative distribution function to recover the t-score that gives an area under the curve matching the p-value. As such, only the p-value and a degree of freedom are needed.

Once the t-score is recovered, Equation 4.2 is used to calculate the CI. One caveat of this approach is that p-values might be reported using inequalities, although this practise is declining [27]. In these cases, the CI will be larger and more likely to cause type 2 errors. Thus, it is often preferable to use the t-score when reported (section 4.4.1.1).

#### 4.4.1.3 From independent means and standard deviations

Depending on whether the study design is within or between subject, the group means (and t-test) will be dependent or independent. Given the means and standard deviations of two between subject groups, an independent t-test can be calculated to obtain a t-score. Equation 4.2 can then be used to obtain the CI. Note that the same does not apply to a dependent t-test for a within subject condition because the required standard deviation of the paired differences cannot be estimated from means and standard deviations alone. In this case, one of the solutions above should be used.

#### 4.4.1.4 From a CI at a different confidence level

Given a confidence interval at confidence level  $\alpha_0$ , we can adapt equation 4.2 to calculate the CI at  $\alpha_1$ :

$$MoE_{\alpha_1} = \frac{MoE_{\alpha_0}}{t_{\alpha_0}(df)} \times t_{\alpha_1}(df) \quad (4.3)$$

#### 4.4.1.5 From CIs on separate means

Given two CIs on means of two independent groups, the CI of the difference can be recovered by first calculating the mean and standard deviation of each group, and then using the method from section 4.4.1.3. Assuming the t or normal distribution were used, the CI is symmetrical such that its centre is

the mean of the sample. For the standard deviation  $sd$ , it can be recovered by rearranging the equation that calculates the  $MoE$  (half of the CI),

$$sd = \frac{MoE_{\alpha}}{t_{\alpha}(N-1)} \times \sqrt{N} \quad (4.4)$$

In case another method such as bootstrapping was used to calculate the CI, the recovered  $sd$  will be approximate, although it is reasonable to calculate the CI on the difference of means (section 4.5.5).

Note also that, for two dependent groups, the separate CIs are not enough and the t-score or p-value will be needed to recover the standard deviation of the paired differences.

#### 4.4.2 Converting to p-values

Given a CI on the mean difference of two groups, and the means of both groups  $M_1$  and  $M_2$ , equation 4.2 can be rearranged to recover the t-score,

$$t_{score} = \frac{t_{\alpha}(df)}{MoE_{\alpha}} \times (M_2 - M_1) \quad (4.5)$$

Then, the t-score is converted into a two-tailed p-value using the the cumulative distribution function of the t-distribution.

Note that this assumes the t distribution was used to calculate the CI. The validity of this conversion with bootstrapped CIs is evaluated in section 4.5.5.

#### 4.4.3 Converting to Standardized Effect Sizes

Whereas section 4.4.1 used the mean difference as an “unstandardized” effect size, standardized effect sizes such as Cohen’s  $d$  might be of interest to readers when comparing results that are on different scales and from different experiments [70]. Many effect sizes have been proposed, but most can be converted from one to another. Thus, below we show how to obtain Cohen’s  $d$ , and how to convert it to a different effect size. These equations can be trivially extended to the case where only CIs are available by first converting the CI to a p-value (section 4.4.2).

##### 4.4.3.1 From standard deviations

Cohen’s  $d$  is calculated by dividing the mean difference by a ‘standardizer’ which differs depending on the study design. For a between-subject study, the recommended standardizer is the pooled standard deviation [71, 210]. Given the standard deviations of two groups  $SD_1$  and  $SD_2$ ,

$$d_{\text{between}} = \frac{M_2 - M_1}{\sqrt{\frac{(N_1-1)SD_1^2 + (N_2-1)SD_2^2}{df}}} \quad (4.6)$$

For a within subject design, the standardizer is usually a pooled average of the standard deviations [71],

$$d_{\text{within}} = \frac{M_2 - M_1}{\sqrt{\frac{SD_1^2 + SD_2^2}{2}}} \quad (4.7)$$

4.4.3.2 *From a t-score (or p-value)*

Cohen's  $d$  can be obtained from just a  $t$ -score (and, by extension, a  $p$ -value). We report the equations obtained from Daniël Lakens [210]:

$$d_{\text{between}} = t_{\text{score}} \sqrt{\frac{1}{N_1} + \frac{1}{N_2}} \quad (4.8)$$

$$d_{\text{within}} = \frac{t_{\text{score}}}{\sqrt{N}} \quad (4.9)$$

4.4.3.3 *Converting between effect sizes*

The conversions between effect sizes have been well covered by previous work, especially in the context of meta-analyses. From Cohen's  $d$ , it is possible to calculate the point-biserial correlation and Hedges'  $g$  [62]; odds ratios [31]; and the "common language effect size", also referred to as the "probability of superiority" [94].

4.4.4 *Other Considerations*4.4.4.1 *Multiple comparisons and corrected p-values*

If  $p$ -values are corrected, then a converted CI will also be adjusted which may not reflect the expected confidence level. Corrected  $p$ -values often appear in reports using exploratory contrasts to account for multiple comparisons (e.g., post-hoc analysis after an ANOVA). Methods for estimation-based statistics traditionally avoid this issue by planning the analysis a priori to only focus on a few comparisons (typically no more than the degrees of freedom) [71], or to adjust CIs using different corrections [95]. When the original "uncorrected" CI is desired, it can be obtained either by not relying on the  $p$ -value (e.g., using the  $t$ -score or standard deviation), or by "unadjusting" the  $p$ -value, assuming the original correction method is known.

The goal is to recover the uncorrected  $p$ -value  $p$  given the corrected  $p$ -value  $p^*$ . For a Bonferroni correction [262],  $p = p^*/n$  where  $n$  is the number of pairwise comparisons done. For a Šidák correction [355],  $p = 1 - (1 - p^*)^{1/n}$ . For a Holm-Bonferroni correction [164],  $p = p^*/(n - i)$  where  $i$  is the position of the  $p$ -value in the list of sorted  $p$ -values of all pairwise comparisons. In some cases, the recovered uncorrected  $p$ -values might be inexact: first, a correction for multiplicity typically increases  $p$ -values and might make them exceed 1 in which case the  $p$ -value is often rounded to 1 and some precision is lost. Second, for Holm-Bonferroni, the rank cannot be recovered when the correction changed the ordering of the  $p$ -values. These issues are investigated in detail in section 4.5.4.

4.4.4.2 *t-test Variations*

The equations above apply to common variations of  $t$ -tests including those for unequal variances, and unequal sample sizes. For example, for Yuen and



Welch's t-test, the degrees of freedom will be different, but the equation to convert the p-value to CI will remain the same, given the correct  $df$  is used. Additionally, in some situations, different tests are equivalent to t-tests. For example, the result of an ANOVA on a condition with two-levels will be identical to a t-test, and the t-score is the square root of the F-score [158, 159].

#### 4.4.4.3 *Non-parametric statistics*

Tests such as Wilcoxon signed rank and Mann-Whitney are typically reported when data does not follow the assumptions of a t-test. The equations above cannot be used with these tests and it is unclear how CIs could be recovered without the underlying data to find its distribution, and without the test giving an indication of what that distribution might be.

#### 4.4.4.4 *Conversions With Incorrect Reports*

One might also wonder what the conversion would do in cases where a t-test was applied on data that clearly breaks t-test assumptions. First, it is important to recognize that a t-test might still be a reasonable choice: for example, the central limit theorem states that, in many cases, the mean of the data will be normally distributed given a large enough sample size, and thus, a t-test can be used. This might explain why t-tests are so prevalent in CHI papers (61% of papers that mentioned a statistical test). But more importantly, the equations are meant to convert the results, not to fix them. Little can be done if a statistical report uses the wrong test and obtains potentially erroneous results.

#### 4.4.4.5 *Chaining Equations*

Most equations can be rearranged to calculate the measurements they involve. For the sake of brevity, we only presented each equation once. However, we provide an open source JavaScript library with more than 50 equations and possible rearrangements<sup>5</sup>. Given a set of measurements, the library will iteratively calculate all possible values. For a given value of interest, the library can also describe the possible ways to calculate it, and what values would be needed. It can also identify possible inconsistencies when there are multiple ways of calculating a value, but they yield different results (using a relative error threshold, currently 0.1).

## 4.5 CONVERSION ACCURACY

While the equations in section 4.4 are exact, written reports often round numbers, may use small samples, and calculate values using methods that could impact the conversion accuracy. Consistent with validation approaches used in the statistics literature [99, 421], we conduct Monte-Carlo simulations of common statistical reports. Our three experiments use the conversion

<sup>5</sup> <https://ns.inria.fr/loki/statslator/>

equations above to test the accuracy of: (1) converting reports of t-tests to CI; (2) converting reports of post-hoc pairwise comparisons with corrections to CI; and (3) converting reports of confidence intervals calculated via t-distribution and bootstrap methods to p-values.

The simulations are in python using numpy [149] and pingouin [388] for statistical tests and distributions. The bootstrapped confidence intervals are calculated using arch [352]. The code is available for replication<sup>6</sup>.

#### 4.5.1 Data Generation

As is standard with statistical simulations [99, 421], the datasets used in our fictional reports are automatically generated to test a wide range of study designs. All three experiments use generated designs that are prevalent in HCI with the following shared conditions:

- **DESIGN:** Either between-subject or within-subject.
- **SIZE:** The size of each group, either 8, 12, 24, or 40, following the range of sample sizes commonly found in HCI studies [42].
- **DECIMALS:** The number of decimals used to round all values (mean, standard deviation, t-score, and bounds of CI). For p-values, the rounding is done using the number of significant digits to more closely match what would be reported in a paper. For example, with 1 decimal, a  $p=0.048$  is rounded to 0.05.

*While we use designs common in HCI, do not see this as an endorsement for using small sample sizes or a specific study design.*

Although some experiments might add conditions, they are all performed on *at least* 4 SIZES  $\times$  2 DESIGNS  $\times$  10,000 repetitions. We use 10,000 repetitions as it has been shown to provide precise approximations with designs typical of HCI studies [320]. The same datasets are re-used when varying DECIMALS. We note cases where experiments add other conditions as appropriate.

##### 4.5.1.1 Generation Process

The groups in a dataset are generated by randomly sampling group SIZE values from a standard normal distribution  $\mathcal{N}(\mu, 2)$  where  $\mu$  is set to 0 for 50% of groups (to simulate groups having equal means). Otherwise,  $\mu$  is drawn randomly from  $\mathcal{N}(0, 1)$ . For within-subject data, a random intercept is added per subject. The random intercept is unique to the subject and randomly drawn from  $\mathcal{N}(0, \sigma^2)$  where  $\sigma^2$  is randomly chosen to be either 0.1, 0.5 or 0.9, following the values proposed by Elkin et al. [99] to represent a “reasonable ratio between within-subject variance and between-subject variance”.

#### 4.5.2 Metrics

We report standard metrics used in statistics literature to validate CIs (coverage probability [356]) and p-values (type 1 error rate and power [99]). Recall

<sup>6</sup> <https://ns.inria.fr/loki/statslator/>

that our goal is to validate the equations presented in section 4.4. Thus, we consider a conversion as “correct” if it recovers a p-value or CI that yields a similar score with these metrics than the corresponding p-value or CI calculated from the raw data. Below, we clarify the meaning of each metric.

#### 4.5.2.1 Coverage of CI

The coverage corresponds to the proportion of calculated CIs that contain the true population mean. For a confidence level of 95%, this proportion should be as close as possible to 95%. This means that over a large number of repetitions, we expect 95% of the 95% CIs to contain the true population mean.

#### 4.5.2.2 Type I error rate

Given a significance level  $\alpha$ , the type 1 error rate is the proportion of false positives where a true null hypothesis is rejected ( $p < \alpha$ ). We use a significance level of 0.05 that is common in HCI, so this proportion should be as close as possible to 5%. Over a large number of repetitions, we expect 5% of p-values to be below 0.05 (even though they are type 1 errors).

#### 4.5.2.3 Power

Statistical power is the proportion of true positives where a false null hypothesis is correctly rejected ( $p < \alpha$ ). The closer the power is to 100%, the more statistically powerful the test. In practice, power can be much lower, especially when the sample size is small.

### 4.5.3 Experiment 1: t-test reports to CIs

This first experiment simulates conversions to obtain 95% CIs from reports comparing two groups with a t-test. In total, 240,000 reports are simulated ( $4 \text{ SIZES} \times 2 \text{ DESIGNS} \times \text{DECIMALS} \times 10,000 \text{ repetitions}$ ). For each report, the 95% CI is calculated using either t-score and means (section 4.4.1.1); p-value and means (section 4.4.1.2); or means and standard deviations (section 4.4.1.3). As baseline, we report the coverage of the 95% CI calculated using the t-distribution of the sample raw data.

#### 4.5.3.1 Results

On average, three decimals is enough, two decimals give reasonable estimates, but one decimal yields narrower and overoptimistic CIs that do not capture 95% coverage. Table 4.1 shows coverage of CIs depending on conversion method, the study design, and the numeric precision of the values used. Perhaps because rounding errors propagated, “means + stds” produced the least accurate results with a coverage of 93.8% at two decimals and 90.4% at one decimal.

Table 4.1: Mean coverage (and standard deviations) of the 95% CIs calculated from different conversion equations and for different DESIGNS, and DECIMALS. The closest the values are to the one obtained from raw data, the better.

Values Used	DESIGN	DECIMALS		
		1	2	3
t-score + means	within	.918 (.27)	.947 (.22)	.950 (.22)
p-value + means	within	.927 (.26)	.948 (.22)	.950 (.22)
raw data	within	.950 (.22)	.950 (.22)	.950 (.22)
t-score + means	between	.917 (.28)	.947 (.22)	.950 (.22)
p-value + means	between	.927 (.26)	.948 (.22)	.950 (.22)
means + stds	between	.904 (.29)	.938 (.24)	.942 (.23)
raw data	between	.950 (.22)	.950 (.22)	.950 (.22)

#### 4.5.4 Experiment 2: Corrected p-values to CIs

This experiment simulates reports of multiple pairwise t-test comparisons as would be done post-hoc after an omnibus test, such as ANOVA. It differs from experiment 1 in that one report may contain 3, 6, or 10 COMPARISONS (corresponding to an independent variable with either 3, 4, or 5 levels) and the p-values are adjusted to counteract the multiple comparisons problem. The goal is to evaluate the impact of these corrections on the calculated CI, and test the approximations to “unadjust” them. In total, 2,160,000 reports are simulated (3 COMPARISONS  $\times$  3 CORRECTIONS  $\times$  4 SIZES  $\times$  2 DESIGNS  $\times$  3 DECIMALS  $\times$  10,000 repetitions). The CORRECTION applied to the p-values is either Bonferroni [262], Holm-Bonferroni [164] (that we refer to as Holm to avoid confusion), or Šidák [355]. As baseline comparison, we report coverage for 95% CI calculated using the t-distribution of the sample raw data.

##### 4.5.4.1 Results

On average, corrected p-values tend to increase the confidence level of the CI to match a 99% CI. For reasons mentioned in section 4.4.4.1, reversing a correction is approximate and works best for Šidák. For Holm, the reversed correction results in recovering lower CIs (down to 92% confidence) and recovering the CI from the t-score should be preferred. Table 4.2 reports the breakdown of coverage for the different corrections and uncorrected p-values.

#### 4.5.5 Experiment 3: CIs to p-values

This experiment simulates reports that include 95% CI and for which we would like to recover p-values. In total, 720,000 reports are simulated (3 CI METHOD  $\times$  4 SIZE  $\times$  2 DESIGN  $\times$  3 DECIMALS  $\times$  10,000 repetitions). The CI METHOD to calculate the 95% CI is either the t-distribution (referred to as t-CI),

Table 4.2: Mean coverage (and standard deviations) of the 95% CIs calculated from p-values adjusted using a different CORRECTION. For uncorrected values, the closest the values to the the one obtained from raw data, the better.

CORRECTION	DECIMALS		
	1	2	3
Bonferroni	.968 (.18)	.988 (.11)	.990 (.10)
Šidák	.968 (.18)	.988 (.11)	.990 (.10)
Holm	.964 (.19)	.984 (.12)	.986 (.12)
Bonferroni uncorrected	.938 (.24)	.957 (.20)	.959 (.20)
Šidák uncorrected	.931 (.25)	.951 (.22)	.952 (.21)
Holm uncorrected	.916 (.28)	.925 (.26)	.925 (.26)
raw data	.950 (.22)	.950 (.22)	.950 (.22)

Table 4.3: Mean type 1 error rate (and standard deviations) of the p-values calculated from the 95% CI obtained using different CI METHODS and varying DECIMALS precision. The closest the values to the the one obtained from raw data, the better. Coverage of the 95% CI provided for reference.

CI-METHOD	Coverage	DECIMALS		
		1	2	3
t-CI	.948 (.22)	.047 (.21)	.051 (.22)	.051 (.22)
percentile-CI	.935 (.25)	.078 (.27)	.083 (.28)	.083 (.28)
studentized-CI	.944 (.23)	.049 (.22)	.055 (.23)	.055 (.23)
bCA-CI	.935 (.25)	.075 (.26)	.08 (.27)	.081 (.27)
raw data	-	.051 (.22)	.051 (.22)	.051 (.22)

or popular bootstrapping methods such as percentile CI [96] (percentile-CI), the studentized CI [82] (studentized-CI), or the bias-corrected and accelerated CI [97] (BCa-CI). Bootstrapping methods use 2,000 resamples. As baseline, the results for the p-value obtained from appropriate Student's t-tests are reported.

#### 4.5.5.1 Results

Overall, p-values recovered from BCa and percentile bootstrapped CIs tend to inflate the type 1 error rate, but are statistically more powerful. This increased number of type 1 errors might be explained by these methods generating CIs with a coverage that does not match 95%. This finding is consistent with previous work that found the percentile and BCa method to perform poorly given small samples ( $N < 50$ ), whereas at this size the t-distribution or the studentized bootstrap is usually best [98, 421]. For comparison, our experiments use sample sizes between 8 and 40.

Table 4.4: Mean power (and standard deviations) of the p-values calculated from the 95% CI obtained using different CI METHODS and varying DECIMALS precision. The closest the values to the the one obtained from raw data, the better. Coverage of the 95% CI provided for reference.

CI-METHOD	Coverage	DECIMALS		
		1	2	3
t-CI	.948 (.22)	.300 (.46)	.307 (.46)	.307 (.46)
percentile-CI	.935 (.25)	.353 (.48)	.36 (.48)	.361 (.48)
studentized-CI	.944 (.23)	.295 (.46)	.306 (.46)	.307 (.46)
bCA-CI	.935 (.25)	.349 (.48)	.356 (.48)	.357 (.48)
raw data	-	.307 (.46)	.307 (.46)	.307 (.46)

P-values recovered from CI calculated from a t-distribution and studentized bootstrap tend to match the p-values that would have been obtained had a t-test been run, even at low one-decimal numeric precision. Table 4.3 shows the breakdown of type 1 error rate given the different CI methods and number of decimals. Table 4.4 shows the same breakdown for statistical power. Note that power may appear low, but this is consistent with what is expected, and has been shown before for such small sample sizes [225].

#### 4.6 STATSLATOR PDF VIEWER

We developed the Statslator PDF viewer for readers to interactively translate between statistical reporting presentation styles in existing documents and generate statistical complementary presentations like effect sizes and graphical charts. The tool can help readers interpret statistical reports. The user interface was designed to make the capabilities of the conversion equations

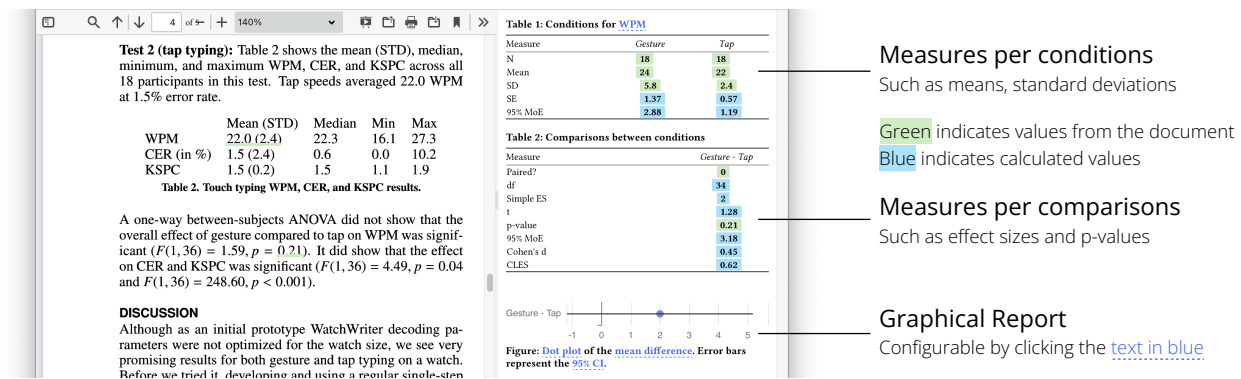


Figure 4.3: Statslator user interface after selecting a paragraph with statistical information from WatchWriter [134]. The panel on the right shows the statistical measures extracted and calculated, as well as a configurable graphical report.

in our library transparent to the reader, so they are aware of the provenance of the data, what calculations are done, and the quality of conversions.

Readers use Statslator to open and view a PDF document, then select content with statistical reporting they wish to translate into a new presentation style or complementary presentation. The new presentations appear in a sidebar, are highly configurable, and present related statistical values to explore and validate, or values that are easier to interpret correctly such as the common language effect size [196]. The reader can also choose from different representations, such as animated hypothetical outcome plots [172] and interactive plots, which, despite being powerful representations, are unlikely to be found in existing documents because of publishing formats and workflows. We describe the tool and its features in more detail using three use case scenarios.

#### 4.6.1 Changing the Style of Statistical Report

Sam got a new smartwatch and decides to review the literature to find the best text entry method for this device. Sam first stumbles across “Watch-Writer” [134], an article describing a keyboard for smartwatches. The article reports a user study that compares two ways of operating the keyboard either through taps or gestures. At first glance, it appears that the gesture version is preferable. The article reports “A one-way between-subjects ANOVA did not show that the overall effect of gesture compared to tap on WPM was significant ( $F(1, 36) = 1.59, p = 0.21$ ). It did show that the effect on CER and KSPC was significant ( $F(1, 36) = 4.49, p = 0.04$  and  $F(1, 36) = 248.60, p < 0.001$ ).”<sup>7</sup> However, Sam is not very familiar with NHST, and the article does not report effect sizes making it difficult to know if it is worth investing time to learn the gesture technique. Thus, Sam selects the text mentioning the statistical results and the table that reports the means for each condition.

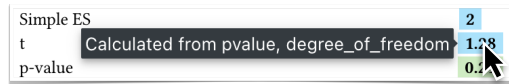
##### *Automatic Extraction & Verification*

After the selection, the panel on the right is updated to display two tables filled with statistical values from the selection or calculated, as well as a figure of the comparison (figure 4.3). The background of the cells are coloured based on the provenance of the information: green indicates that the value was obtained from the text or entered manually; and blue indicates that the value was converted. Sam decides to verify that the extracted data is correct by hovering over each value to obtain detailed information. When hovering over a value obtained from the text, a tooltip shows the sentence where the value was extracted from and the corresponding value in the PDF is underlined.

t	1.28
p-value	0.21
95% CI	3.14

<sup>7</sup> The article mentions a one-way ANOVA with two levels which is strictly equivalent to a t-test and thus is supported by our tool.

When hovering over a value that was converted by our equations, a tooltip shows details of the calculation. The most accurate conversions are prioritized. For example, for the CI the t-score and means are used even though the standard deviations, p-values are also available.

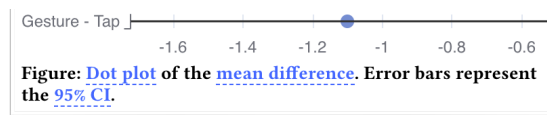


### Switching between Dependent Variables

By default, the generated report shows the results for words per minute (WPM). Sam clicks the blue underlined text in the caption of the first table in the panel to change the dependent variable and show the KSPC. Sam realizes that Cohen's  $d$  for the comparison of KSPC is quite large ( $=5.5$ ) compared to the other dependent variables compared. Sam cannot recall the interpretation of Cohen's  $d$ , but the tool shows the common language effect size ( $=.99$ ) that Sam knows to interpret as "when gesture was compared to tap, in 99 of 100 pairs gesture had a lower KSPC than tap."

### Configuration of the Graphical Statistical Report

Similarly, the figure generated by the system is interactive and shows the 95% confidence interval on the mean difference.



Sam knows about the overlapping rule for the 95% CIs of independent means and decides to change the figure by clicking the blue underlined text in the caption from "mean difference" to "means".



However, after some tinkering, Sam is not sure anymore of the correct interpretation of a 95% confidence interval. Instead, Sam switches the chart to an animated hypothetical outcome plot [172] which, after a few seconds, helps Sam gain intuition for the distribution of the data.



### 4.6.2 Comparing Two Reports

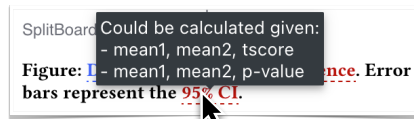
After further inspection, Sam realizes that WatchWriter relies on a statistical decoder which would makes it difficult to enter words that are out-of-



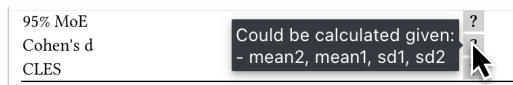
vocabulary. Instead, Sam investigates two alternative techniques that support OOV: SplitBoard [165] and Swipeboard [50]. While both articles present user studies, the two techniques are not compared with each other. Worse, the two studies use different protocols and different study durations, and Sam decides that judgment solely based on the reported means might be misleading. However, both studies include a comparison to ZoomBoard [283], a third text entry technique. To get a rough idea of how the two techniques compare, Sam decides to look at the standardized effect sizes for the two techniques compared to ZoomBoard. Sam had already opened the papers in Statslator, so Sam begins by selecting the paragraphs containing the statistical results in both papers.

#### *Helping the System with Missing Values*

When extracting data from SplitBoard, Sam notices that links for some values are red indicating that they are not available.



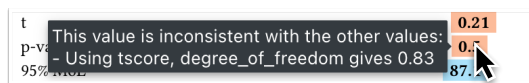
Sam hovers over the missing values in the table to get an indication of what information would be needed for that value to be calculated.



When hovering over Cohen's  $d$ , the system indicates that the means and standard deviation are needed to perform the conversion. However, the SplitBoard paper does not report aggregated means with standard deviations. Sam decides to retrieve these values from the line chart of the WPM using an accurate chart data extraction tool [182, 248]. For SplitBoard, the  $t$ -score was directly obtained enabling the calculation of Cohen's  $d$ . Once the missing values are added in the table, the other values are calculated and Sam can review the effect sizes: Cohen's  $d = 0.64$  for SwipeBoard, and  $d = 7.27$  for SplitBoard.

#### 4.6.3 *Checking Correctness*

While browsing recent preprints, Sam finds a brand new text entry method that looks promising. As always, Sam starts by selecting the statistical report to display the results in a different style. This time, the  $p$ -value and  $t$ -score have a red background: the system detected an inconsistency after cross-checking different ways values can be obtained.



It appears that the  $t$ -score does not match with the reported  $p$ -values. Sam knows how these mistakes can easily occur when writing papers [229, 275] and decides to send an email to the authors to warn them.

#### 4.6.4 Implementation

Statslator is implemented using TypeScript using React [317] with PrimeReact [309] for the interface; PDF.js [288] for the PDF viewer; and ECharts [11] to generate dynamic visualizations. Below, we detail the implementation of the text extraction to recover statistical values.

##### 4.6.4.1 Extracting Statistical Information

Upon selecting a text in a document, the text is extracted and sent to OpenAI's ChatGPT through the official API<sup>8</sup>. ChatGPT takes care of extracting the statistical information contained in the selection, even if the text contains tables and complex sentence structures. ChatGPT is a large language model based on a transformer architecture [390] which are state-of-the-art in many natural language processing tasks, especially when dealing with complex sentence structures [316]. The task of extracting statistical information is no exception: in our tests, ChatGPT outperformed all alternatives. ChatGPT requires a hand-crafted *prompt* that explains the task, we engineered the following prompt through repeated experiments:

```
<excerpt from the paper>
Answer with this JSON structure:
{"conditions": [/* reported numbers that refer to a condition following this JSON format:
  [<number>, <type> /* example: mean, sd, upper CI */, <condition>]*/],
"comparisons": [/* reported numbers that refer to a t-test comparison following this JSON
  format: [<number>, <type> /* example: p-value, t-score */, <condition1>, <
  condition2>*/]}]
```

We use “ChatGPT3.5-turbo” which has an input limit of about 3,000 words, forbidding long text selections<sup>9</sup>

For privacy reasons or because it is a paid service, we also support alternatives to ChatGPT. First, readers can always input the numbers manually. Second, we also provide an extraction algorithm that relies on regular expressions. The algorithm searches for APA symbols such as “M=” to extract statistical values (similar to section 4.3). We group values based on their order of appearance. For example, the first mentioned mean and standard deviation are grouped as a single condition. Similarly, the first p-value is associated with the first comparison of the two conditions mentioned.

## 4.7 DISCUSSION

Our work highlights three aspects: (1) even though most papers report statistics in a specific way, they usually contain enough information to convert them into a different statistical reporting style; (2) most reporting practices are compatible with accurate conversions; and (3) a PDF viewer that embeds these conversions can enable readers to control the presentation of statistics

<sup>8</sup> <https://platform.openai.com/docs/api-reference>

<sup>9</sup> The limit has been increased with newer versions of ChatGPT and full papers could now be parsed directly.

in existing documents to better understand the results, compare documents, and verify correctness. Before outlining the limitations and future work, it is important to clarify what this work is not.

In no circumstances does this work replace the proper practice of statistics by authors, nor reduce claims made by scientists promoting a specific approach to statistics. The differences between an NHST and estimation approach are more than just the presentation of the results. For example, estimation-based thinking also implies a different way of formulating research questions and drawing conclusions which can hardly be an afterthought [44]. And best practices involve planning studies and following open science procedures [58, 71, 88]. Instead, our work helps *readers* desiring a different presentation of statistical results, perhaps to draw their own conclusions. If anything, our hope is that showing that conversions are possible will motivate authors to choose the method most appropriate to them and their research questions [211] without worrying about possible push-back. As expressed by Andy Cockburn in response to an alt.chi article encouraging authors to use estimation and avoid dichotomous reports: “*Sometimes, however, the author would prefer to NOT report dichotomous outcomes (for good reasons), but is compelled to do so by their fear/knowledge that if not included, reviewers will expect it and criticise its absence*” [27].

#### 4.7.1 *Limitations*

##### 4.7.1.1 *The analyses might have missed some papers and study designs*

With large-scale experiments, it is difficult to consider all cases. In section 4.3, when quantifying the proportions of each report at CHI, some papers might have fallen through the cracks, either because they use complex sentence structures, or because the reports are done in figures and tables. Our analysis was meant to motivate the feasibility of conversions and have reference proportions to decide the most appropriate input values for the equations.

Similarly, there are an infinite number of parameters and study designs that could be tested in the experiment section 4.5, but we chose to focus on those most prevalent in HCI. This had the side-effect of steering our experiments and equations towards the use of small samples which have been covered relatively poorly in the past. However, as the sample size increases, many of these considerations are no more relevant considering the central limit theorem [105].

##### 4.7.1.2 *The conversions might fail in some situations*

As a corollary of the infinite space of study designs, we cannot guarantee that the conversion will be correct in all cases. Most statistical tests have underlying assumptions that might in fact be violated by the data and yield unexpected results. Generally, our conversion equations are based on the assumption that the authors respected the assumptions of the methods they used. However, it is not uncommon for scientists to use an inappropriate test [229, 408] or

make mistakes [275, 276], especially considering the challenges associated with statistics in HCI [188]. In these cases, the results will most likely be as incorrect as their original presentation.

#### 4.7.2 Future Work

##### 4.7.2.1 Support for more papers through access to more data

The main hurdle in converting a statistical report is the lack of data. We focused on t-tests and papers reporting means and p-values or t-scores because we found these to be the most common at CHI. Conversion between any statistical report is theoretically possible given access to the raw data. Thus, one extension of our work could be to leverage that data when available. For example, the system could automatically pull data from a repository such as <https://osf.io/>. However, “Open Data” has still a long way to go in communities such as CHI where less than 1% of papers make data available [1, 248]. Other times, the data is in the document but buried inside data visualizations. In these cases, tools to “reverse-engineer” data visualizations (e.g. [182, 248]) could make the system work with a broader set of scientific articles.

*This idea of recovering data from charts is explored in the next chapter.*

##### 4.7.2.2 Support for other conversions

Future work could look into supporting more tests and conversions towards a broader set of reporting styles. Tests such as Wilcoxon signed-rank and Mann-Whitney U would be a natural extension given their prevalence. Of course, the challenge is that the underlying distribution of the data is unknown. Additionally, effect sizes could be further supported for different tests, including  $\eta^2$  for ANOVAs. And a CI on these effect sizes could also be calculated given limited data using approaches such as the noncentrality parameter [73, 366].

In terms of statistical reports, we focused on NHST and estimation because they are fuelling many debates within the scientific community [45, 88]. However, other approaches such as Bayesian statistics could be supported. For example, a Bayesian t-test has been proposed and can be calculated given a t-score and a prior [328]. The prior could be controlled by readers to reflect their optimism and knowledge, similar to what Dragicevic et al. proposed [91].

##### 4.7.2.3 Statistical linting, meta-analyses, and statistical education

There are many use cases that could be derived from our system. First, our focus was on readers, but the mechanisms leveraged to detect inconsistencies could power a statistical linter for authors. Similar to *statscheck* [276] that detects inconsistencies between the reported p-value and reported test statistic, a statistical linter could leverage our system to cross-check the different ways

of obtaining a value. This would allow the detection of serious problems such as using a t-test that does not match the study design.

Similarly, many of our equations could be useful in meta-analyses, especially within fields like HCI that deal with small samples for which typical meta-analyses practices are overoptimistic [8, 9]. Our system could help scientists recover accurate data into a customizable and standardized statistical measure.

Finally, we assumed readers have some experience and preferences regarding statistics, but our PDF viewer could also be used as an educational tool. Transitioning between statistical representations is particularly useful to develop an intuition [71, 74]. Akin to explorable multiverse analyses reports, some options could be educational to give readers a better grasp of certain concepts [91].

#### 4.8 CONCLUSION

While much of the debate around statistics has been focused on how authors should practise and report them, little has been done to support readers and the thousands of documents already published. Through theoretical and empirical evidence, we showed that a majority of CHI papers report enough information to be converted to different statistical reporting styles and that the conversions remain mostly accurate under common reporting practices. We also describe the design and implementation of a PDF viewer to turn existing papers into the statistical reporting style readers prefer. Our hope is to provide an immediate solution to reconcile readers with statistical reports, all while unburdening authors to let them focus on proper statistical practices.

# 5

## EASY AND ACCURATE INTERACTIVE DATA EXTRACTION FROM COMPLEX VECTOR CHARTS

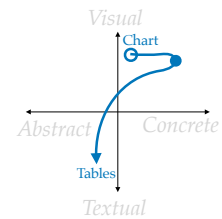
The previous two chapters focused on helping readers with numbers and statistics reported in-text. While textual reports are common, data is often presented with charts causing the previously proposed approaches to fall short. Thus, in this chapter, we investigate ways to help readers recover data presented through charts, and ways to reuse this data to create interactive charts.

### 5.1 MOTIVATION

Charts are often the preferred way of presenting data because they offload cognitive work to the visual system [213, 267]. For readers, accessing the numerical data of charts unlocks a broad range of applications: they can explore the data to better understand it [36, 91], generate new visualizations [256, 319, 337], redesign existing charts [147, 303, 340], answer questions [184, 193, 296], and generate textual summaries [55, 87, 280], as well as make existing visualizations accessible [53, 116], interactive [202, 362], and more informative [201, 209]. Further, researchers need this data to replicate analyses and compare results. However, despite the push for Open Science, not all scientists publish their data. This has been particularly true in the field of Human-Computer Interaction (HCI); an analysis of 509 CHI papers published before 2018 found that data was provided in less than 1% of cases [1]. Much of authors' hesitation comes from privacy concerns and little incentive or perceived benefits [398], suggesting they are unlikely to change their practises. Even if providing data became commonplace, the issue remains for previously published papers for which the data has long been lost.

One solution to recover data is to analyze charts, a practice called “*chart reverse-engineering*” [302]. By carefully locating series and inferring their position in the axis coordinates, the underlying data can be estimated. Of course, the more complex and dense the chart, the more tedious and error-prone the process becomes. Consequently, tools such as ChartSense [182] and Web-PlotDigitizer [324] offer semi-automatic features to make the process easier. The core idea is to automate the recovery of the chart's structure, identifying every line, rectangle, and text in the image, and then inferring the element's role such as axis, series or legend.

However, most chart reverse-engineering approaches rely on *what can be seen* using pixels, and all previous tools operate only on raster images. Yet, vector graphics are commonly shared on the web [21] or in scientific publications [54, 55]. In fact, publishers often recommend, or even require the use of vector graphics for their scalability without loss of resolution [174, 361]. As a result,



In our design space  
fig. 1.1,  
ChartDetective  
recovers a table of  
numbers from charts  
and reconstructs  
interactive charts  
with the data

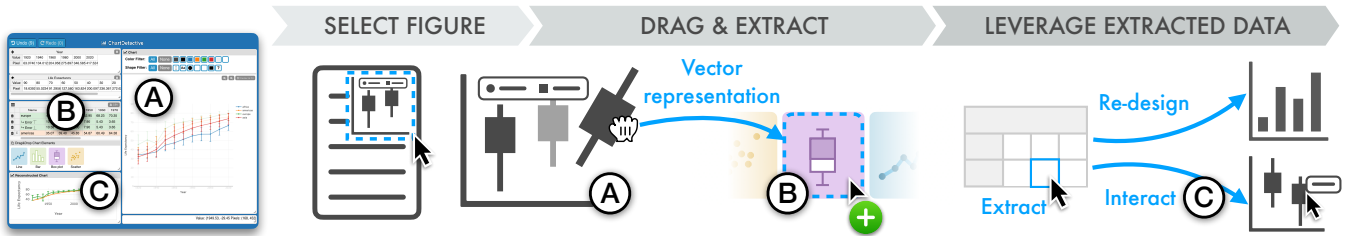


Figure 5.1: ChartDetective is a system capable of recovering a chart underlying data by leveraging its vector representation. Users select figures from PDF documents and then (A, B) drag-and-drop elements that they wish to extract. (C) The extracted data can be leveraged for downstream tasks such as redesigning or interacting with the figure.

vector charts are ubiquitous, but their potential benefits are lost when existing systems rasterize them [81, 182, 304, 387].

This focus on raster images is a missed opportunity to improve chart reverse-engineering tools in terms of accuracy, usability and performance. With raster graphics, image resolution limits the quality of the data. Even assuming perfect accuracy from the recognition system, information is irreversibly lost, either because pixels do not capture the full resolution of the original data or because an element such as the legend hides part of the information. A pixel in a raster chart may represent a fraction of a unit or millions of units, even if the original data provided much finer resolution. The problem is aggravated when the raster image has compression artifacts or when the chart is dense with overlapping elements. In contrast, charts embedded as vector graphics are ideal for reverse-engineering because they encode the complete image structure and reference all components, even if hidden or overlapped, with an exact position and size.

Leveraging the extra benefits provided by the vector format is challenging because it requires understanding the specifics of the file format, knowledge of how the chart was generated, and the ability to access and operate vector graphics editors. In fact, little is known about how to recover the data from vector charts, how accurate the extracted data is compared to using raster images, and if the extra information encoded by the format can help the reverse-engineering process. To the best of our knowledge, only Choudhury et al. [54] describe an approach for separating curves from vector line charts. All other approaches focus exclusively on raster images.

In this chapter, we introduce ChartDetective, a tool to extract underlying data from charts by leveraging their vector specification (Figure 4.1). The approach creates an interactive pipeline to extract data from a chart: a chart in a vector format, such as SVG and PDF, is processed and presented in a user interface where its underlying data is extracted using an integrated set of interactive selection, filtering, and previsualization mechanisms. Leveraging vector information has several advantages, enabling: novel features (e.g., filtering mechanisms); support for a wide variety of charts such as bar, line, scatter, and box plots; data recovery with greater accuracy and precision than other approaches; and extraction of charts exhibiting challenging characteristics

such as diverse styles, thousands of data points, multiple encodings, and occluded elements.

Our work makes the following contributions: (1) Highlight of the advantages of vector graphics over raster images from a theoretical perspective. (2) Design and implementation of a tool to extract underlying data from vector charts demonstrating how using a vector representation enables new features and results in high-quality underlying data. (3) Experiment results showing the system is usable when extracting charts with challenging properties from real scientific publications. (4) A technical evaluation using a dataset of synthetic and in-the-wild charts validating superior accuracy of extracted data compared to existing approaches for raster images.

## 5.2 BACKGROUND AND RELATED WORK

Charts can be represented in two formats: raster images or vector graphics. Below, we review how data can be recovered from both formats and what are the theoretical advantages of vector graphics.

### 5.2.1 *How Can Data be Recovered From Charts?*

If not readily available, data can often be partially recovered from charts as cleaned and aggregated data subsets. Consider how an author creates a chart: first, a chart is generated using visualization tools such as matplotlib, ggplot, excel, or tableau in order to turn tabular data into a visualization like a bar chart that readers can quickly comprehend. The visualization is then exported either as a rendered image in a raster file format (e.g., PNG, JPG, BMP) or re-encoded into a vector file format (SVG, EPS, PDF), and shared by being included in a document or a web page. Recovering the data visualized by a chart is later accomplished by identifying each data point as a shape with a location and size, and transforming those into the local coordinate system defined by the chart axes. With vector charts, the position and size of each shape are recovered from the definition of vector graphics. In contrast, for raster images, the information has to be measured.

### 5.2.2 *Advantages of Vector Graphics*

There are several characteristics of vector graphics that make them advantageous for extracting chart data.

#### 5.2.2.1 *Higher Theoretical Precision*

Because of how raster and vector graphics encode information, the precision of the data should be higher for vector graphics.

*Raster images* are composed of pixels. Given a raster image with one linear axis representing  $n$  units displayed on  $p$  pixels, one pixel represents  $n/p$  units. For example, a linear axis ranging from 0 to 1000 displayed over 100



pixels means that each pixel represents  $1000/100 = 10$  units. Thus, a value of 0 is indistinguishable from a value of 9 because they are the same pixel. In other words, achieving a high accuracy when recovering the position of an element in a raster image requires a comparatively high resolution, inevitably increasing the size of the image file.

*Vector graphics* define shapes in real number coordinates. Thus, precision is limited by the number of decimals used to define coordinate positions and rounding errors due to coordinate transformations. To accommodate 32-bit processors, the PDF format uses the “single-precision floating-point format” and limits floating point numbers to approximately five decimals (ISO 32000-1:2008§C.2). The SVG file format encodes coordinates decimal numbers in strings and does not limit the number of decimals<sup>1</sup>. As for transformations, both PDF and SVG specifications recommend using double-precision floating-point numbers (ISO 32000-1:2008§7.10.5.1) to reduce rounding errors when rendering. However, document viewers perform these operations and could use a higher precision format if needed. Thus, data values encoded in PDF charts can be theoretically recovered with up to five decimals with no impact on file size (the floating value will occupy 32 bits regardless of the number of decimals). The precision of values extracted from SVG charts is theoretically not bounded. Returning to our previous example, obtaining the same five-decimal level of precision with a raster chart ranging from 0 to 1000 would require 100 million pixels.

#### 5.2.2.2 *Recovery of Occluded Data*

The vector graphic can include all geometry in a visualization, regardless of what is viewable in a final rendering. In particular, by default occluded shapes are included, even if completely hidden when rendered. This enables the recovery of occluded data, a common issue with many charts [79, 182]. For example, a legend often hides part of a series, line series might overlap or cross each other, or a dense scatterplot might have clusters of indistinguishable points due to stacking.

#### 5.2.2.3 *Reduced Ambiguity*

Classifying the role of elements in a chart is challenging in general [80, 182, 302] and raster image compression makes this even more difficult. For example, artifacts like blurry edges and irregular fill colours make elements hard to automatically separate. Additionally, rasterized text such as alphanumeric labels and annotations has to be located and recognized using OCR. With vector graphics, shapes are clearly identified and text is often directly accessible.

---

<sup>1</sup> <https://www.w3.org/TR/SVG/>

#### 5.2.2.4 Ubiquitous

Most charting tools offer to export in vector formats, and this format is commonly used to share charts. On the web, the SVG format is natively supported and often preferred in a context in which pages are rendered on different screens of different sizes. Specifically, charts are commonly shared online in the SVG format [21]. Because of their scalability properties, vector graphics also represent a substantial proportion of all charts included in documents. For example, the popular PDF format allows the inclusion of different types of content such as text, fonts, raster images, and, specifically vector graphics [247]. In fact, major publishers such as IEEE [174] and Springer [361] recommend the use of vector graphics, because “*Creating and saving your graphics in vector format will ensure that your graphics appear as clearly as possible in your final published article*”, and “*Vector graphics (rather than rasterized images) should be used for diagrams and schemas whenever possible*”.

Quantifying the proportion of vector charts shared on different platforms and medias is difficult. As an example relevant to the HCI community, we counted<sup>2</sup> that vector charts represented 38% of the 5,855 charts published in the last six years (2015-2021) of proceedings at the Conference on Human Factors in Computing System (CHI). In a similar analysis, Choudhury et al. found up to 70% of vector charts across the top-50 computer science conferences spanning all fields [54].

#### 5.2.3 Involving Users to Improve Accuracy of Chart Data Extraction

A large body of work looked at fully-automatic pipelines for chart extraction, see Davila et al. [79] for a recent survey. While a fully automatic approach might be desirable, Davila et al. found that most approaches struggle when faced with charts in-the-wild. They list common characteristics of charts from PubMedCentral papers<sup>3</sup> noting “*despite being common, none of the works covered here dealt explicitly with these and other chart complexities*”. In fact, this motivated a chart mining competition held annually since 2019 [78]. Yet, as of 2022, the best approach (relying on large deep-learning models [241]) could recover only 69% of the data from charts in-the-wild, and the accuracy of the recovered data is not reported [80]. These poor performances may be attributed to the great diversity of charts [182] and the difficulty to obtain large annotated datasets, forcing automatic approaches to use artificial datasets and limit their scope to specific chart-styles and encodings. As a result, these approaches can fail when charts deviate even slightly from the training dataset [79].

When fully automatic approaches fall short, a common solution is to resort to manual or semi-automatic approaches to chart extraction [330]. Manual approaches to chart extraction such as Digitize [304] and Ycasd [137] rely on human annotations: after a calibration step to define the axes, the user needs

<sup>2</sup> We manually annotated which figures were charts after extracting all figures in the six years of CHI papers. We calculated the proportion of those that did not contain a single raster element.

<sup>3</sup> <https://www.ncbi.nlm.nih.gov/pmc/>

to click on every data point in the chart. Semi-automatic approaches provide tools relying on computer-vision to facilitate and speed up manual extraction. For example, WebPlotDigitizer [324], Engauge Digitizer [119], and DataThief III [387] include automatic selection tools based on masking and colour filtering. The parameters of the underlying algorithms (e.g., curve fitting, blob detector, line tracing) can be tweaked for better results. ChartSense [182] goes one step further by automatically extracting marks from a chart and requiring users to only specify critical features like the y-axis, and to check and correct the automatic selection. By involving users, these tools support a larger variety of charts.

In this work, we also use a semi-automatic approach to support a greater diversity of charts. However, our algorithms leverage structural information in vector graphics for more accurate extraction.

#### 5.2.4 Leveraging Structured File Formats

Charts can be embedded using formats with more structure than raster images (e.g., HTML, D3.js). Perhaps because of their ubiquity, most of the work on chart data extraction revolves around rasterized charts. Indeed, any format displayable on-screen can be trivially converted to a raster image. However, for the task of chart extraction, the structure will need to be recovered through an often imperfect vectorization step.

Instead, others have circumvented this vectorization step by using formats which preserve structure and semantics. For example, D3 [34] and Vega-Lite [338] directly embed chart data and specifications, making it possible to redesign an existing visualization [147], create re-usable styles [148], search visualizations based on structure and style [166], answer questions [193] and generate visual explanations [193]. However, these approaches are limited to charts embedded in these specialized formats and do not support the broader spectrum of charts in formats such as PDF or SVG.

Although the semantic role of each shape is lost, the vector format constitutes a middle ground as it provides precise information about chart geometries and vector charts are widely used [21, 54]. Previous work leveraged the vector formats to classify visualizations [21, 348], create visualizations [256], retrieve visualizations based on their structure [227], and generate chart animations [119]. While some of these work try to recover information about visualizations including charts, they are not concerned with obtaining the precise underlying data. Instead, they use simplifications, for example, by assuming the data is already available [119] or by recovering only high-level characteristics [21, 348], often sufficient to accomplish their goals. Closest to our work, Choudhury et al. [54, 55] proposed a fully automatic pipeline that extracts information from line graphs in a PDF to generate natural language summary descriptions. However, as is common with automatic pipelines [79], their solution relies on strict assumptions. For example, the approach assumes each line series has a unique colour, axes lines are close to the image boundary, tick marks intersect with axes lines, and legends are close to curve paths.

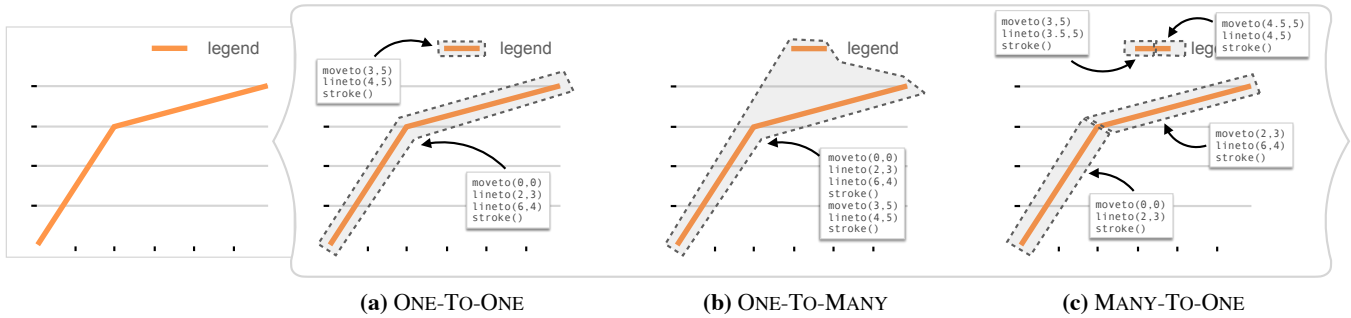


Figure 5.2: The same chart can be formed with different vector shape arrangements: (a) ONE-TO-ONE when shapes match semantic elements; (b) ONE-TO-MANY when one shape maps to multiple semantic elements; and (c) MANY-TO-ONE when many shapes map to a single semantic element.

In this work, we also focus on vector charts. However, we avoid making strict assumptions about chart layout and design. Instead, we adopt a semi-automatic approach to recover data from diverse charts using different styles (line, bar, scatter, and box plots).

### 5.3 CHALLENGES AND DESIGN GOALS

We first summarize the main challenges when extracting data from vector graphics, then propose a set of system design goals. We use these to drive the design of ChartDetective, a new system leveraging vector graphics specifications to extract data from charts.

#### 5.3.1 Challenges of Vector Chart Extraction

When Jung et al. [182] designed ChartSense to extract data from rasterized charts, they faced three main challenges: 1) chart styles are diverse; 2) visual entities can overlap; and 3) there is no off-the-shelf solution for text-region-detection. While reverse-engineering vector charts help with some of these challenges (see Section 5.2.2), some remain and new ones arise.

*C1: Chart Diversity* – Charts vary in the way data is represented as graphical shapes and style. To better understand this diversity, we manually reviewed and annotated the 5,855 charts we extracted from the proceedings of CHI from 2015 to 2021. The majority could be classified into 12 categories: bar chart (43.3%); line chart (25%); scatter plot (9.6%); box plot (9.4%); stacked bar (9.3%); heat map (0.9%); pie chart (0.8%); violin chart (0.7%); density plot (0.6%); radar chart (0.4%); and stacked density plot (0.1%). Some combined different encodings (2%), for example a bar chart combined with line series, or a box plot using scatter points. While difficult to quantify, we observed many variations in style, such as embellished charts [20], diverse colour palettes, annotations, and overlays. Tools to extract data from charts require flexibility to adapt to this diversity.

*C2: Inconsistent Vector Specifications* – A raster image is the result of exactly one configuration of pixels but vector graphics can be generated from a theoretically infinite number of shape arrangements. We define *shape* as a single geometric shape defined in a vector graphics language and *chart element* as a semantic element in a chart (e.g., a series, an axis, a legend). In our exploration, we found three relationships between shapes and chart elements (Figure 5.2).

- ONE-TO-ONE is when each shape maps to a unique chart element (e.g., a line maps to a series).
- MANY-TO-ONE is when multiple shapes represent a single chart element. For example, there were two common ways of representing dashed-lines: applying a “dashed-line” style to a line or using several smaller lines, one per “dash”. Even contiguous chart elements like a line series are not necessarily encoded as one polyline. For example, matplotlib has a tendency to split a single line series into smaller connected lines.
- ONE-TO-MANY is when a single shape represents multiple chart elements. The vector format is flexible enough to allow the definition of disconnected shapes (by using a *moveto* primitive when defining the path). This behaviour is often exploited to draw all the bars from a bar chart using the same shape, or drawing the legend and the series at the same time.

The challenge thus becomes how to divide or group shapes to get a one-to-one mapping in order to match humans’ perception of a single shape and make the extraction of data possible.

*C3: Hidden Shapes* – Vector graphics may contain shapes that are invisible in the rendered image such as shapes occluded by other shapes. However, some *hidden shapes* are meaningless and introduced by mistake using vector editing tools such as Inkscape. For example, we found text and annotations completely occluded by other shapes. They serve no purpose because they are invisible when rendered, and most likely result from mistakes. We also observed several examples of shapes hidden by modifying their colours instead of being removed. For instance, a user might hide axes or grid lines by setting the stroke and fill colour to match the background. However, these shapes remain in the vector specification. Of course, these hidden shapes should be ignored, but identifying them systematically is difficult because they take various forms.

*C4: Rendered Text* – Text in vector graphics can be specified using text-specific vector graphics command to position a string of characters using attribute like font and size or by forming letters using geometric shapes. For the latter, the text cannot be directly recovered as each letter is represented visually not semantically. In that case, identifying text regions and extracting the text then becomes as challenging as with raster images.

### 5.3.2 Design Goals

Using those main challenges, we formulate a set of design goals to guide the development of our system. They consider previous limitations regarding the lack of flexibility and the poor performance when faced with charts *in-the-wild* [78, 79], leverage the advantages of semi-automatic rather than fully-automatic solutions [182, 330], and follow recommendations for mixed-initiative systems [168, 273].

*D1: Maximize Data Accuracy* – For reliability and repeatability, high accuracy means a low relative error between the extracted values and ground truth values in the original data. Previous work seldom reports accuracy, yet accuracy was necessarily limited by the resolution of the raster images [79]. We consider accuracy as the utmost priority and aim to leverage vector graphics to obtain high-fidelity data.

*D2: Support Diversity* – Across various forms of charts (see Challenge 1), flexibility is required to support different ways of encoding data (e.g., line, bar, scatter, box) and variations in style (e.g., colour, size, shape, organization). In practice, this means making few assumptions [79], and likely incorporating user interaction in the extraction process to disambiguate alternative extraction outcomes.

*D3: Minimize User Interaction* – While a fully automatic approach would be ideal, in practice, the user has to be involved—if only to check that the result is correct. Previous work can be placed on a continuum from fully manual [137, 304] to semi-automatic [81, 173, 182, 245, 324, 387] to fully automatic [79]. Our goal is to minimize user involvement by automating tedious and long tasks.

*D4: Simplify Verification* – Checking data extracted from a large, dense chart could entail verifying thousands of data cells. Users should be able to quickly check that the chart was accurately extracted, identify mistakes (if any), and correct them.

## 5.4 CHARTDETECTIVE

ChartDetective is a system to extract underlying data from vector charts by leveraging the vector information. A live version of ChartDetective is accessible online: <http://ns.inria.fr/loki/chartdetective>. Below, we detail ChartDetective’s interface and functionalities. The functionalities try to tackle each challenge and design goal identified in 5.3. As such, direct references are added in parenthesis whenever a functionality responds to a challenge or design goal.

### 5.4.1 Interface

ChartDetective has two interfaces: one to upload a file or document and select a chart to extract, and one to extract data from a chart. The data extraction

*Feel free to try the tool on this thesis document. All charts are in the vector format.*

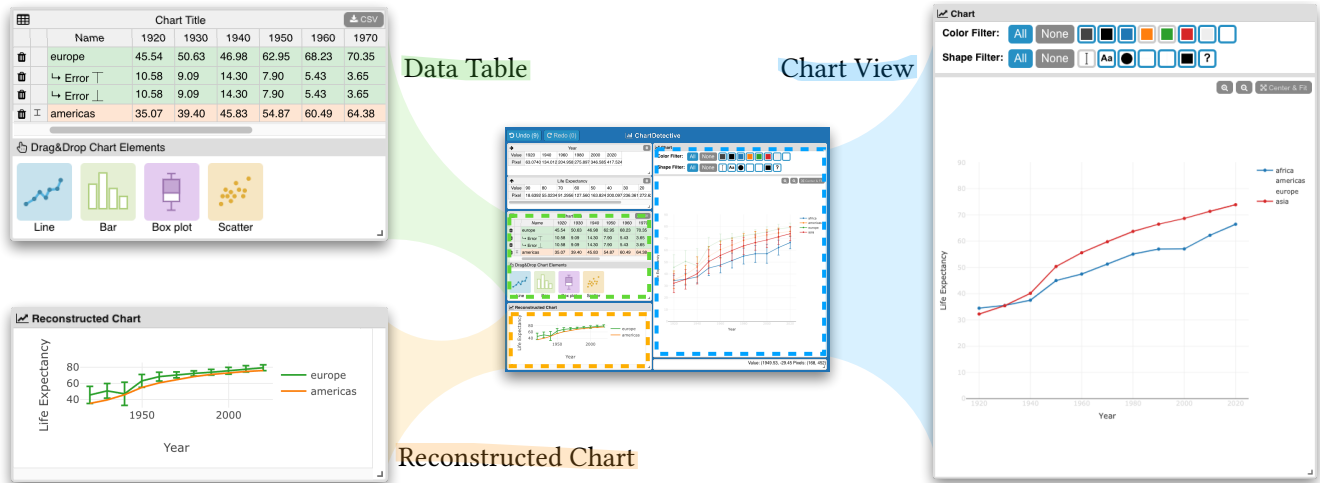
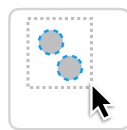


Figure 5.3: ChartDetective is composed of three views: the Chart View showing the chart being extracted; the Data Table with the tables for X and Y axes and the extracted values; the interactive Reconstructed Chart.

interface (Figure 5.3) consists of three main views: 1) The **Data Table** displaying the data extracted from the charts so far; 2) The **Chart View** showing the chart undergoing data extraction; and 3) the **Reconstructed Chart** which recreates portion of the original chart using data extracted so far. The interface deliberately presents the information in multiple views [400]; all views are showing at all times, side-by-side, and the interface can be re-arranged by dragging and resizing the three views to adapt to different screen resolutions.

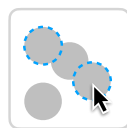
#### 5.4.2 Selection of Chart Elements

The selection of an element in the chart initiates the extraction process. ChartDetective proposes several ways to perform a selection even when the targets are small or occluded (C2, C3).



**Simple Selection** – Using the **Chart View**, a user selects shapes composing the chart by either clicking on them *one-by-one* or by using a marquee selection through a mouse-dragging motion for multiple selection and small hard-to-select objects. As is common with vector software, users can also add and remove elements from their selection by holding the **shift** key.

Shapes under the cursor or included in the marquee selection, are highlighted to preview the selection. A blue animated dashed outline highlights shapes because it is salient for shapes of different sizes and colours. Once selected, shapes are grouped, surrounded by a blue rectangle, and become draggable.



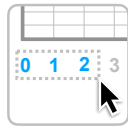
**Fine-grained Selection** – While simple selections work well for basic charts, selecting elements becomes tedious and slow as the number of data points increases or their size decreases. In a worst-case

scenario, selection may be impossible for overlapping elements (C<sub>3</sub>). There are two mechanisms to help in these situations.

- Users can zoom-in using the mouse wheel and pan by dragging while holding the `space bar`. This helps when small elements are hard to select or distinguish when the full chart is viewed.
- There are also two view filtering mechanisms: a colour filter (Figure 5.4b) and a shape filter (Figure 5.4c). When a chart is loaded, all unique colours and shapes used by the chart are identified and displayed as filter buttons. The user can toggle these colours and shapes to remove or add associated shapes from the rendered chart. This helps particularly with dense charts. For example, users can isolate a specific series in a scatter plot by filtering per shape (e.g., only keeping circle-shaped markers or green-coloured dots). Filters can be combined like a logical “AND” (e.g., to select only red circle-shaped markers, see Figure 5.4d). Once only the elements of interest are left, selection is easier and can be done with a quick marquee selection (D<sub>3</sub>).

### 5.4.3 Extraction of Data

To extract data, ChartDetective relies on drag-and-drop interactions where elements selected in the `Chart View` are dropped in the appropriate area of the `Data Table`. Depending on the drop zone, different algorithms are used to extract and analyze the shapes (D<sub>3</sub>).



*Extract Axes* – Extracting an axis is accomplished by selecting at least two tick marks in the chart, then dropping them on the corresponding horizontal or vertical axis of the `Data Table`. Typically, extracting both the X and Y axis requires two drag-and-drop interactions. Extracting the axis title requires another drag-and-drop by first selecting the title in the chart view, then dropping it on the title of the data table.

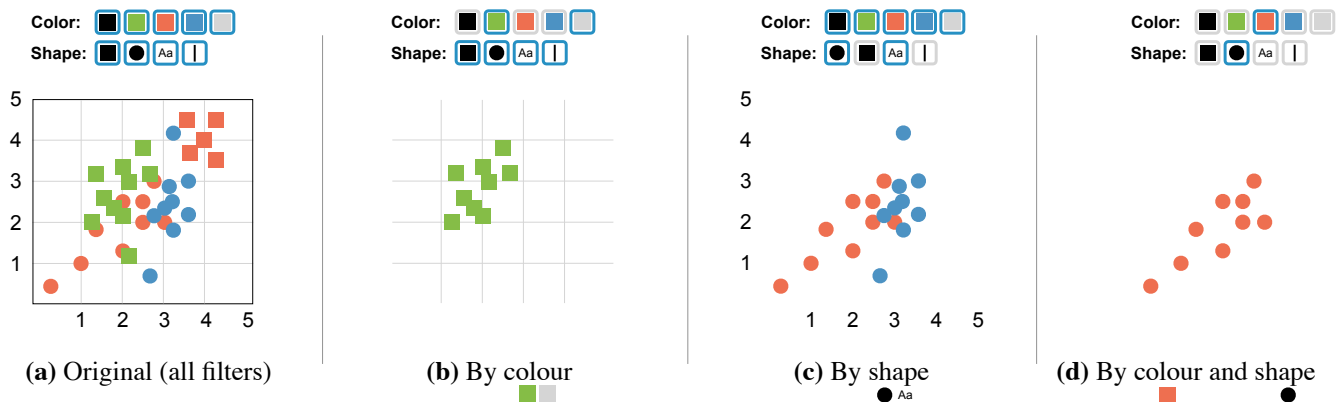
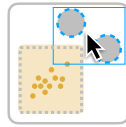






Figure 5.4: (a) Shapes in the can be filtered (b) by colour or (c) by shape. (d) Filters can also be combined.

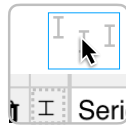




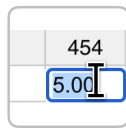
*Extract Data Points* – To add a new series shapes are dropped on one of four zones indicated by a title, an icon, and a colour: line , bar , box plot , scatter . Each drop zone corresponds to a unique encoding of the data. ChartDetective supports four types of data visualization used by the four most common chart types in CHI publications: bar charts, line charts, box plots, and scatter plots (C1). Supporting other visualizations is a matter of writing the corresponding algorithm which takes a vector shape as input and outputs data points.

Once the data is extracted, a new row is added to the table using the colour of the series to be easily identifiable (D4). The corresponding shapes in the [Chart View](#) become translucent and unselectable to allow the selection of shapes potentially hidden behind. These translucent shapes also act as a visual guide to immediately see what remains to be selected.

Whenever possible, ChartDetective automatically mines the name of the extracted series by searching for a legend in the figure. The algorithm works in two steps: 1) find another shape with the same colour as the one extracted; and 2) extract the text at the right of the shape and use it as the name of the series. While this algorithm has obvious failure cases (e.g., black-and-white charts), in practice this high-level assumption is more often correct. Errors can be corrected by dropping the legend directly in the cell indicating the series name.



*Extract Error Bars* – ChartDetective supports the extraction of error values represented as bars or “whiskers”. In ChartDetective, error bars are always linked to an existing series. As such, after extracting a series and adding a new row to the [Data Table](#), the user can select error bars in the chart and drop them on a zone at the left of the corresponding series row. The error bars are matched to series data points based on order and the upper and lower bounds are calculated.

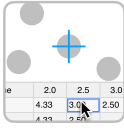


*Modify Extracted Values* – Extracted series and axes in the [Data Table](#) can be modified or removed. The [Data Table](#) is editable like a spreadsheet: clicking a cell edits a value. Data points can also be added to an existing series by first selecting the shapes on the [Chart View](#) and then dropping them directly on the row of an existing series. The added shapes and extracted data points are incorporated into the series.

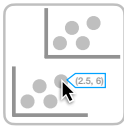
Similarly, the title of all tables and the name of all rows can be edited manually (C4) or updated by dropping text selected from the [Chart View](#). The selected letters are merged to form words and sentences when dropped into the [Data Table](#). If no text glyphs are found in the selection, the text is recovered using an Optical Character Recognizer (C4). This is done by first rendering the selected shapes on a blank canvas before passing it to the recognizer.

#### 5.4.4 Verify Results

Mistakes can happen when extracting the data from charts, for example: data points can be missed, elements may be incorrectly interpreted as data, and legends might be mismatching. In addition to providing a view of the data table and an option to export it as a CSV file, ChartDetective provides passive and active mechanisms to verify the success of the extraction (D1, D4).



*Reconstructed Overlay* – Users can actively check that the data cells match their expectation by examining a data point overlay updated when positioning the mouse cursor above cells in the **Data Table**. Two different overlays are shown: 1) when hovering over axis values, a vertical or horizontal bar shows the extent of the axis where ticks were extracted; 2) when hovering over a data point from a series, a blue cross is rendered at the corresponding position in the chart. This allows the user to verify that a data point is correctly extracted and inspect the mapping between series shown in the chart and series in the data table (D1, D4). For example, to find and fix a potential mismatch in the legend.



*Reconstructed Interactive Chart* – As data is extracted, a second chart is progressively reconstructed in the **Reconstructed Chart** view. To make verification easier, the reconstructed chart shares the same visualization and style such as colours and marker shapes. This allows the user to glance at the **Reconstructed Chart** and compare it with the **Chart View**: a perfect extraction creates a perfect match between the two views (D1, D4). The reconstructed chart is interactive; users can get information on hover (e.g., exact values), hide series, and zoom in on a particular area of the chart. Additionally, the chart can be exported to an HTML file, allowing the generation of interactive charts directly after extracting a static chart.

#### 5.4.5 Getting Started and Interacting

ChartDetective supports traditional and advanced interaction mechanisms in terms of signifiers, feedback, and feed-forward to support exploratory behaviours and help users get started.

##### 5.4.5.1 Discoverability

ChartDetective follows common guidelines to promote discovery such as limiting the number of commands available at any given time, making commands distinguishable, and providing continuous feedback [249]. Because interactions relying on drag-and-drop can be hard to discover [243], we took special care to inform users *when* they could initiate a drag-and-drop interaction and *where* they could drop their selection. New users unaware of the drag-and-drop interface are likely to click one of the icons below the **Data Table**. Doing so opens a ToolTip [140] showing a brief explanation and animation of the drag-and-drop interaction to extract new series. Addition-

ally, when a drag-and-drop interaction is initiated, possible drop zones are highlighted based on the shapes being dragged (Figure 5.5). For example, a drag selection of text elements causes an overlay over all zones accepting text like the table title and series' names (Figure 5.5B). Conversely, a drag selection containing shapes will only highlight zones accepting shapes, and hovering over a zone accepting only text turns the zone red and the pointer becomes a "prohibition sign" (Figure 5.5C) to mark the zone as invalid.

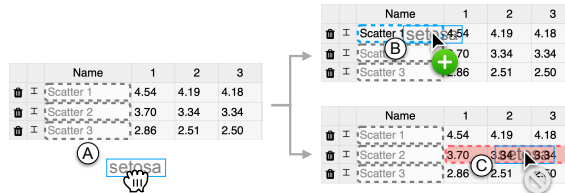


Figure 5.5: During drag-and-drop interactions, (A) drop zones compatible with the selection are highlighted to indicate *where* elements can be dropped. (B) The drop zone turns blue if hovered with a selection of the proper type, and (C) red otherwise.

#### 5.4.5.2 Safe Exploration

The interface is designed so that users understand the consequence of their actions and that all actions can be undone. All commands provide detailed feedback after being executed through notifications at the bottom of the screen. For example, when the data extraction fails, a message is shown to indicate what might be the reason (e.g., "too few shapes in the selection."). Additionally, there is a complete undo/redo mechanism to recover from any action.

#### 5.4.6 Implementation

ChartDetective is implemented in TypeScript using React<sup>4</sup> for the interface, Plotly.js<sup>5</sup> to reconstruct an interactive version of the chart, Tesseract.js<sup>6</sup> to recover rendered text, and PDF.js<sup>7</sup> to parse and render PDFs. While there are multiple vector formats, internally, we use the PDF representation as it is the most low-level and any vector format can be trivially converted to PDF. As such, ChartDetective supports all PDF documents conforming to the ISO-32000 (PDF) specification, and also natively support the SVG format using `svg2pdf`<sup>8</sup>. All modern web browsers are supported. The full source code is available online: <http://ns.inria.fr/loki/chartdetective>.

<sup>4</sup> <https://reactjs.org/>

<sup>5</sup> <https://plotly.com/javascript/>

<sup>6</sup> <https://tesseract.projectnaptha.com/>

<sup>7</sup> <https://mozilla.github.io/pdf.js/>

<sup>8</sup> <https://github.com/yWorks/svg2pdf.js/>

#### 5.4.6.1 *Access to Vector Specifications*

We modified PDF.js to store and provide low-level vector graphics commands after parsing the PDF. This includes retrieving the full list of shapes forming each page of a PDF and determining their final location and size after processing all transformations, group positioning, clipping, and buffered rendering. This allows the selection of a sub-part of the PDF by only keeping shapes completely within a defined area.

#### 5.4.6.2 *Pixel-Perfect Selection*

Because shapes can take complex forms, we implemented selection using a “hit-test buffer” for pixel-perfect selection with little computational cost. This means charts are rendered twice: once to show a preview and once in an off-screen buffer in which each shape is assigned a unique colour. Shape selection is achieved by retrieving the colour of the pixel underneath the pointer. The hit-test buffer is only redrawn when absolutely necessary such as a change of zoom or when a shape filter changes.

#### 5.4.6.3 *Shape Filtering*

The colour filter is relatively straightforward to implement: create a list of colours to filter then hide shapes with any of those colours. The shape filter requires computing a form descriptor: a vector of numbers describing a shape. To create effective shape filters, the form descriptor must not be too specific while also not too general that all shapes would match. We use a normalized Freeman chain-code with 8 connectivity [112]; this descriptor is invariant in translation, scale, and rotation, and is robust against slight variations of aspect ratios.

#### 5.4.6.4 *Shape Alignment and Attributes*

While ChartDetective makes no assumptions on the style of the charts, it relies on attributes which are fundamental to the way information is visualized. The pseudo-code is provided in Appendix A. All shape selections are first sub-divided to recover a consistent specification (C2) before passed to the extractors.

1. Alignment: The centroid of shapes such as axis ticks, line series, scatter plot markers is used to recover their position. For example, it is assumed that lines go over the centre of the data points. In a vertical bar chart, the top of a bar is used to get the associated value.
2. Grid line: If a grid line is found close to the tick, its position is used instead, because we found it to be slightly more accurate.
3. Box plot: It is assumed that box plots use the original and widely used representation first introduced by Tukey [386]: The inner quartiles are represented as a rectangle including any stroke outline when calculating

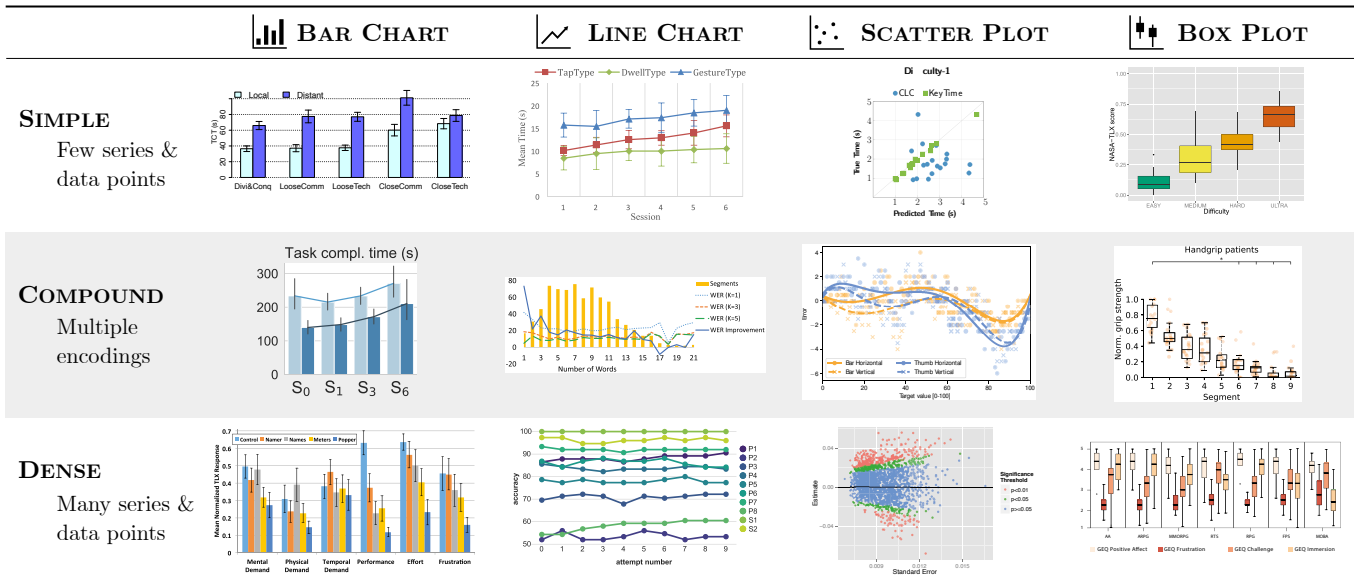


Figure 5.6: Charts that participants had to extract during the usability study. Charts were extracted from the following CHI proceedings (left-to-right, top-to-bottom): [231] [418] [221] [113] [41] [389] [59] [19] [107] [183] [252] [178]

values. The median is a line inside this rectangle, and any stroke outline is ignored when calculating the value.

### 5.5 USABILITY STUDY

We conducted a user study to see if the current implementation of ChartDetective fulfills our design goals in terms of supporting diversity (D1) and minimizing user interaction (D3). This study focuses on *usability*, answering the question: can participants use ChartDetective? A follow-up study measures the quality of the extracted data when compared to other tools (Section 5.6).

#### 5.5.1 Participants

We recruited 13 participants (22 to 34 age range, mean = 27.8, 7 identified as male and 6 identified as female)<sup>9</sup>. We screened participants for basic knowledge of charts: all participants were familiar with bar charts, line charts, box plots, scatter plots and error bars (self-assessed on a 5-point scale). Remuneration was \$15 CAD.

### 5.5.2 Dataset of Charts to Extract

We extracted 12 charts from the proceedings of CHI from 2015 to 2021. We consider the four most popular chart types at CHI: line charts, bar charts, scatter plots, and box plots. Further, we collected three charts per type, according to complexity:

- **Simple:** Few series and few data points that all use the same encoding.
- **Compound:** Two or more encodings are combined to represent data points. For example, a bar chart with lines, or a box plot with scatter points.
- **Dense:** Large number of series and data points, but all data points use the same encoding.

COMPOUND and DENSE charts have been notoriously difficult to extract using existing systems [79, 182] and thus pose a real challenge. Note that using our categories above, previous work has exclusively been tested on SIMPLE charts. We chose charts randomly amongst those fitting these criteria to maximize diversity while remaining ecologically valid (see selected charts in Figure 5.6). All charts were used by participants unaltered from the original paper; as a result, some are missing titles or legends, have overlapping or hidden elements, and some have grouped or separated shapes.

### 5.5.3 Procedure and Design

Participants took part remotely. After watching a two-minute video tutorial demonstrating the use of ChartDetective, participants were asked to extract the underlying data of 12 charts as accurately as possible and to think-aloud while doing so. The experimenter only intervened during the first four SIMPLE charts to answer questions and guide participants if necessary; participants worked independently for the remaining eight COMPOUND and DENSE charts. Participants advanced to the next chart by pressing a “Done” button or after five minutes, whichever came first. After each SIMPLE chart, the experimenter asked participants to identify 1) what was difficult; 2) what was easy; 3) what was tedious; 4) what was fast; and 5) what was slow. For each answer, participants also rated its importance on a 5-point scale.

We recorded the participant’s screen and microphone, as well a log of interactions with ChartDetective and the final extracted data. After the session, participants completed a questionnaire including a System Usability Scale (SUS) [179]. Finally, the experimenter conducted a semi-structured interview.

The order of the charts varied across participants: The four SIMPLE charts were always first, followed by the eight remaining charts. The the charts order was counter-balanced within these two groups.

Overall, each participant extracted data from  $4 (\text{CHART TYPE}) \times 3 (\text{CHART COMPLEXITY}) = 12$ .

<sup>9</sup> Our study was reviewed and approved by our institutional research ethics board. Consent was collected from all participants.

Complexity	Bar Chart	Line Chart	Scatter Plot	Box Plot	Total			
	✓ x - +	✓ x - +	✓ x - +	✓ x - +	✓	x	-	+
SIMPLE	100 0 0 1.3	100 0 0 0	100 0 0 1	100 0 0.4 0	100	0	0.1	0.6
COMPOUND	100 0 0 1.5	100 0 0 0.1	89.6 1.5 12.3 9.2	99.8 0.1 0.2 0	97.4	0.4	3.1	2.7
DENSE	100 0 0 0	100 0 0.7 0	100 0 0 0	98.6 0.4 1.8 3.9	99.7	0.1	0.6	1
Total	100 0 0 0.9	100 0 0.2 0	96.5 0.5 4.1 3.4	99.5 0.2 0.8 1.3	99	0.2	1.3	1.4

Table 5.1: Breakdown of the success rate when comparing the series extracted by participants to the series of the ground-truth data. All values are percentages. CORRECT (✓), INCORRECT (x), MISSING (-), and UNWANTED (+).

#### 5.5.4 Results

##### 5.5.4.1 Success Rate

To test the success rate in terms of usability, we compare participants' data to data extracted by one author before the experiment. The reasoning is twofold: first, we want to isolate the usability aspect and are not concerned by the fidelity of the data extracted by our tool at this stage, only by how well can participants use ChartDetective; second, the success rate can be directly interpreted as a measure of how close participants were to using the tool like an expert user, represented by the author who extracted the data. As such a series from the participant data is matched with a series from the author data (using a best-fit approach). We then classify each data point (i.e., cell in the data table) of each series in one of the following four categories.

- **Correct (✓)**, for a data point that is expected (i.e., present in ground-truth data) and that is strictly equal to the ground-truth value.
- **Incorrect (x)**, for a data point that is expected but is not equal to the ground-truth value.
- **Missing (-)**, for a data point that is expected but was not extracted (i.e., present in ground-truth but not in the participants').
- **Unwanted (+)**, for a data point that was not expected (i.e., present in participant data but not in ground-truth).

We measure success rate by calculating the rates of these four categories. For the CORRECT and INCORRECT rate, we divide the count by the minimum between the number of data points in ground-truth and the number of data points in the participant data. For the MISSING rate, we divide the count by the number of points in ground-truth. For the UNWANTED rate, we divide the count by the number of points in the participant data.

*Overall Success Rate* – Overall, participants extracted charts with high success: 99% (SD=5.9) of the extracted data were CORRECT, with only 0.2% (SD=1.5) INCORRECT data points (D1). Table 5.1 presents the breakdown of the results.

In fact, participants achieved perfect success rate in terms of CORRECT data for all bar charts and line charts, and above 98% for all other charts. The only exception being the COMPOUND scatter plot with only 89.6% (SD=17.7) of CORRECT data. Below, we further investigate the cause of some of these results.

*Confusion for COMPOUND Scatter Plot* – We found that the lower scores for the COMPOUND scatter plot were due to participants misunderstanding the chart. In fact, the data points extracted by participants were CORRECT, but not separated in series as it should have been. Both P1 and P5 interpreted different series as one single series (e.g., grouping all yellow dots as one single series, instead of distinguishing between crosses and circles). Because we calculate success rate by matching one series to another, if a series is missing, its data points count as INCORRECT. Similarly, the extra data points merged within the same series count as UNWANTED data. If, instead, we look for data points independent of series, the percentage of CORRECT data for the COMPOUND scatter plot reaches 99%.

*Filtering Causing UNWANTED Data* – Participants sometimes selected the legend as part of a series. For example, with the SIMPLE scatter plot, all participants made the selection of series easier using colour-filtering. However, doing so isolates the data points making the legend appear as part of the series. As a result, three participants selected circles from the legend, creating UNWANTED data.

*Selection Difficulties with Box Plots* – Box plots required a precise selection of the whole element at once. However, when the boxes were close to other elements (e.g., the axis in the COMPOUND box plot), some participants inadvertently included other elements as part of the box, creating an INCORRECT data point.

#### 5.5.4.2 Time

On average, participants extracted charts in 3min 6s (SD=1min 18s) (D3). Only 2 participants were not able to complete extractions within 5 minutes: P4 could not extract all error bars for the DENSE bar chart and P10 did not have time to select the last two blue lines in the COMPOUND scatter plot. The most dense charts were not necessarily the slowest to extract. For instance, extracting the DENSE scatter plot and its 2,000 data points took only 1min 58s (SD=23s). But overall, DENSE charts were the slowest (M=3min 17s, SD=1min 22s), followed by SIMPLE charts (M=3min 12s, SD=1min 10s) and COMPOUND charts (M=2min 50s, SD=1min 19s). Still, all average times were well under 4 minutes, confirming that ChartDetective minimizes user interaction enough to allow the extraction of charts in reasonable time (D3).

#### 5.5.4.3 Error Bars and Series' Names

Participants correctly extracted 98.4% (SD=11) of the error bars and 82.8% (SD=34.1) of series names. These results were calculated on a subset of charts



considering that not all charts had error bars or legends. It is unclear why the series names score is lower; some participants did not extract the series names for no apparent reasons, even though they were aware of the feature as they all did it for SIMPLE charts.

#### 5.5.4.4 Usability

On average, the System Usability Score was 90 (Mdn=90, SD=4.2). For reference, a System Usability score above 85 is considered excellent [18].

On a 5-point scale, participants rated all features of ChartDetective as useful (4 or above): participants “strongly agree” on the usefulness of the colour filter (Mdn=5, SD=0, [■■■■]), the shape filter (Mdn=5, SD=0.7, [■■■■]), the selection system (Mdn=5, SD=0.3, [■■■■]) and the reconstructed chart (Mdn=5, SD=0.7, [■■■■]). Additionally, participants “agree” that the overlay was useful (Mdn=4, SD=1.5, [■■■■]).

Regarding participants self-assessed performance with ChartDetective, they all agreed that they could extract and reconstruct charts accurately (Mdn=5, SD=0.5, [■■■■]) and that they were in control of what they wanted to extract (Mdn=5, SD=0.6, [■■■■]). Finally, they all agreed that they would like to use the system again (Mdn=5, SD=0.4, [■■■■]).

#### 5.5.4.5 Strategies

While the tasks were identical across participants, they sometimes adopted different strategies to extract the data.

*Filter to Isolate, to Declutter, or to Guide?* – All 13 participants used the filters but we observed three distinct strategies: filtering to *isolate* only the element to select (i.e., only one active filter); filtering to *declutter* the image by removing the few elements that were preventing a selection; filtering to *guide* the selection by going through each colour one-by-one (sometimes multiple times) to be sure not to miss any series. Participants using the *declutter* strategy had the advantage of preserving visual context. For example, it made it easier to distinguish marks that are part of the legend and would have looked like data had the *isolate* strategy been used.

*Step Order* – The order in which to perform the extraction was often a trade-off between speed and cognitive load. Some participants extracted all series (e.g., bars, lines) before moving on to error bars or legends, making the selection process easier by repeating the same task and reusing the same filters. Others preferred to extract the error bars and the legend right after extracting the corresponding series, making it easier to match a series with its meta-data. Participants would often decide on a strategy based on chart complexity.

*Selection Strategy* – Participants either selected elements precisely, often one-by-one, using the zoom-in function if necessary, or they made a first rough selection then refining it by using SHIFT to add to or remove elements. Some participants also relied on the ghost shape mechanism to speed up

the selection of the last series: once a series is extracted, its shapes become unselectable, meaning that when only one series is left, the shapes that are selectable will necessarily belong to the last series.

#### 5.5.4.6 Comments

Overall, participants were positive about ChartDetective and its functionality. Below, we group participant comments from the interview and during the study around a set of themes that were frequently mentioned. Due to the use of semi-structured interviews, some of these themes were only mentioned by a subset of participants.

*Learnability* – A few participants commented on learnability. All agreed that the tool was quick to learn.

*P6 – “I liked that the cognitive load was pretty low, like, it was super fast to learn... I got markedly better after like 2, 3 tries. I really liked that it had a lot of the traditional settings and feel to it.”*

*P12 – “I’ve been using it for less than an hour and I already feel at ease”*

Specifically, P7, P11, and P13 commented about the drag and drop interface saying it made the interaction easy due to the visual feedback of what is being dragged and where it can be dropped.

*P7 – “That is actually quite useful that it shows you the... sort of see-through thing you’re dragging.”*

*P11 – “Drag’n’drop is truly useful. It’s super clean, like you can easily select and then you immediately see where you can drop.”*

*P13 – “I liked how it was organized, how you could... like it was easy to have that one navigation bar on the side and pull everything over and see it appear on the [Reconstructed] chart below.”*




*Most appreciated features* – An overwhelming majority of participants commented about the colour and shape filters, most cited them as one of their favourite features.

*P8 – “Filters are super useful. Really facilitate the task. Some [charts] would even be impossible [to extract] without.”*

*P12 – “The filters, I really thought it was a killer-feature. Your chart is super crowded, you ask yourself: «Wow, how am I gonna do that, it’s too difficult», I do two clicks, then it becomes super easy.”*



Other features were less often mentioned by participants as something they liked: the reconstructed chart (P2, P11 P13), the ghost shapes (P3, P7, P8), the overlay (P8), how the legend is sometimes automatically matched with series (P2), and the coloured table (P12).

*What was tedious, difficult or slow?* – After each SIMPLE chart, participants were asked what they found tedious, difficult or slow and how much using a 5-point scale. The most frequently mentioned difficulty was the selection of

elements (P5, P6, P7, P10, P12, P13), although they still rated it as relatively easy on average (from very hard to very easy: Mdn=4, SD=.83, ). Specifically, the selection of error bars were most often mentioned as moderately tedious (from extremely to not at all: Mdn=4, SD=.71, ) and slow (from very fast to very slow: Mdn=3, SD=.75, .

*P1 – “Selecting error bars. That was difficult, just because they were overlapping.”*

*P8 – “Selecting error bars [was slow] because you have to select them a bit like... one-by-one to [...] distinguish them between series.”*

*What was easy or fast?* – Similarly, we asked participants what they found easy and fast. The drag-and-drop interface was most commonly cited as being easy (from very hard to very easy: Mdn=5, SD=0, ). The selection after applying filters was most often mentioned as fast (from very fast to very slow: Mdn=1, SD=.37, .

*P3 – “The drag and drop... that tool is easy to understand and use. And it is easy to isolate the data you want to collect [using the filters].”*

*P2 – “Selecting the points, thanks to the filtering, it was really fast”.*

*Accuracy perception* – While participants were highly accurate overall, P7 mentioned that the artificiality of the task might have had an impact on the quality of the data extracted.

*P7 – “I’m not sure how accurately I actually covered... copied the charts, I have to say. Because the image [Reconstructed chart] was relatively small, and I did not spend a lot of time looking at data points if they were correct or not. It was more like a «meh» roughly looks the same, fine, cool. [...] Obviously because it is not data that I am invested in so I don’t care if it is accurate or not.”*

## 5.6 DATA QUALITY STUDY

The goal of this study is to measure the quality of the data obtained from vector charts and ChartDetective relative to what could be obtained using rasterized images and existing tools (D1). Quality is defined as how similar the extracted data is compared to the original data that was used to create the chart.

### 5.6.1 Dataset

We create a new dataset of charts for which we know the *exact* underlying data. To cover a wide diversity of chart styles and chart generators, we mix generated charts with charts from CHI papers:

- **Generated Charts:** We generated four different charts (1 bar chart, 1 line chart, 1 scatterplot and 1 boxplot) with four different chart generators: Microsoft Excel for Mac version 16.61, Python matplotlib version 3.5.1, Javascript plotly version 4.10.0 and R ggplot2 version 3.3.6. All 16 generated charts use a dataset on life expectancy and GDP per country obtained from GapMinder<sup>10</sup>. Each chart visualization presented different information: line charts show the evolution of the life expectancy over the years for four regions of the world and with error bars; scatter plots show the life expectancy depending on a country’s GDP; and box plots show the life expectancy per region and for male and female). Generators used the default style parameters.
- **Extracted Charts:** Using our dataset of papers with charts published at CHI between 2015-2020, we used a script to find those with vector charts and with data available on Open Science Framework<sup>11</sup> (OSF). Only 23 papers fulfilled this criteria (74 papers had an OSF link, but 44 of those did not contain qualifying vector charts, and 7 had no data in their OSF repository). Using reasonable effort, we cleaned and recreated the data used by charts in 14 different papers and extracted between 1 and 3 different charts per paper. The final study dataset counted 26 charts (13 bar charts, 6 scatter plots, 5 line charts, and 2 box plots).

### 5.6.2 Baseline

Like ChartSense [182], we use WebPlotDigitizer as our baseline. Other tools either do not provide their source code<sup>12</sup> or a working implementation [182], do not provide a full pipeline to obtain the data from charts [54, 55, 302], or are limited in the styles and types of charts that they support [55]. Moreover, our comparison here focuses on the best achievable results using vector graphics compared to raster images. In that regard, the result obtained with raster images should be comparable across tools. Thus, in the rest of this section we use “rasterized charts” to refer to charts extracted using WebPlotDigitizer.

### 5.6.3 Procedure

One author with hours of experience with both ChartDetective and WebPlotDigitizer extracted all charts from our dataset as accurately as possible using both tools. The author had no time limitation and ensured the data was as accurate as possible. To use WebPlotDigitizer, extracted charts were rasterized at 300 dot-per-inch (DPI) which is considered high-resolution and recommended by IEEE [174]. Generated charts were obtained from chart generators and directly outputted as PNGs for WebPlotDigitizer (300DPI), and PDFs for ChartDetective.

<sup>10</sup> <https://www.gapminder.org/data/>

<sup>11</sup> <https://osf.io/>

<sup>12</sup> On request, authors of ChartSense could not provide their source code due to proprietary reasons.

## 5.6.4 Results

Dataset	n	Vector Charts	Raster Charts
Generated			
excel	4	0.24% (0.19)	0.37% (0.21)
matplotlib	4	0% (0)	0.16% (0.08)
ggplot2	4	0% (0)	0.13% (0.12)
plotly	4	0% (0)	0.13% (0.09)
Extracted	26	0.13% (0.27)	0.68% (0.69)
Total	42	0.11% (0.23)	0.50% (0.60)

Table 5.2: Average relative error of the values obtained from vector or raster charts. Standard deviation shown between parenthesis.

Like ChartSense [182], relative error measures how close the extracted data is to ground-truth:

$$Relative\ Error = \left| \frac{v_{groundtruth} - v_{extracted}}{v_{groundtruth}} \right|$$

We assume the chart represents the data precisely and accurately, so relative error is solely attributed to the extraction tool. Note that the expert always made sure no data points were missing. Thus, the ground truth data and extracted data had the same number of points and in the same order. We calculated the relative error of each pair of data points from the ground truth and extracted data, and then aggregated them using the mean of all the relative errors. This measure corresponds to how close the extracted data is to the ground truth.

Overall, data extracted from vector charts using ChartDetective had a significantly lower relative error than data extracted from rasterized charts using WebPlotDigitizer (0.11% vs 0.5%, Student’s t-test  $p < .05$ ). This is more than 4 times lower, corresponding to a factor of 78%. Table 5.2 breaks down the relative error for tool and dataset.

*Effect of Generator* – The data obtained with ChartDetective from generated vector charts was identical to the ground-truth with the exception of Excel charts (relative error matplotlib: 0% SD=0, ggplot2: 0% SD=0, plotly: 0% SD=0, excel: 0.24% SD=0.19). After further investigation, it appears that charts generated with Excel are using some approximations and do not perfectly represent data. While the problem exists with all types of charts such as line, box plots and scatter, the issue becomes obvious by generating bar charts with exact real values. Examining the SVG description of bar charts generated by Excel, positions of bar tops is inconsistent even when data is a series of real and regularly spaced values (e.g., 50, 60, 70). This is demonstrated by zooming into a bar chart to see how bars do not line up consistently with corresponding grid lines (Figure 5.7). We verified this behaviour with macOS Excel (version 16.61) and Windows Excel (version 2205).

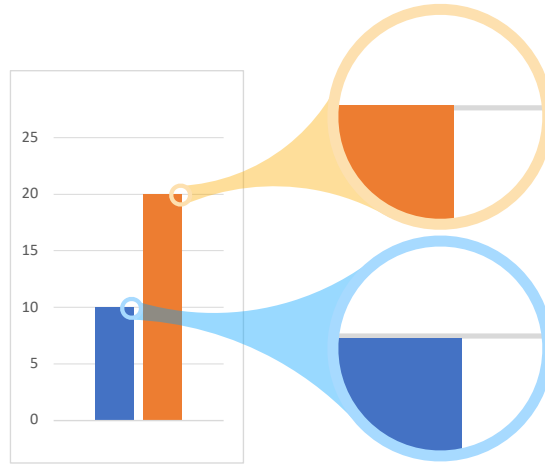


Figure 5.7: Minimal example of a chart generated by Microsoft Excel and for which the data is imprecisely depicted. Despite the bars representing exact values (10 and 20), they do not consistently line up with corresponding grid lines: the grid line is either below (blue bar) or above (orange bar).

## 5.7 DISCUSSION

Despite the importance of extracting high-fidelity data, approaches to chart reverse-engineering predominantly focus on rasterized formats, limiting the accuracy of the data obtained. We provide theoretical and empirical evidence showing that extracting charts using their vector representation has advantages that can lead to an improved quality of the extracted data. We also detail the design and implementation of the ChartDetective system demonstrating how vector information can be used to provide new features, and how it can be processed to obtain underlying data. Through a usability study, participants found the system highly usable and were capable of extracting even the most challenging charts. A second study demonstrated that extracting a chart using its vector representation lead to higher accuracy of data than when extracting the same chart in raster format and using existing tools.

### *Comparison to ChartSense*

Jung et al. also use the relative error to compare their ChartSense system to WebPlotDigitizer [182]: their system achieved 0.7% whereas WebPlotDigitizer achieved 0.81%. For comparison, we found a relative error of 0.11% with ChartDetective and of 0.5% with WebPlotDigitizer. Differences in our methodology likely explain different results for WebPlotDigitizer: Jung et al. obtained their result from a user study with 16 participants whereas our results were obtained by an expert user. Furthermore, our dataset was different: Jung et al. used line and bar charts found on Google Images with at most two series and nine marks per series. In contrast, we used a dataset of generated charts and charts published at CHI, including charts with hundreds of marks.

Regardless, both studies suggest that relative error below 0.5% may be out of reach for raster chart data extraction using existing approaches and that ChartDetective fulfills its goal of maximizing accuracy (D1).

#### *Control of Anchor Points*

ChartDetective differs from other manual and semi-automatic tools in that users select whole shapes and let the system decide how to handle them to extract the data (D3). In contrast, other systems often rely on users directly specifying *anchor points* to define the exact point depicted by a marker, even if this marker takes various forms and sizes. One advantage of our system generated anchor points is reduced standard deviation (D1): the same shape selection always results in the same value. In contrast, giving users control over anchor points inevitably results in lower precision due to selection errors or simply because users have to “guess” the centroid of shapes. This can vary greatly depending on the style (e.g., thick lines or large markers) and the forms of the shapes [141, 329]. We advocate for a shape-selection approach because users can reliably select shapes (see Study 1) and that our assumptions regarding anchor points were valid across a diverse set of charts (see Study 2).

#### *Open Science at CHI*

Corroborating the findings of previous work [1, 398], and further motivating the need for chart extraction methods, we experienced first-hand the difficulty of obtaining data related to CHI papers. Of the 3,673 papers published at CHI from 2015 to 2020, only 74 papers contained an OSF link (2%). Our automatic mining approach likely missed data published using other methods like custom webpages. But more importantly, even within these papers, we could not always reproduce the charts. This was sometimes due to missing data (the OSF link contained other material) or because only raw data was provided without guidance to reproduce processed data used in charts. For example, the cleaning procedure, aggregation method, and formulas applied were missing. Additionally, there was often a mismatch between the data names in the chart and labels in the raw data.

#### 5.7.1 *Limitations and Future Work*

##### 5.7.1.1 *Support For Raster Charts*

A large portion of charts remain embedded as raster graphics and cannot leverage the benefits provided by ChartDetective. A tempting alternative to using tools such as WebPlotDigitizer [324] could be to vectorize raster charts so that they can be used with ChartDetective. New state-of-the-art vectorization algorithms [228] might provide the best approximation for the location of shapes representing chart elements and possibly help disentangle overlapping shapes. However, many benefits provided by “original” vector

charts would be lost and the quality of the input raster image will limit the vectorization process. While a vectorization approach can extend our system to rasterized charts, it seems unlikely to provide substantial benefits over using raster-based extraction tools.

#### 5.7.1.2 *Optical Character Recognition of Rendered Text*

When the user study was conducted, participants had an earlier version of ChartDetective where rendered text could not be automatically retrieved, and required participants to enter it manually. While we found rendered text in vector charts to be relatively rare in practice, we added OCR support in ChartDetective (C4). Preliminary tests suggest excellent performance: we rendered all text in the charts used in the usability study, and the OCR engine was able to recover 97.5% of all characters correctly. A more extensive evaluation is needed to make definitive conclusions.

#### 5.7.1.3 *Diversity of Chart Styles*

The many ways in which charts represent and encode data is one of the main difficulties faced by reverse-engineering approaches [78–80, 182]. We choose to evaluate our tool on real charts published at CHI that exhibited challenging properties like high density, overlapping shapes and mixing encoding (C1, D2). We encourage other work to do so as well, considering such charts are abundant in practice. Of course, our dataset is not universally representative. First, we only examine charts in the HCI research community, but others communities might have different practices regarding charts. Second, the HCI community is arguably more aware of good data visualization practices. This is both a strength of our dataset because HCI charts may be more creative in their use of marks and visual channels, but also a weakness because charts may be clearer and exhibit fewer flaws [43].

Further, we cannot guarantee that our tool is general enough to handle all charts. ChartDetective relies on fundamental attributes of charts and on the structure of the vector representation. We verified these were reasonable and applied to major chart generators, but charts could use different encoding structures. Moreover, our system focused on the four most common data visualizations (bar, line, scatter, and box plots), but more work is needed to implement extractors for other types such as stacked bar, violin, and pie charts.

#### 5.7.1.4 *Automatic Selections Through Suggestions*

Considering the limitations of previous work, a goal of ChartDetective was to preserve some user-control to allow the selection of specific data, the support of complex and diverse charts (D2), and the verification of the results (D4). While the time it took participants to extract charts was under 4 minutes, this could be further shortened by automating some tasks (D3). For example, participants often mention the selection of error bars as the most tedious. We



believe this could be improved through suggestions generated by continuously learning from user actions: after selecting the error bars for one series, the system could learn to recognize the characteristic shapes composing error bars and suggest repeating the action for other series. General selection suggestions could also be learned from the community and be offered prior to the user first selection, only based on the shapes identified in the chart. We believe this active learning approach with suggestions is the best compromise between incorporating automation while preserving high controllability [330]

### 5.7.2 *Applications of ChartDetective*

ChartDetective can power several downstream tasks that require access to accurate data when only the charts are readily available. Specifically, readers interested in re-analyzing the results presented in a chart can use ChartDetective to extract underlying data and then use it as input to their analysis, or to compare their results against. Other applications include the use of ChartDetective as an intermediate step to re-design existing charts: a chart found online or in a document might benefit from being redesigned if it is poorly structured or deceptive [43, 246, 292], uses a representation ineffective to support users' task [267, 314], is overblown and shows too many data points [385], or is not accessible because of its colour palette and style [194, 412]. For all these scenarios, the chart can be loaded in ChartDetective to let users select only the data of interest. Users can then export the underlying data to be visualized in an authoring tool, or, they can use the automatically reconstructed interactive chart and tweak its specification such as changing its aspect ratio to avoid deceptive charts that exaggerate or undermine slopes [292], its scale to remove truncated axes [43], its colour to make it print- and colorblind-safe [194], and its encodings to make it align with the user's task [267, 314].

### 5.7.3 *Takeaway for Chart Authors*

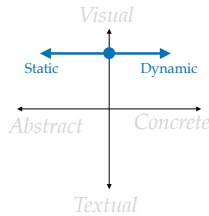
Through this work, we hope to encourage authors to share their figures as vector graphics. Besides facilitating data extraction, vector graphics have numerous advantages: high quality at any resolution; more accessible; easily modifiable; and typically smaller in size. All major chart generators have an option to export charts as vector graphics which can then be directly imported into documents such as L<sup>A</sup>T<sub>E</sub>X, MS Word documents, or web pages. We also recommend authors carefully choose chart generators because they can differ in how well they represent data. For example, we found that MS Excel generated less accurate charts than either matplotlib, ggplot2, or plotly. Although these differences are invisible to the naked-eye (Figure 5.7) they are a concern in the context of chart extraction.

## 5.8 CONCLUSION

We presented ChartDetective, a tool within the pipeline to extract data from charts using their vector representation. Through theoretical and experimental evidence, we showed the benefits of using vector graphics to extract data compared to using raster images. We identified the challenges associated with building such a system, demonstrated opportunities for novel features, and evaluated its usability and quality of the extracted data. Recovering complete and accurate data is the first step to tackle downstream tasks such as redesigning existing charts or making them dynamic, interactive, and accessible. Besides helping users recover this data, we hope our system serves as a building block to leverage the wealth of information currently locked inside static visualizations.

# 6

## BRINGING INTERACTIVITY TO STATIC DIGITAL DOCUMENTS



In our design space fig. 1.1, Chameleon allows the overlaying of more abstract and concrete interactive visuals.

The previous chapters proposed ways to augment documents with interactive representations. However, their implementation required the use of a custom PDF viewer. This approach is limited to a document format (PDF) and prevents readers from using their preferred software. In this chapter, we investigate how to support novel interactive representations such as the ones previously proposed, but for all document formats and document viewers.

### 6.1 MOTIVATION

The meaning of the word *document* denotes a textual record [39], originally considered to be a paper-based artifact. However, the word *document* is now often used to refer to digital files alongside paper-based artifacts, and, while both paper-based and digital documents are still textual records, by nature a digital file can encapsulate multimedia information and interactive features alongside text and static images.

Despite the opportunity for dynamic and interactive content, many documents available on the internet have been produced using various file formats (e. g. docx, pdf, ppt). The applications used to generate these document types provide only limited support for incorporating interactivity into disseminated versions of the documents they produce.

There are, however, many potential benefits to interactivity. In his UIST 2014 closing keynote abstract, Victor [395] notes that

*'Human beings naturally have many powerful modes of thinking and understanding. Most are incompatible with static media. In a culture that has contorted itself around the limitations of marks on paper, these modes are undeveloped, unrecognized, or scorned.'*

Even a task as simple as describing an interaction technique or interactive scenario requires careful wording when presented in static form, often requiring a *comic strip* figure describing the interaction in several steps, rather than letting the reader *experience* it (see Figure 6.1). Similarly, reporting scientific results often requires an author to make difficult decisions toward which data, and in which form, to report in a paper or while giving a research talk, rather than interacting with the data or allowing a reader to interact with alternative visualizations or analyses of the data. Even writing a simple how-to document to perform an everyday task can be complicated if all information is static. Some companies have attempted to make limited interactivity available in their formats. For example, PDF files can embed multimedia and interactive content [121, 139], but integrating interactivity into a PDF file is far from

This is a good illustration of the media richness theory where a rich animated figure would convey the information better than a static one.

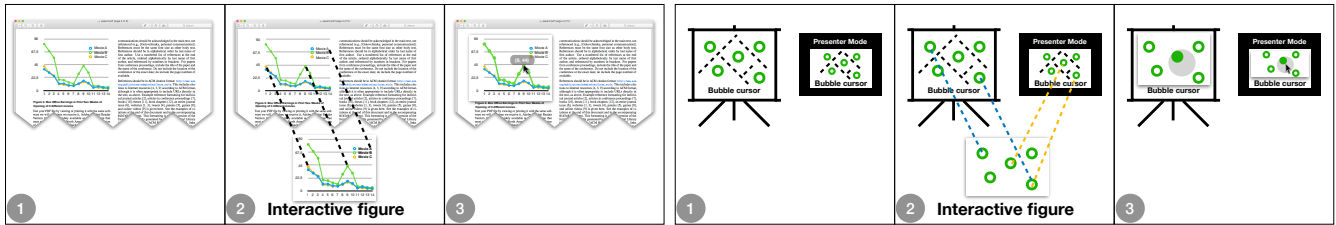


Figure 6.1: (1) Static documents such as PDF (left) and presentations (right) can be augmented with Chameleon by (2) locating corresponding interactive figures in the Chameleon database and (3) overlaying them on top of the existing documents.

easy [139] and leverages Adobe Flash which is being discontinued [5]. Similarly, PowerPoint users can embed videos in their presentations or design staged animation that can be used as a mean to illustrate a dynamic situation. However, features are usually limited to specific file formats and the result can vary depending on the software version used to display the presentation. Moreover, limited support is provided for interactive figures.

Since interactive figures cannot be embedded directly into the document, authors tend to build dedicated external web pages linked in the document [277]. However, the extra work required from readers makes them less likely to engage with the supplementary material. Moreover, having multiple views significantly impacts cognitive overhead mainly due to the effort required for *comparison* and *context switching* as well as display space requirements [400]. Similarly, interrupting the *flow* of a presentation in order to manipulate an interactive figure using another application harms the *immersive*, *engaging* and *effectiveness* aspect of the story being presented [83].

One solution to enable interactive content in documents would be to replace the associated document formats with a novel, more flexible, file format that would more easily enable interactivity. For example, Dragicevic et al. [91] created a web-based tool in order to demonstrate the exploration of multiverse explanations of statistical analyses in research papers. However, existing file formats remain firmly anchored in users' habits and replacing them would require significant changes, both in software and infrastructure. Furthermore, ideally interactivity would not be limited to new documents. Archived documents could, in many cases, also benefit from interactivity, but the process of fully transcoding and then adding interactive elements would be prohibitive in most cases.

What would be more ideal would be to preserve the existing formats and workflow, and, instead, to provide tools to augment digital versions of the files such that both new and legacy documents can become *reactive documents* [395]. Expanding on this point, support for the creation, dissemination, and viewing of reactive documents [395] should allow rich and flexible access to interactive features, preserve existing document formatting for visual correspondence with paper versions, leverage existing file formats, and allow users to continue to use their current document viewing tools rather than forcing them to adopt different software applications.

In this chapter we introduce Chameleon (Figure 6.1), a tool that combines computer vision feature matching algorithms with an open database format to allow the incorporation and overlaying of dynamic content over any type of document (e. g. PDF files, Microsoft Word and PowerPoint documents, eBooks, etc.) without modifying existing applications or the source document. Using Chameleon, these documents can be easily upgraded with user-generated HTML5 interactive content. Chameleon thus allows the simplified provision and viewing of an enhanced version of a research paper with embedded interactive demonstrations and videos. It can also be used to perform live demonstrations of interaction techniques while giving a presentation without having to switch tools.

Chameleon accomplishes this via two interfaces: a background service and an editing tool. The background service runs on the user's system and analyzes raw, pixel-based on-screen content to identify static content for which interactive replacement content exists in its database. It then replaces this static content with interactive content on the visual display, seamlessly allowing a user to interact with the interactive version of the content. Scale and placement is preserved within the existing document, allowing easy visual correspondence between static-based versions of the document and on-line interactive versions of the same document. Alongside a display system service, Chameleon incorporates an editing tool to allow users to dynamically select static content within a specific file, associate interactive content with that static content, and verify the correct rendering of their interactive content. Since Chameleon relies on computer vision to analyze display content, identify static content with interactive enhancements, and seamlessly overlay that content completely automatically and transparently to the user, it guarantees its compatibility with any type of document (past, present or future) and any application a user currently uses to view that document, because it relies on what any document viewer ultimately does: displaying the document as pixels on-screen.

Our work makes the following contributions: 1) It introduces the notion of dynamic layering for augmenting static digital documents. 2) It presents the design and implementation of Chameleon, a document augmentation tool that implements this notion of dynamic layering on the macOS and Linux operating systems. 3) It reports on the results of a study comparing feature matching algorithms on non-natural images and for different scale levels. 4) It reports on the results of an experiment evaluating the performance of Chameleon for augmenting a corpus of documents, in this case a set of research papers and slideshows. 5) It details the results of a first-use study evaluating our design choices and the usefulness of Chameleon.

## 6.2 BACKGROUND AND RELATED WORK

We first detail previous attempts to make static documents more dynamic before reviewing vision-based techniques previously used in the contexts of GUIs and documents.

### 6.2.1 Making Static Documents Dynamic

In his work on *Explorable Explanations* [393], Victor highlights three ways that reading environments can encourage greater inquiry via interactivity: reactive documents, explorable examples, and contextual information. Each of these allow readers to develop enhanced understanding either by manipulating data to see how model outputs change, by interacting with content widgets to experience behaviours, or by interacting with content to obtain additional contextual information on concepts in the document.

While Victor's analysis of current state-of-the-art indicates that much work remains to be done in interactive documents, it is also true that many researchers have tried to create interactive experiences around documents. In terms of paper-based interaction with documents, the PADD system architecture [144] and the related PapierCraft system [230] which leveraged Anoto's digital paper support the remixing of printed information in various ways. If, instead, the documents are scanned into a computer, the ScanScribe system looks specifically at rough documents, essentially hand drawn sketches, and ways to structure and organize this material for post-hoc digital manipulations [339]. Finally, if information is captured in both paper and digital formats, systems such as ButterflyNet seek to combine myriad data sources including hand-drawn text, images, and other digital information into a cohesive single document [415]. However, the output of these systems remains little more than static content that has been remixed in various ways to produce a new, albeit static, document.

In the digital domain, recent research explored how to incorporate dynamic media into documents. For example, the Webstrates system [200] leverages a custom web server that provides shareable dynamic media to support flexible document creation, essentially a form of *transclusion* [272]. Alongside Webstrates, Wincuts [376] also allows users to carve out regions within windows, reposition those subregions, and then continue to interact with information in the subregion. However, the primary goal of Wincuts was to optimize screen real estate during, for example, document creation, rather than to replace static content with interactive content for explorable explanations. In ways similar to Wincuts, tools like d.mix [151] or Clip, Connect, Clone [114] mix information from multiple web pages to create mashups: web pages juxtaposing elements from others [76]. In contrast, Chameleon transcludes HTML content using dynamic layering onto figures of static documents.

Other options in the digital domain that can support Victor's explorable explanations include HTML documents, which can include interactive scripts, or a new document format such as ePub, which can be redesigned to incorporate interactive artifacts [406]. Consider, first, HTML documents; one reason that PDF persists as a format for document distribution is that PDF supports the flexible dissemination of documents by combining both content and visual formatting in a single, compact data format. The goal of presenting information in written form is the design of an artifact – a formatted document – that supports the ergonomics of reading. HTML rarely achieves this simplicity of presentation: new devices and browsers frequently render

aspects of a document idiosyncratically, which can make reading an awkward experience, and even mobile versus desktop versions of documents often require bespoke aspects to ensure proper presentation in their respective formats. However, even if we advocate moving to a new format (either HTML or some novel new data format), in both cases this means that our extensive archive of existing documents must either be transcribed and re-imagined with interactive content, or must be maintained in original non-interactive format.

Even writing a simple how-to document to perform an everyday task can be complicated if all information is static. Some companies have attempted to make limited interactivity available in their formats. For example, PDF files can embed multimedia and interactive content [121, 139] through the inclusion of SWF files. However, the interactivity supported by PDF files still suffers from several limitations. First, integrating SWF into a PDF file is far from easy [139]. Second, SWF files are typically produced using Adobe Flash (even though they can also be produced in other ways using third-party software), which is being discontinued [5], so Adobe, itself, encourages designers to build interactive content using Web standards such as HTML 5 [4]. Third, this interactivity is barely supported by PDF viewers (basically, only Adobe Acrobat Reader will run the SWF file). Finally, the SWF file has to be embedded into the original document, which means that a static or legacy document cannot be converted into an interactive one without producing a new PDF file.

There has been significant efforts in circumventing PDF limitations by creating new kinds of documents viewers. For example, Utopia Documents [15] is a novel PDF viewer integrating visualization and data-analysis tools. By inspecting the content and structure of PDF files, Utopia Documents is able to integrate dynamic objects and annotations in the documents. In the same vein, Elastic Documents [17] generates visualizations from extracted tables in PDF documents and then display the relevant ones depending on the reader's focus. However, these viewers do not allow the transclusion of interactive content and are limited to the PDF file format.

We believe that the optimal approach to supporting the building blocks of explorable explanations is to incorporate reactive documents, explorable examples, and contextual information within the confines of existing static documents. Essentially, the goal of Chameleon is to support the overlaying of interactive features within the pre-existing presentation constraints of current documents. We do this by selectively replacing static content with dynamic content without manipulating the underlying document format and without any modification of existing document viewing applications.

### 6.2.2 *Using vision-based techniques with GUIs and documents*

Our concept is based on the identification of figures by analysing pixels on screen. This analysis needs to be robust against scale variations considering documents can also be displayed at different zoom levels and on different

screen resolutions. These requirements make techniques based on template matching not suited for the task [416].

There exists a body of work in overlaying content on images within documents. For example, Kong et al. introduced Graphical Overlays[201], in which information is extracted from chart bitmaps in order to overlay graphical aids, such as gridlines on top of the charts. However, the user first has to capture a bitmap version of a chart and then upload it to a website to get the augmentation. In the same vein, Lu et al. [237] augment existing web-based visualizations by providing a suite of interactions such as filtering, comparison and annotation to help users visualizing information differently. Chameleon differs from the above systems in that it enables the support of any kind of augmentation without being limited to specific figures or specific interactions. In addition, the augmentation is co-localised with the document the user is reading rather than being rendered in another window.

A series of systems exist that seek to analyze GUI elements dynamically in an interface [86, 419]. First, the domain of these systems – identifying GUI elements for augmentation – differs from the domain of explorable explanations, in which we wish to augment document elements. Further, the approach used by these systems cannot be generalized to the domain of document annotation. Specifically, for example, Dixon et al.’s Prefab system reverse engineers an interface by analysing pixels on screen [86] using invariant features. The use of invariant features is possible when analyzing GUI elements because these are auto-generated during execution. However, invariant features cannot be applied in the case of document images: when a document is resized, all pixels of the figure are affected and there are no invariant features *a priori*. Another interesting approach was proposed in Yeh et al’s Sikuli [416]. They applied classical computer vision techniques to identify an individual GUI element on the screen. The GUI element could then be used in search (e.g. to learn how to use the GUI widget), or dropped into scripts (to automate interface execution). To identify elements, Sikuli uses the SIFT [236] feature descriptor to extract features from elliptical patches detected by the MSER detector [250]. They then associate each of these features with features extracted from the screen. Finally, using a voting system and checking if a transformation can be computed from the associated features, they precisely locate the image on the screen. Using this method along with OCR and template matching, they report a success rate of 70.5% when performing search on a UI element, and they report less than 200 ms for identifying a single target GUI element on the screen. Given their focus on GUI elements, it is unclear whether a similar approach can work for arbitrary analysis of images in documents.

In the context of detecting plagiarism, methods have also been proposed to detect if two images are similar. Meuschke et al. [258] combined Perceptual Hashing with other algorithms to detect similar figures. However, since figures need to be extracted from the document first, these methods cannot be applied to localize figures. Similar to Sikuli [416], Iwanowski et al. [177] used Feature Matching, but their evaluation was focused on a small corpus of photographs and they acknowledge that more work is needed to confirm that it generalizes to other kind of figures (charts, drawings, screenshots).



Therefore, we evaluate SIFT [236] and MSER [250] alongside other image detectors and descriptors during the design of Chameleon. Furthermore, we explore how best to augment documents (versus GUI elements [86, 416, 419]) and present Chameleon as an end-to-end system to support both creating and interacting with explorable explanations.

### 6.3 CHAMELEON

The core idea of Chameleon is to augment pre-registered figures of digital documents by layering HTML5 resources over them (Figure 6.1). It uses feature matching algorithms to identify the pre-registered figures, and overlays on them an undecorated WebView that runs the corresponding HTML5 resource. Chameleon has been developed in C++ using the Qt 5.12 framework and OpenCV 4.0.1, together with platform dependent APIs. OpenCV provides feature matching algorithms while Qt provides the WebKit engine to render and interact with the HTML resources. Chameleon currently runs on macOS and Linux using respectively Cocoa and X11 APIs<sup>1</sup> and could also be implemented for Windows by replacing the platform dependent API calls.

Chameleon comprises two main parts: a REGISTRATION TOOL to register a figure from a document with its augmented version and an AUGMENTATION TOOL to analyse opened documents and *augment* them if *augment-able* figures are found.

#### 6.3.1 The registration tool

The registration tool provides the interface to select a region of a document that will be registered in the database. To accomplish this, the user first opens a document that contains the target figure and navigates to its location. Using the *Register* command in the Chameleon menu, she opens a screenshot of the current viewport in a new window in which she highlights the region she wants to augment (typically the portion where the target figure is displayed) by dragging the mouse. The interface also provides a text field to enter the URL of the augmented resource. After validation, the augmentation is stored in a database. In order to compress the size of the database and for privacy reasons, we do not store the actual image, but only the pre-computed regions and descriptors of this image as provided by feature matching algorithms (see following sections for details) along with the document file size, its MD5, the dimensions of the selected region, and the url of the augmented resource. One potential downside of this approach is that, if an augmented document is modified, its MD5 changes. To address this, the registration tool also tracks modifications to augmented documents using DTrace [405], and, when an augmented document is modified, the user is prompted to either ignore or register the new, edited version of the document.

<sup>1</sup> Source code available at [ns.inria.fr/loki/chameleon](https://ns.inria.fr/loki/chameleon)

### 6.3.2 *The augmentation tool*

The augmentation tool runs in background and loops on the following six steps:

1. The *analysing opened files* step monitors the opened documents and determines those that contain figures that should be augmented (i. e. with entries in the database). This step allows Chameleon to drastically reduce the number of figures to search for from the database.
2. The *capturing windows* step takes screenshots of windows for files with at least one entry in the database (that is, at least one figure that should be augmented). The list of applications monitored can further be specified by the user.
3. The *locating figures* step analyses these screenshots to localise figures that should be augmented.
4. The *augmenting figures* step augments the located static figures using the associated HTML5 resources.
5. The *tracking figures* step detects changes in the position and size of the static figures to overlay the augmented versions; thus Chameleon does not need to repeat the whole analysis process for a figure that is already augmented.
6. The *figure synchronization* step redirects mouse and key events from identical augmented figures. This step can be very useful for presentation tools (i.e. PowerPoint) to synchronize the presenter and slideshow views. In this way the presenter can interact with the presenter view and the results are directly reflected in the slideshow.

#### 6.3.2.1 *Step 1: analysing opened files*

The augmentation tool monitors file access on the system using DTrace [405] on Linux and macOS. It retrieves the size of each opened file and queries the database to get the list of files with augmented figures of the same file sizes. Finally it computes the MD5 for the remaining files to filter the remaining documents. We avoid computing the MD5 for all opened files as it can be relatively long for large files.

#### 6.3.2.2 *Step 2: capturing windows*

The augmentation tool takes a screenshot of every window containing documents to be augmented, using the XGETIMAGE [422] function on Linux, and CGWINDOWLISTCREATEIMAGE [13] on macOS. These functions provide the content of a window even if partially hidden, without decorations.

#### 6.3.2.3 *Step 3: locating figures*

Chameleon uses feature matching algorithms to determine regions and descriptors in the screenshots and detect potential figures to be augmented.

This step takes the longest time in the augmentation process. We therefore re-compute regions and descriptors only when the window is scrolled or resized. The feature matching algorithm then tries to associate the regions found with the ones stored in the database to detect the figures present and determine their location and size.

#### 6.3.2.4 *Step 4: overlaying interactive figures*

For each figure found, the augmentation tool opens an undecorated window with transparent background. These windows cannot be focused, resized or moved. Their size and position are dynamically updated to be the same as the figures to augment. Each window contains a `QWEBENGINEVIEW` [312] used to load the remote HTML content located at the URL associated with the figure to augment. Users can interact with augmented figures using the mouse cursor. In addition, they can hide and show augmented figures by using a button located at the top left corner useful to review the original. Users can also get additional information about the augmented figure in a drop down menu and switch between different augmented figures available in the database (in the case where multiple augmented figures exist for an individual figure in the source document). Augmented figures are pre-loaded in background when opened files containing augmented figures are found in step 1. The window is first fully invisible and made visible when a figure is found in step 3. We use a 500ms delay and a 1000ms fade-in effect, tuned via informal pilot testing, to inform the user that a figure is augmented and to let her verify that the augmented figure matches the static one.

#### 6.3.2.5 *Step 5: tracking figures*

Linux provides an API called AT-SPI (Assistive Technology Service Provider Interface) [115] that can send a notification when a window is scrolled or resized. macOS provides the same service through its Accessibility API [12]. Window translation is tracked by pooling their position when a mouse drag event is detected. Each window displaying an augmented figure registers callbacks to be notified when one of the properties of the window displaying the document changes. These callbacks are called with a small latency making it possible to re-position and re-size the augmented figures with hard-to-notice visual artefacts. When a figure is cropped (typically during or after scrolling), we apply a mask on the augmented window so that it is cropped in the same way and does not extend outside the document window. This step is not mandatory for using Chameleon, as overlaying interactive figures over a static document can be achieved without it, but it results in a smoother integration of the augmented figures.

#### 6.3.2.6 *Step 6: figure synchronization*

When Chameleon detects multiple identical figures on the screen, the overlaid augmented versions can be synchronized. The augmented figure with the

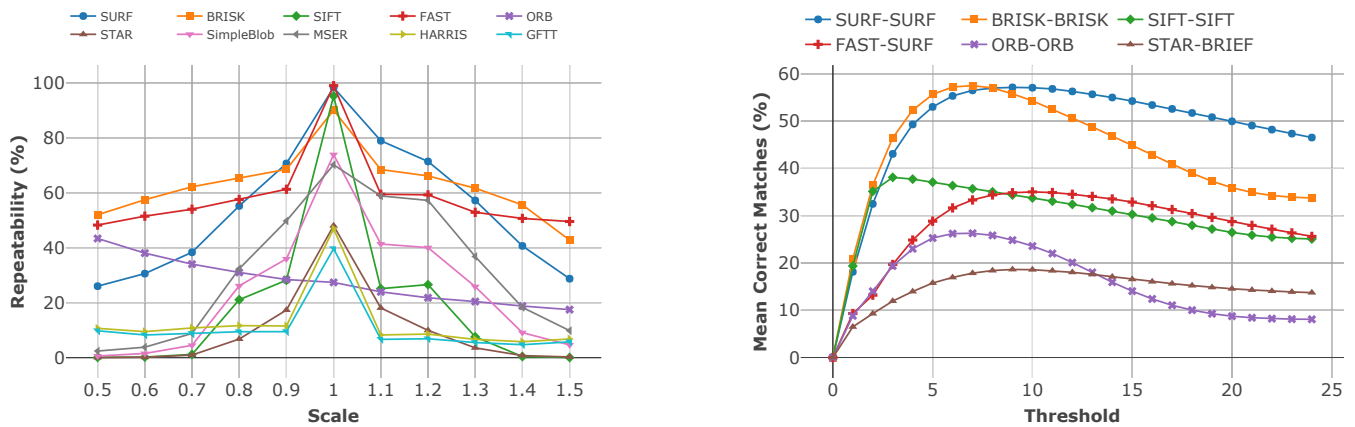


Figure 6.2: *Repeatability vs scales, and Mean Correct Matches vs thresholds of our dataset.*

highest resolution is tagged as the main figure; others are considered copies and resized to match their overlaid static figure resolutions. All the events received by the copies, e.g. keyboard and mouse events, are redirected onto the main figure. This step, enabled by default, can be disabled via Chameleon’s menu.

### 6.3.3 Authoring augmented figures

Chameleon augments documents using HTML5 resources, easing sharing in a community of users. Authoring interactive content is a matter of creating and publishing on-line the HTML5 resources to be used for augmentation, with the only requirement of an aspect ratio similar between the HTML5 resource and the figure. Existing tools can be leveraged depending on the type of interactive figures to be overlaid in a document. For example, a video can be embedded by first uploading it on a video sharing website such as *YouTube* and then using the embeddable URL with Chameleon. From an existing R script, outputting an interactive chart in lieu of a static one can be done without any change to the code using *plotly* for R [301]. Finally, when the interactive figure is already available online, we provide a JavaScript function capable of isolating a specific element of a webpage (i.e. hiding everything except the element) resulting in a webpage, containing only the figure, that can then be used in Chameleon.

*Alternatively, ChartDetective presented in the previous chapter could help generate interactive charts.*

In the following three sections, we present three user studies. The first two user studies drive the design of Chameleon. The first examines the performance of ten different feature matching approaches to determine which works best for the specific use-case of Chameleon. Next, a second experiment evaluates the two best performing algorithms to determine their ability to precisely overlay augmentations onto pre-existing document figures by calculating the average overlap precision between a figure and its augmentation. Finally, we explore the usability of Chameleon via a summative, first-use study [415].

## 6.4 FEATURE MATCHING ALGORITHM COMPARISON

Chameleon requires feature matching to accurately locate figures displayed on screen. Many feature matching algorithms have been proposed, but it remains unclear which ones are the most promising for scale-independent location of figures in static documents. In this section, we compare the performance of various feature matching algorithms against these needs.

Feature matching between two images A and B can be described as a 5-step process:

1. For each image, a first algorithm called a *Detector* is applied to extract a certain number of key regions (described by an x;y coordinate and a diameter).
2. Then, a second algorithm called a *Descriptor* takes the regions identified by a given detector and returns a *descriptive vector* of numbers describing each of these regions.
3. Next, either via brute force or a dedicated algorithm, the system measures a *distance* between the *descriptive vector* of each key region of image A and B (using Hamming or Euclidean distances), and returns a *list of matches* between both images (tuple, one for each image).
4. After that, only the *matches* with a *distance* below a specific *threshold* are kept in order to eliminate most incorrect matches. This results in a *list of associated regions*.
5. Finally, the system takes the *list of associated regions*, as well as the x;y coordinates of these key regions to identify an appropriate homographic transformation using the RANSAC algorithm [106].

### 6.4.1 Comparing feature matching algorithms

Feature matching algorithms are usually compared with a set of natural images using the following metrics:

- *Repeatability*: Evaluates the performance of *detectors* (step 1 above) by measuring the proportion of similar key regions extracted from two images containing the same object [261].
- *Precision and recall*: Evaluates the performance of a *detector-descriptor* pair (step 2 and 3 above) by measuring the number of key regions from a first image that are correctly associated to key regions from a second image [260].

However, already published comparisons of feature matching algorithms suffer from several limitations making them unhelpful to find the most adapted algorithm for Chameleon's needs. First, algorithms are tested on natural images and photos, while documents contain a wider variety of figures (e.g. state diagrams, photos, bar charts, and other data visualizations). Second, robustness to scale is seldom measured during evaluations of algorithms,

whereas it is the main deformation that Chameleon faces since documents are displayed at different scales. Third, precision and recall are less adapted to compare detector-descriptor pairs when different detectors are tested: some detectors yield no key regions for harder-to-detect objects. Thus, descriptors associated with those detectors would be given an unfair advantage because they would not be tested on figures with harder-to-detect features. To address this issue, we use (instead of the precision and recall) a *Mean Correct Matches* metric that computes the mean percentage of correctly associated regions per image. This metric takes into account detectors yielding no key regions and decreases the overall percentage of correct matches accordingly.

#### 6.4.2 Dataset

We evaluate the different feature matching algorithms on two datasets. The first is a *scientific papers* dataset introduced by Clark et al. [56], composed of 150 research articles from 3 different conferences (NIPS, ICML and AAAI) from 2008 to 2014. All figures in this document set were annotated manually. In order to add diversity to the figures, we gathered a second *presentation* dataset. 100 presentations were randomly selected in 10 different categories from the SpeakerDeck website [102]. We extracted from the first 20 pages of each presentation all images whose height and width were larger than 50 pixels. In total, the dataset comprises 1660 figures from 2741 pages. Each of these figures was matched against the PDF page containing the figure, rasterized at 72 DPI. To evaluate the influence of scaling on the results of feature matching algorithms, we applied a scale transformation to the rasterized PDF pages. Tested scales were comprised of every 0.1 scaling step in [0.5, 1.5], and the resizing was based on a bilinear interpolation, the technique observed in Adobe Acrobat Reader DC version 2018 on macOS Sierra running OS X version 10.12.6. Knowing the position and the size of the figure in the scene, we compute as ground truth the homography relating a figure to its PDF page.

#### 6.4.3 Results

To reduce the pairs of algorithms tested, we first evaluate detectors, and then match descriptors with the best detectors.

##### 6.4.3.1 Detector evaluation

We choose to evaluate all the detectors implemented by default in OpenCV 4.0.1. Those detectors include the most commonly used (BRISK [224], FAST [327], GFTT [354], Harris [150], MSER [110], SIFT [236], SURF [23]) as well as more recent ones (ORB [331], SimpleBlob, STAR [6]).

Figure 6.2 shows the mean repeatability score over our two datasets, by scale level, for each detector. Unsurprisingly, almost all detectors perform best at a scale level of 1. The only exception is ORB which, while performing

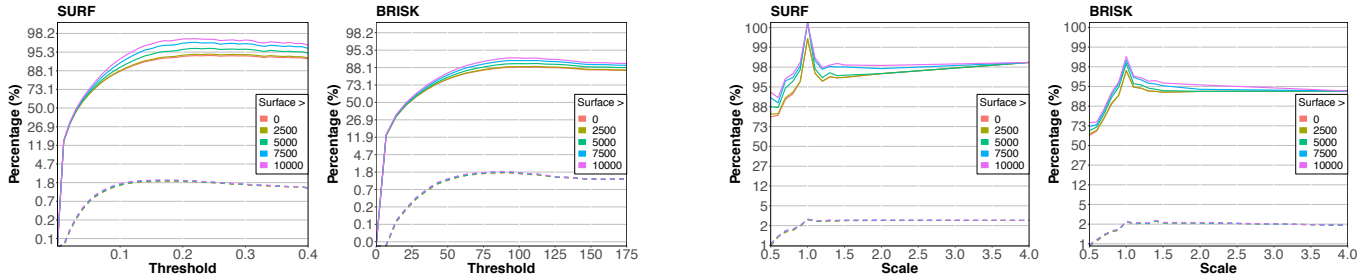


Figure 6.3: Evolution the true positive rate and false positive rate (dashed lines) depending on the feature-matching algorithm threshold and scale at which a figure is displayed. The y axis uses a logit scale.

relatively poorly for all scale levels, performs better for smaller scale levels, explained by the fact that the key regions produced by ORB are larger on average than for the other detectors and that the repeatability metric relies on the overlapping of regions. Therefore, when the scale of image B is reduced, the overlapping ratio is likely to increase [261].

Overall, FAST, SURF and BRISK achieved the best results across all scale levels. Their repeatability score is close to 100% (respectively 99.8%, 99% and 95%) at a scale of 1.0, and, except for SURF, is above 50% for all other scale levels. Other algorithms either performed very poorly or are not robust to scale variation, which is a critical criteria for Chameleon. SIFT used by Sikuli [416] performs well for a scale of 1.0 but its performance quickly drops at other scale levels.

#### 6.4.3.2 Descriptor evaluation

Descriptors need key regions to be extracted by a detector; therefore we associate them with their corresponding detectors, as described by their authors. FAST was also included in the evaluation even though it is only a detector because of its high repeatability. In order to determine the descriptor algorithm to use with FAST, we piloted its association with every descriptor algorithm implemented in OpenCV 4.0.1, using a random subset of our image data. These pilot tests suggested that the descriptor SURF worked best. Thus, we compared BRISK-BRISK, FAST-SURF, ORB-ORB, SIFT-SIFT, STAR-BRIEF and SURF-SURF, with their default parameters.

Figure 6.2-right shows the evolution of *Mean Correct Matches* for different *thresholds*. We associated a unique number to each threshold tested in order to show the results in the same chart. All algorithms were tested using 25 different thresholds, in the range [0-175] for BRISK-BRISK, [0-0.25] for FAST-SURF, [0-100] for ORB-ORB, [0-425] for SIFT-SIFT, [0-75] for STAR-BRIEF and [0-0.25] for SURF-SURF.

Overall, we observe that SURF-SURF and BRISK-BRISK outperform the other algorithms, both of them reaching a *Mean Correct Matches* of more than 55% while the other algorithms do not exceed 40%. These results have several implications. First, algorithms that, reportedly, outperform SURF or SIFT perform more poorly on our dataset (e.g. ORB and STAR). Second, algorithms

used in the literature for similar motivations (i.e. figures and GUI elements) did not offer the best performance (typically SIFT, used in [416], which was outperformed by both SURF and BRISK). Finally, these results help drive design, including an exploration of options for implementation. For example, because SURF is patented, one could decide to use BRISK to obtain similar results.

## 6.5 FEASIBILITY EVALUATION OF CHAMELEON

The comparison of feature matching algorithms described in the previous section suggests that BRISK-BRISK and SURF-SURF would be the best candidates to perform online feature matching in the context of Chameleon. However, this comparison does not reflect Chameleon’s performance for dynamically augmenting digital documents. Therefore, we conducted an evaluation to measure the number of figures correctly identified in a document by these algorithms.

### 6.5.1 Method

We used the same dataset as in the previous experiment (totalling 1660 figures). While using a computer, documents can be displayed at different scale levels; therefore, we tested for each *page* (presentation slide or PDF page) the 12 following *scale* factors: 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 2.0, 4.0. For each image pair, image A being a figure and image B a scaled page, we computed the correspondences by using either BRISK-BRISK or SURF-SURF (referred as BRISK and SURF for simplicity in the next paragraphs). A figure was considered found when the rectangle obtained from the correspondences had positive width and height and its aspect ratio was similar to the figure identified (i. e.  $abs(1 - aspectRatioA/aspectRatioB) \leq 0.1$ ). Based on this identification, we classified every outcome of the algorithm into one of these categories :

- True positive (TP): the figure is present and found
- True negative (TN): the figure is not present and not found
- False positive (FP): the figure is not present but found
- False negative (FN): the figure is present but not found

For TPs, we also computed the overlap ratio of the rectangle found with the real rectangle of the figure in the page.

### 6.5.2 Results

Because feature-matching performance is likely to be impacted by the *surface area* (*surface*) of the figure in pixels (height×width×scale), we report the results by categorizing the figures in 5 groups depending of their surface:  $surface \geq 0$ ,  $surface \geq 2500$ ,  $surface \geq 5000$ ,  $surface \geq 7500$ ,  $surface \geq 10000$  pixels. 10000 pixels thus corresponds to an image of  $100 \times 100$  pixels at scale 1.0.



**ACCURACY** As seen previously, feature-matching algorithms match key-regions from different images if the euclidian-distance between their respective descriptive vectors is below a certain threshold. Choosing a value for this threshold is the first step towards using feature-matching algorithms to identify figures displayed in documents. Figure 6.3 illustrates the proportion of TP and FP (dashed lines) of BRISK and SURF depending on this distance threshold and figure sizes. As we can see, the proportion of TP rapidly increases for both algorithms to reach a cap value. This cap value seems to be reached at a similar threshold value, regardless of the size of the figure, suggesting that we can choose a single threshold for all figures. Therefore, the threshold yielding a high TP rate of while keeping a low FP rate, was 112 and 0.22 respectively for BRISK and SURF.

The two rightmost graphs in Figure 6.3 respectively show the proportion of true and false positives for BRISK and SURF using these thresholds, for each scale factor and depending on the size of the figure. Overall, the proportion of TP (respectively FP) was of 88.4% (1.5%) for BRISK and 94.4% (1.8%) for SURF over the whole dataset. Unsurprisingly, we observe a peak of performance at scale 1.0 for both algorithms, regardless of figure surface, which can easily be explained by the fact that this is the optimal testing condition in which the figure contained in image B (the page of the tested document) has the exact same size as image A (the figure we are looking for). More interestingly, we observe that the proportion of TP barely decreases for scales higher than 1.0 (98.6% TP for SURF and 94.1% for BRISK above scale 1.0), whereas it decreases more significantly for scales lower than 1.0. This can be explained by the fact that scaling down the size of the figure may make the task more difficult for detector algorithms, resulting in different key regions extracted. In addition, we observe that the lower the size of the figure, the more the proportion of TP decreases for smaller scales. In addition to reaching a lower peak of performance than SURF, we can see that BRISK is also more impacted by scale factors decreasing by 4 points between scale 1.0 and scale 4.0, and 32 between scale 1.0 and 0.5, whereas SURF decreases 2 and 18 points, respectively. Finally, the average overlap rate for true positives is over 98% for both algorithms, meaning that when the image is found, the position and size are usually correct.

We also measured the time taken by SURF and BRISK on average, for each document, to 1) extract key regions of the image candidate to augment using the detector, 2) compute their descriptive vectors, 3) match them with the database and 4) compute the homographic transformation if a figure that should be augmented was found. Without any optimisation (such as multi-threading or GPU), assuming the document is displayed in 1080p, the time to find a figure on a 3.2GHz Intel Core i5 is approximately (in ms):  $t_{SURF} = 300 + 80 * x$  and  $t_{BRISK} = 350 + 90 * x$  with 'x' as the number of augmented figures for a given document.

In the context of Chameleon, we use SURF over BRISK because it yields a higher proportion of TP, while keeping the proportion of FP below an acceptable threshold of 2%.

## 6.6 FIRST-USE STUDY

Our previous experiment demonstrates that Chameleon can accurately augment digital documents. In this section, we investigate whether (**Q1**) users can use Chameleon; (**Q2**) they find it faster and more convenient to use than existing solutions; and (**Q3**) they would use Chameleon in everyday life.

## 6.6.1 Procedure

For this study, we recruited 12 participants (23 to 42 age range, mean = 28, 6 identified as female and 6 identified as male), all graduate students in computer science as they are the most likely to have used, or at least know, alternative methods to augment digital documents. Participants were first shown an interactive demonstration of the bubble cursor [138] and the original paper presenting the technique. They were then asked “How would you include this interactive figure in the document?”. Their answers were manually transcribed. The interviewer then introduced them to Chameleon and showed them how to use it through an example. Participants were then asked to use Chameleon in the three following scenarios described below. For each scenario, we measured the time to accomplish the task. Participants then completed a questionnaire and answered open-ended questions. Finally, we showed participants a document augmented using Chameleon and the same document without Chameleon but with an external web page containing all the interactive figures. Participants had to choose the one they preferred and explain why.

**Scenario 1: Interactive Demonstration in Paper.** The bubble cursor [138] is a pointing technique which reduces the distance to a target by increasing the activation area of targets. While the original paper includes *comic strips* of images to explain the technique, as Ben Schneiderman notes, “*An interface is worth a thousand pictures*” Hence, we implemented an interactive version of the bubble cursor using an HTML canvas and Javascript. We then asked participants to augment Figure 6 of the original bubble cursor paper with the interactive resource that we provided.

**Scenario 2: Supporting Explorable Multiverse Analyses.** Dragicevic et al. [91] proposed Explorable Multiverse Analyses Reports (EMAR) as a way to increase the transparency of research papers. However, they acknowledge that the main obstacle to their adoption is the lack of tools to support their creation and integration. Chameleon partially solves this problem by supporting interactivity in both archived and new research papers. For this scenario, we used the mini-paper *Prior* accessible online [90] and created by Dragicevic et al. In this case, the interactive figure was already created and we wish to include it into a PDF document. We extracted the figure by using Javascript and hiding everything but the `<div>` containing the interactive figure. We also transcribed the mini-paper to PDF. We then asked participants to create an interactive document by overlaying the interactive figure in the PDF document.

**Scenario 3: Adding Interactivity to Presentations.** For this final scenario, we asked participants to augment a PowerPoint presentation with two slides: one containing a static image and a link to a *YouTube* video and the other a static image of a bar chart generated using plotly and R [301]. We gave participants the export link provided by YouTube, and a link to the HTML file as generated by plotly and R. We then asked participants to augment the presentation by including the video and the interactive version of the chart directly into the presentation.

### 6.6.2 Results

When asked how they would include interactive figures into a document, 6 participants answered that they would include a link to the interactive figure in the document, 9 participants proposed transcribing the document to HTML, 1 participant proposed converting the figure to Flash and including it in the PDF and, finally, another participant proposed creating a new document viewer supporting interactive figures.

Participants completed the scenarios without difficulty in 58s (SD=18s) on average for scenario 1, in 58s (SD=17s) for scenario 2 and 1m32s (SD=25s) for scenario 3 (Q1).

Although they were not asked to perform a controlled (A vs B) study with the approaches they described, all 12 participants agreed that they would rather use Chameleon over their initial described approaches. However, 3 participants noted that Chameleon was not easier or faster than adding a link to the interactive figure in the document but agreed it gave a better result (*"I think the URL is easier, faster... but the result is better... definitely better in... using this tool"*). Similarly, 5 participants believed that transcribing the document to HTML might give a similar or possibly better result for the reader but that Chameleon was easier and faster to use (Q2).

Using a 5-point Likert-type scale, participants also rated the integration of interactive figures in documents using Chameleon when scrolling (Mdn=4.5, SD=0.9), scaling (Mdn=4, SD=1.2), moving the window (Mdn=4, SD=1) and if the augmentation was sized to the static figure (Mdn=5, SD=0.4). Finally, all participants answered yes to the question "Would you be willing to put extra effort in order to create interactive figures if you could include them in your documents" (Q3).

While some participants noted that having figures in another view allows scaling and viewing of the figure independent of the document, all participants preferred having the interactive figures within the document. One participant mentioned that *"You do not necessarily know where the figures [on the external view] are in the document... Instead [with Chameleon] you always have the figure close to the text referring to it."*

## 6.7 DISCUSSION

The final distribution of Chameleon includes both the registration tool to create and the augmentation tool to interact with explorable explanations. As noted in the section describing the augmentation tool within Chameleon, by combining delay with fade-in, we allow users to perceive the original figure in the document and see the change to an augmented version of the figure helpful to identify rare cases of mismatch, and inform users of an augmentation. Alongside these effects, Chameleon supports various configuration options. For example, some users may wish to have the original figures viewable by default and to intentionally invoke augmentation when desired; for these users, Chameleon can be configured such that augmentations have to be activated explicitly for figures via the button mediator as no augmentation is displayed initially. Similarly, if in augmented mode, figures can be turned off via this same button, allowing users to switch between the original static versions and the augmented versions flexibly during document reading. Users also have the option to place augmented figures in floating windows (i.e. resizable and movable) if they wish to manipulate them independently from the document.

Chameleon relies on a feature matching algorithm carefully chosen through a systematic analysis of the detector and descriptor algorithms of the literature. Because these algorithms were never tested on figures commonly included in documents and on varied scale factors, we performed this analysis and evaluated both the accuracy and time of these algorithms on a real-world data set. Results show the very good performance of SURF over a wide range of scale levels. The lower percentage of TP below scale 1.0 is not critical as users are less likely to expect the augmentation of a small-sized figure given the limited interaction space it offers. SURF also allows for real time augmentation with processing times around 600 ms.

6.7.1 *Chameleon in Practice*

*Augmentation Longevity.* Because augmentations are not embedded in the documents, they suffer from the same issues as files hosted online. Augmentations could be lost if their host disappears or they are otherwise removed. This would result in a document without interactivity; the document would only display the original static figures (as if Chameleon was not enabled). It may be possible to mitigate this using peer-to-peer hypermedia protocols such as IPFS [175]. An augmentation lives as long as one of the node in the network has it stored.

*Application Scalability.* As additional users leverage Chameleon and as augmented figures become more commonplace, scalability concerns regarding the number of figures in a document, the number of figures on-screen, the number of augmented documents that exist in the world, and the number of simultaneous users may concern some readers. Chameleon deals with these issues as follows. First, because the cost of each additional figure in a

document is relatively small (a fact we note when evaluating Chameleon in our second study), multi-figure documents are still handled in a reasonable time by Chameleon: for example, with 5 augmented figures in a document, it would take 700ms for them to appear the first time, (e.g. 10 = 1.1s; 20 = 1.9s). Similarly, multiple on-screen figures can be managed because, as we note earlier during design, transformations do not have to be re-computed once figures are found; Chameleon can simply track position changes on the display. Finally, modern database systems can easily handle large numbers of documents and users. Documents are filtered using their MD5, data storage is limited to descriptive vectors not figures, and embedded resources are linked client-side. As well, modern database systems have evolved to handle large user loads.

*Privacy.* Chameleon works by taking screenshots of some windows. However, to preserve privacy, these screenshots are not publicly transmitted; instead, they are analyzed locally, descriptive vectors are extracted locally, and only these descriptive vectors are stored in the database and used to match figures with their augmentations. Chameleon can also be used to provide interactive content to more confidential documents via databases that support user or group security. In essence, privacy issues are limited as Chameleon only exchanges image feature vectors with a database, not entire images. This is not to say that there are no privacy considerations. If a user is augmenting a figure in a document, then the server delivering the augmentation will know which document is being read on a user's computer. However, any cloud based document system suffers from similar problems: as an example from the field of Human-Computer Interaction, we often leverage the ACM digital library during our research, meaning that this digital library is aware of topics being examined by researchers, and tools such as Overleaf have significantly more information on documents we create. If a user wishes to avoid this awareness, it is also possible to run Chameleon locally by downloading a local copy of the database (whose size should remain minimal considering the small amount of information recorded for each document). Finally, users can disable Chameleon if necessary for certain documents/applications if they would prefer complete privacy at the cost of losing augmentations to figures in the documents/applications.

*Chameleon at the Community Level.* The philosophy behind Chameleon, alongside augmenting documents for the user, is to support both canonic and community augmentations of a document. By default, the creator of the static document should be identified and have the ability to augment the document with canonical augmentations. For new documents, these canonical augmentations can be introduced at the time of document creation, supporting reactive behaviours, dynamic experimentation with parameters, interactive widgets to demonstrate input techniques, open-data collected via experimentation, and augmented information referencing follow-on research. Where possible, archived versions of PDF documents can also be easily augmented post-hoc by their creator, ensuring that even archived documents continue to evolve over time. Alongside creators, other members of the community are also able to augment a document with their own augmentations if they wish. We

believe that users could benefit from noncanonical augmentations, in the form of a community augmentation (similar to pull requests on GitHub which let a user tell others about changes she has pushed to a repository, and discuss and review the potential changes with collaborators). In the end, the creator of the document should decide whether or not the community augmentation will systematically augment the document. Even if the creator is not available to create or promote to canonical augmentations, the community could still discuss, vet, and promote augmentations that allow the large archive of PDF documents to incorporate reactive features.

## 6.8 CONCLUSION AND FUTURE WORK

Inspired by Victor's work on the concept of explorable explanations, we present Chameleon, a system that leverages computer-vision based feature matching and a database of active content to graft interactivity onto static figures within a corpus of documents. This chapter describes an evaluation of various feature matching algorithms to design the system. As well, we present the two tools within Chameleon, a registration tool to allow a user (e.g. the document creator) to graft the interactive figure onto the original document, and an augmentation tool that allows a reader to interact with the augmentation. Because Chameleon works on screen-based pixels, it can augment both new documents and pre-existing documents, thus allowing our pre-existing archive to benefit from interactivity in support of a better interactive experience when consulting static documents. Based on a systematic evaluation of descriptors and detectors, we identified SURF as the best algorithm given its overall high performance and robustness to scale adjustments.

Our implementation was designed to work with desktop applications, but Chameleon could also be integrated to mobile devices, assuming that the mobile OS allows a service to run in the background, analyze displayed content, and overlay HTML5 views on top of content. If the OS does not allow background services, Chameleon could relatively easily be integrated into open-source systems like Android. An implementation of Chameleon on mobile is left as future work.

The already good accuracy and performance of Chameleon can be optimized in several ways that we plan to explore as future work. One approach could be to directly integrate Chameleon in a window manager, providing direct access to the pixels of each window before being displayed. Augmented figures could be directly drawn in the pixel buffer of the window, making the augmentation flawless. The window manager being responsible for handling window movement and resizing, we would know exactly when those actions happen and apply them on the augmented figures. Finally, the registration tool of Chameleon could provide an indicator showing the percentage of similarity with figures already registered in the database for the document in order to notify the users that an augmentation is likely to result in a false positive.

# 7

## CONCLUSION AND FUTURE DIRECTIONS

---

This thesis was driven by the goal of improving the reading experience of existing digital documents. Specifically, my thesis was that

*Existing documents can be transformed and enhanced with interactive, visual, and textual representations that help readers understand information.*

To test this hypothesis, we proposed novel representations that draw upon cognitive psychology theories and that can be generated from existing documents without involving authors. Below, I summarize the contributions of this thesis, discuss implications, and outline future directions.

### 7.1 SUMMARY OF CONTRIBUTIONS

This thesis contributes fundamental knowledge related to the design of representations and their benefits, and the design of interactive systems and their applications. Below, I categorize these contributions using the taxonomy proposed by Wobbrock and Kientz [409]. Artifact contributions are further divided using the classification proposed by Fogarty [108].

#### 7.1.1 *Artifact Contributions*

This thesis presents four interactive systems that contribute by accomplishing novel functionalities: Charagraph allows annotating documents with charts and turning data-rich paragraphs of text into visualizations; Statslator lets readers translate statistical reports to make them more familiar by offering alternative values and visualizations; ChartDetective enables the recovery of accurate data underlying static charts; Chameleon allows incorporating interactive content within documents, including documents using static file formats, or already disseminated.

These four systems also contribute in how they accomplish their goals: Charagraph introduces an interactive extraction system to recover numbers in text and several interactions to customize and merge generated visualizations; Statslator outlines a text extraction pipeline with error mitigation features and statistical checking functionalities; ChartDetective proposes filtering features and chart extraction algorithms for vector graphics; and Chameleon adapts feature matching algorithms to the task of detecting figures in documents.

## 7.1.2 Empirical Research Contributions

The implementation and validation of the interactive artifacts lead to several empirical contributions: A user study found that participants could annotate documents with Charagraphs and benefited from having these visualizations; an analysis of CHI papers detailed how statistics are typically reported in scientific documents; three simulations found that p-values and confidence intervals reported in papers could be converted with reasonable accuracy; a user study found that participants could extract the data from complex charts when using ChartDetective; an experiment found that the data obtained using ChartDetective is highly accurate; a simulation found that feature matching algorithms could be used to detect figures in documents reliably; a first-use study highlighted the usability and the benefits of Chameleon over alternatives.

## 7.2 AUGMENTING DOCUMENTS BY READERS, PUBLISHERS, OR AUTHORS?

There are many ways a document can become polymorphic. The main limitation being the information available to the actor performing the “augmentation”. Following the linear life-cycle of documents, three actors can be considered: first authors (fig. 7.1a), then publishers (fig. 7.1b), and finally readers (fig. 7.1c). This thesis focused on augmenting documents during this last reading phase but polymorphism could happen earlier. Below, I reflect on these different phases and the implications of involving these different actors to augment a document.



Figure 7.1: Life-cycle of a document with respect to the actors and resources available. Solid lines indicate the resource is available. Dashed lines indicate the resource is less likely to be available. As such, (c) readers are more likely to have access to the document than the code or data. Inversely, (a) authors have access to all code, system, and data.

*Readers.* Augmentations happening at the reading phase are most impactful because they apply to all documents, even ones published decades ago. This is also the most direct form of augmentation: whereas authors guess what the audience needs to know, readers know what they need. However, unless authors followed open science practices, data and code are unlikely to be available. This means that polymorphic content has to be constructed from the information already in the document. Such informationally equivalent representations may still benefit readers, as we have shown throughout this thesis.



Alternatively, additional information can be retrieved through other means such as external databases (this is the idea proposed by NewsView [118] and Contextifier [170]). In all cases, just-in-time augmentations made by readers are most likely to be adopted if they require a low effort.

*Authors.* Perhaps because authors have access to all information such as code, analytical decisions, data, and results, it is often expected that they should be the ones creating interactive documents. Most previous work looked at helping authors craft these kinds of documents. While this would be ideal and allow polymorphic representation that are informationally augmented, in this thesis I argue that the burden for authors makes it unlikely to be widely adopted. Besides, authors have no guarantee that their work will actually benefit readers as it is difficult to know in advance the utility of an interactive figure, for example.

*Publishers.* An alternative that would remove burden from authors while allowing the use of additional data is to involve publishers. Authors could submit their document and data, and publishers would take care of generating a document with polymorphic content. Some publishers have taken steps in this direction: the ACM Publishing System (TAPS) requires authors to submit their source files. Internally, the files are processed to generate multiple outputs, including an HTML version of the document that is supposed to be more accessible [383].

*Community.* Another option is to turn documents into collaborative environments where anyone can contribute and add polymorphic content. This is similar to a Wiki [223], except the initial document is already self-contained, and the added polymorphic representation is only meant to augment this initial information. Again, such an approach would run into issues similar to the ones faced in the project of this thesis in that data might not be available. But the advantage is that a community of users might be more willing to spend the time to craft advanced representations. Chameleon is essentially enabling this vision but for existing documents.

### 7.3 LIMITATIONS

While our results provide evidence that the reading experience can be improved for existing documents, there are several limitations that might require further investigation.

*It is unclear if all polymorphic representations are useful.* The design of the representations proposed in this thesis are based upon theories and experimental findings from previous work. While we provide evidence that it would help readers, only Charagraphs were tested in a controlled lab setting with an A versus B protocol to see the effect on reading comprehension. For Statslator, there is good evidence that providing familiar values will help, but we did not validate this experimentally. Similarly, Chameleon was tested for usability, but it is unclear that the added interactive figures help readers. This is a common issue with interactive documents: much work considered including dynamic content [163], yet testing the actual impact for readers is difficult.

Instead, most studies (like the ones in this thesis) resort to proxies such as time spent and number of clicks that might not give the full picture [61].

*If useful, it remains unclear what aspects of the polymorphic representations are most beneficial.* Many of the representations proposed could be considered complex systems in that they combine multiple smaller components and improvements. Most of these components were carefully chosen through iterative design processes, but are hard to evaluate independently. As Olsen puts it “good systems deal in complexity and complexity confounds controlled experimentation.” [282]. Thus, while some experiments show benefits of the representations proposed, they provide limited evidence about the specific aspects that make them useful and effective. Similarly, it might be that some cognitive theories are more powerful than others, or better used in isolation. We instead focused on leveraging them in combination.

*If not useful, then polymorphic representations could be detrimental.* For example, interactive media may be distracting. Readers might invest time exploring a representation even though it does not contribute to their understanding and instead take away from time that could be better used reading the document. This distraction factor is hard to measure and might depend on the reader more than the representation itself. The [media richness theory](#) suggests that messages that are complex require richer media to be understood. But it also suggests that a rich media used for a simple “routine” message leads to “data glut” and confusion because of the surplus meaning [222]. The issue is that what qualifies as “routine message” depends on the readers’ knowledge. This is part of the reason the representations in this thesis are generated on-demand (e.g., after a selection) rather than automatically.

*Our findings might not generalize beyond scientific documents and researchers.* Like with all user studies, the results need to be considered with respect to participant demographics. In our case, participants were primarily graduate students and researchers at a Canadian university. Additionally, we focused on scientific documents, and except for Charagraphs that were also tested on news articles and reports, our experiments used only scientific documents. Thus, our results might not generalize to other documents or different populations.

## 7.4 FUTURE DIRECTIONS

Our exploration of representations to complement existing documents can be extended in several ways, either by proposing novel representations, adapting the systems we proposed into authoring tools, and exploring polymorphic representations in new contexts.

### 7.4.1 Exploring the Design Space of Polymorphic Content

Our design space of complementary representations (fig. 1.1) is generative and can help think about novel polymorphic content. Specifically, the representations in this thesis are biased towards visuals with ChartDetective

being the only project looking at turning a visual representation into numbers. The reason is that documents are already mostly text and visuals are often superior to text for memorability [350] and processing efficacy [69, 259, 360]. Thus, after placing the four projects outlined previously, there remain some gaps that could lead to novel ideas (fig. 7.2).

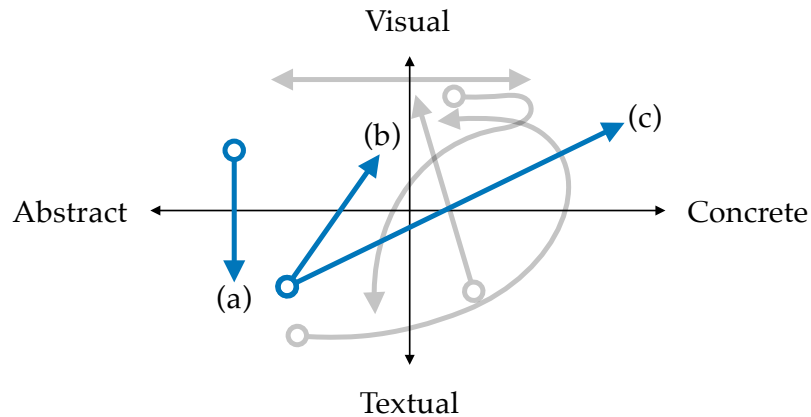


Figure 7.2: Future directions for exploring the design space of polymorphic content. In grey are the directions that this thesis explored. In blue are future directions I believe to be promising: (a) visuals into text; (b) making algorithms more visual; and (c) making math more concrete.

*Visuals into text.* A natural extension would be to explore the conversion of visuals into text (fig. 7.2a). This could be for the purpose of making figures more accessible. In fact, previous work has looked at turning images and charts into detailed descriptions compatible with screen readers [116]. Or, it could be used to generate textual examples from visuals describing processes such as diagrams. For example, a flow chart could be turned into multiple concrete textual examples, one per possible path.

*Making math more concrete.* On the [ladder of abstraction](#), mathematical equations in scientific documents might be one of the most abstract form of textual representation. One future direction could be to make it more concrete (fig. 7.2c). There has been several efforts to make math more concrete by augmenting the equations with annotations [154] or by representing equations with sketches [334]. Alternatively, interactivity could enrich the equations, as per the [media richness theory](#), to execute the equation on example numbers or generate plots and geometric figures [394]. The challenge is to make these augmentations usable directly with the equations in existing documents.

*Making algorithms more visual.* Algorithms presented in scientific documents are another kind of abstract textual representations. One way these could be made more visual would be to convert them to flow charts or block-based programming languages such as Blockly [295]. Even if on the same level of abstraction, these visual representations might help memorization and processing as per the [dual-coding theory](#) (fig. 7.2b). Or, the representation could go down the [ladder of abstraction](#). For example, there have been efforts

using comics to teach programming concepts in a more concrete way [371]. Such representations can be semi-automatically generated [372] and could augment the algorithms reported in scientific documents.

#### 7.4.2 *Helping Authors Craft Polymorphic Documents*

As mentioned in section 7.2, authors could create polymorphic documents from the beginning. The work in this thesis can also inform the authoring of polymorphic documents.

*Tools for readers can help authors.* While the focus of this thesis was on readers, it is interesting to note that many of the tools we proposed could also help authors. For example, Charagraph could be used to quickly generate interactive charts after having written the results section of a document. Essentially, these canonical Charagraphs would be shared with the document directly. Similarly, Statslator could help calculate alternative values and act as a statistical linter to help authors identify typos and misreport of values. And ChartDetective could help turn static data visualizations into interactive ones, especially when the tools used to create the original visualization cannot output interactive charts. Often, tools for readers indirectly help authors.

*Writing with varying levels of abstraction.* An explanation might be better understood if presented through varying degrees of abstraction [152]. Yet, current writing tools often focus on the grammar rather than the content. Instead, a writing tool could identify explanations that are abstract and suggest examples and metaphors to make them more concrete. Inversely, it could identify concrete explanations and offer ways to generalize them.

*Supporting the creation of interactive figures.* Interactive figures are the cornerstone of polymorphic documents as they offer the highest degrees of freedom to support the different ways a representation might vary. Yet, they remain difficult to create. For an author, it is often difficult to convert an idea into an interactive figure, despite the many prototyping software that have been proposed recently. Additionally, it is difficult to think in terms of interactive figures for authors that have been used to seeing and designing static ones. For example, it is unclear how interactive figures should be designed to be most effective. In that regard, I believe some findings in this thesis could shed some light on the design of interactive figures.

#### 7.4.3 *Polymorphism Beyond Documents*

Polymorphism is a way to show the same information through different perspectives. While we used polymorphism to help readers make sense of the information, it could also apply to other contexts beyond documents.

*Polymorphic presentations.* A slideshow is essentially a document that is accompanying a presenter. Many of the augmentation methods proposed in this thesis could apply to slideshows. More interesting is the possibility for the audience to generate their own representations from the slides. For example, an attendee could take photos of slides during a presentation and turn charts

into numbers, turn text into charts, or turn p-values into confidence intervals. Many of the issues a reader might face are also faced by attendees.

*Polymorphic authoring tools.* Polymorphism could be related to “systems that use two or more distinct views to support the investigation of a single conceptual entity” [400]. As such, a polymorphic tool offers several views to modify a single entity. For instance, Sketch-n-sketch [157] is an example of system that offers two views to edit a single SVG drawing. Using the polymorphism framing proposed in this thesis allows to design coherent and useful views: the code view is a form of abstract textual representation, whereas the output view is a form of visual concrete representation. Adding interactivity to both views is a way to make it richer. Similar approaches could be used to design and enhance tools to create charts or 3D elements by providing views leveraging the four quadrants of our design space (fig. 1.1).

## 7.5 FINAL WORD

Thinking about the future of documents does not mean forgetting about documents of the past. And publishing a document does not have to be an end. I propose polymorphic documents as a way to turn existing documents into different forms and representations. Polymorphic documents try to solve the limitations of current digital formats, originating back to when paper was the only document medium. To do so, I investigate different representations of the knowledge within documents through four projects: Charagraph looks at ways to turn data-rich paragraphs into interactive visualizations; Statslator proposes a solution to help interpret statistical reports by translating them into more familiar values and plots; ChartDetective investigates the accurate recovery of data from existing documents to make this data accessible to others, including machines; and Chameleon wonders how to replace figures in existing documents while staying compatible with existing workflows, software, and file formats. All together, these projects explore how much information can be recovered from existing documents, what better representations might convey this information, and how these representations can be generated without involving the original authors.

Whereas approaches relying on document authors to produce better documents have difficulties being adopted, I believe that giving readers the tools to turn existing documents into polymorphic ones has a greater potential to transform our reading experience.

## BIBLIOGRAPHY

---

- [1] Jacob Abbott, Haley MacLeod, Novia Nurain, Gustave Ekobe, and Sameer Patil. "Local Standards for Anonymization Practices in Health, Wellness, Accessibility, and Aging Research at CHI." In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. New York, NY, USA: Association for Computing Machinery, May 2019, pp. 1–14. ISBN: 978-1-4503-5970-2. DOI: [10 . 1145 / 3290605 . 3300692](https://doi.org/10.1145/3290605.3300692). (Visited on 04/05/2023).
- [2] Acland Acland. *From International Blockbusters to National Hits: Analysis of the 2010 UIS Survey on Feature Film Statistics - UNESCO Digital Library*. <https://unesdoc.unesco.org/ark:/48223/pf0000217103>. 2012.
- [3] Mortimer Jerome Adler and Charles Van Doren. *How to Read a Book*. Touchstone hardcover edition. New York: Simon & Schuster, 2014. ISBN: 978-1-4767-9015-2.
- [4] Adobe. *Flash, HTML5 and Open Web Standards*. 2015. URL: <https://theblog.adobe.com/flash-html5-and-open-web-standards/>.
- [5] Adobe. *Flash and The Future of Interactive Content*. 2017. URL: <https://theblog.adobe.com/adobe-flash-update/>.
- [6] Motilal Agrawal, Kurt Konolige, and Morten Rufus Blas. "CenSurE: Center Surround Extremas for Realtime Feature Detection and Matching." In: *Computer Vision – ECCV 2008*. Ed. by David Forsyth, Philip Torr, and Andrew Zisserman. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 102–115. ISBN: 978-3-540-88693-8.
- [7] D. G. Altman and J. M. Bland. "Presentation of Numerical Data." In: *BMJ (Clinical research ed.)* 312.7030 (Mar. 1996), p. 572. ISSN: 0959-8138. DOI: [10 . 1136 / bmj . 312 . 7030 . 572](https://doi.org/10.1136/bmj.312.7030.572). (Visited on 03/23/2023).
- [8] Douglas G. Altman and J. Martin Bland. "How to Obtain the Confidence Interval from a P Value." In: *BMJ* 343 (Aug. 2011), p. d2090. ISSN: 1756-1833. DOI: [10 . 1136 / bmj . d2090](https://doi.org/10.1136/bmj.d2090). (Visited on 03/03/2023).
- [9] Douglas G. Altman and J. Martin Bland. "How to Obtain the P Value from a Confidence Interval." In: *BMJ* 343 (Aug. 2011), p. d2304. ISSN: 0959-8138, 1468-5833. DOI: [10 . 1136 / bmj . d2304](https://doi.org/10.1136/bmj.d2304). (Visited on 03/03/2023).
- [10] R. Amar, J. Eagan, and J. Stasko. "Low-Level Components of Analytic Activity in Information Visualization." In: *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005*. Minneapolis, MN, USA: IEEE, Oct. 2005, pp. 111–117. DOI: [10 . 1109 / INFVIS . 2005 . 1532136](https://doi.org/10.1109/INFVIS.2005.1532136).
- [11] *Apache ECharts*. Apache, 2015. URL: <https://echarts.apache.org/en/index.html>.

- [12] Apple. *Accessibility on macOS*. 2007. URL: <https://developer.apple.com/accessibility/macOS/>.
- [13] Apple. *CGWindowListCreateImage*. 2007. URL: <https://developer.apple.com/documentation/coregraphics/1454852-cgwindowlistcreateimage>.
- [14] American Psychologic Association. *Publication Manual of the American Psychological Association*. 6th edition. Washington, DC: American Psychological Association, Jan. 2020. ISBN: 978-1-4338-0561-5.
- [15] T. K. Attwood, D. B. Kell, P. McDermott, J. Marsh, S. R. Pettifer, and D. Thorne. "Utopia documents: linking scholarly literature with research data." In: *Bioinformatics* 26.18 (Sept. 2010), pp. i568–i574. ISSN: 1367-4803. DOI: [10.1093/bioinformatics/btq383](https://doi.org/10.1093/bioinformatics/btq383). URL: <http://dx.doi.org/10.1093/bioinformatics/btq383>.
- [16] Paul Ayres and Gabriele Cierniak. "Split-Attention Effect." In: *Encyclopedia of the Sciences of Learning*. Ed. by Norbert M. Seel. Boston, MA: Springer US, 2012, pp. 3172–3175. ISBN: 978-1-4419-1428-6. DOI: [10.1007/978-1-4419-1428-6\\_19](https://doi.org/10.1007/978-1-4419-1428-6_19).
- [17] Sriram Karthik Badam, Zhicheng Liu, and Niklas Elmqvist. "Elastic Documents: Coupling Text and Tables through Contextual Visualizations for Enhanced Document Reading." In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (Jan. 2019), pp. 661–671. ISSN: 1941-0506. DOI: [10.1109/TVCG.2018.2865119](https://doi.org/10.1109/TVCG.2018.2865119).
- [18] Aaron Bangor, Philip T. Kortum, and James T. Miller. "An Empirical Evaluation of the System Usability Scale." In: *International Journal of Human–Computer Interaction* 24.6 (July 2008), pp. 574–594. ISSN: 1044-7318. DOI: [10.1080/10447310802205776](https://doi.org/10.1080/10447310802205776). URL: <https://doi.org/10.1080/10447310802205776>.
- [19] Liliana Barrios, Pietro Oldrati, David Lindlbauer, Marc Hilty, Helen Hayward-Koennecke, Christian Holz, and Andreas Lutterotti. "A Rapid Tapping Task on Commodity Smartphones to Assess Motor Fatigability." In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, Apr. 2020, pp. 1–10. ISBN: 978-1-4503-6708-0. (Visited on 05/31/2022).
- [20] Scott Bateman, Regan L. Mandryk, Carl Gutwin, Aaron Genest, David McDine, and Christopher Brooks. "Useful Junk? The Effects of Visual Embellishment on Comprehension and Memorability of Charts." In: *Proceedings of the 2010 CHI Conference on Human Factors in Computing Systems*. CHI '10. Atlanta, Georgia, USA: Association for Computing Machinery, 2010, 2573–2582. ISBN: 9781605589299. DOI: [10.1145/1753326.1753716](https://doi.org/10.1145/1753326.1753716). URL: <https://doi.org/10.1145/1753326.1753716> (visited on 08/29/2022).

- [21] Leilani Battle, Peitong Duan, Zachery Miranda, Dana Mukusheva, Remco Chang, and Michael Stonebraker. “Beagle: Automated Extraction and Interpretation of Visualizations from the Web.” In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. Montreal QC, Canada: Association for Computing Machinery, 2018, pp. 1–8. ISBN: 978-1-4503-5620-6. DOI: [10.1145/3173574.3174168](https://doi.org/10.1145/3173574.3174168). URL: <https://doi.org/10.1145/3173574.3174168>.
- [22] Aaron Bauer and Kenneth R. Koedinger. “Note-Taking, Selecting, and Choice: Designing Interfaces That Encourage Smaller Selections.” In: *Proceedings of the 8th ACM/IEEE-CS Joint Conference on Digital Libraries*. JCDL '08. New York, NY, USA: Association for Computing Machinery, June 2008, pp. 397–406. ISBN: 978-1-59593-998-2. DOI: [10.1145/1378889.1378961](https://doi.org/10.1145/1378889.1378961).
- [23] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. “Speeded-Up Robust Features (SURF).” In: *Comput. Vis. Image Underst.* 110.3 (June 2008), pp. 346–359. ISSN: 1077-3142. DOI: [10.1016/j.cviu.2007.09.014](https://doi.org/10.1016/j.cviu.2007.09.014). URL: <http://dx.doi.org/10.1016/j.cviu.2007.09.014>.
- [24] Fabian Beck and Daniel Weiskopf. “Word-Sized Graphics for Scientific Texts.” In: *IEEE Transactions on Visualization and Computer Graphics* 23.6 (June 2017), pp. 1576–1587. ISSN: 1941-0506. DOI: [10.1109/TVCG.2017.2674958](https://doi.org/10.1109/TVCG.2017.2674958).
- [25] Sarah Belia, Fiona Fidler, Jennifer Williams, and Geoff Cumming. “Researchers Misunderstand Confidence Intervals and Standard Error Bars.” In: *Psychological Methods* 10.4 (2005), pp. 389–396. ISSN: 1939-1463, 1082-989X. DOI: [10.1037/1082-989X.10.4.389](https://doi.org/10.1037/1082-989X.10.4.389). (Visited on 01/12/2023).
- [26] Jacques Bertin. *Sémiologie graphique: Diagrammes, les réseaux, les cartes*. Paris: EHESS, 2013. ISBN: 978-2-7132-2417-1.
- [27] Lonni Besançon and Pierre Dragicevic. “The Continued Prevalence of Dichotomous Inferences at CHI.” In: *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*. Glasgow Scotland Uk: ACM, May 2019, pp. 1–11. ISBN: 978-1-4503-5971-9. DOI: [10.1145/3290607.3310432](https://doi.org/10.1145/3290607.3310432). (Visited on 03/22/2023).
- [28] Lonni Besançon, Yvonne Jansen, Andy Cockburn, and Pierre Dragicevic. *Definitely Maybe: Hedges And Boosters in the HCI Literature*. Preprint. Open Science Framework, Sept. 2021. DOI: [10.31219/osf.io/mjg7h](https://doi.org/10.31219/osf.io/mjg7h). (Visited on 03/23/2023).
- [29] Jay David Bolter. *Writing Space: The Computer, Hypertext, and the History of Writing*. 1st edition. Hillsdale, N.J.: Routledge, Nov. 1990. ISBN: 978-0-8058-0428-7.
- [30] Dennis D. Boos and Leonard A. Stefanski. “P-Value Precision and Reproducibility.” In: *The American Statistician* 65.4 (Nov. 2011), pp. 213–221. ISSN: 0003-1305. DOI: [10.1198/tas.2011.10129](https://doi.org/10.1198/tas.2011.10129). (Visited on 03/23/2023).



BIBLIOGRAPHY

- [31] Michael Borenstein, Larry V. Hedges, Julian P. T. Higgins, and Hannah R. Rothstein. *Introduction to Meta-Analysis*. 1st edition. Chichester, U.K: Wiley, Apr. 2009. ISBN: 978-0-470-05724-7.
- [32] Michelle A. Borkin, Zoya Bylinskii, Nam Wook Kim, Constance May Bainbridge, Chelsea S. Yeh, Daniel Borkin, Hanspeter Pfister, and Aude Oliva. "Beyond Memorability: Visualization Recognition and Recall." In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (Jan. 2016), pp. 519–528. ISSN: 1941-0506. DOI: [10.1109/TVCG.2015.2467732](https://doi.org/10.1109/TVCG.2015.2467732).
- [33] Michelle A. Borkin, Azalea A. Vo, Zoya Bylinskii, Phillip Isola, Shashank Sunkavalli, Aude Oliva, and Hanspeter Pfister. "What Makes a Visualization Memorable?" In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (Dec. 2013), pp. 2306–2315. ISSN: 1941-0506. DOI: [10.1109/TVCG.2013.234](https://doi.org/10.1109/TVCG.2013.234).
- [34] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. "D<sup>3</sup> Data-Driven Documents." In: *IEEE Transactions on Visualization and Computer Graphics* 17.12 (Dec. 2011), pp. 2301–2309. ISSN: 1941-0506. DOI: [10.1109/TVCG.2011.185](https://doi.org/10.1109/TVCG.2011.185).
- [35] Matthew Brehmer and Tamara Munzner. "A Multi-Level Typology of Abstract Visualization Tasks." In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (Dec. 2013), pp. 2376–2385. ISSN: 1077-2626. DOI: [10.1109/TVCG.2013.124](https://doi.org/10.1109/TVCG.2013.124).
- [36] Victor Bret. *Explorable Explanations*. <http://worrydream.com/ExplorableExplanations/>. 2011. (Visited on 06/15/2022).
- [37] Brill. *Brill's Author Guide*. 2023.
- [38] John Brooke. "SUS: A Quick and Dirty Usability Scale." In: *Usability Eval. Ind.* 189 (Nov. 1995), p. 7.
- [39] Michael K. Buckland. "What is a "document"?" In: *Journal of the American Society for Information Science* 48.9 (1998), pp. 804–809. DOI: [10.1002/\(SICI\)1097-4571\(199709\)48:9<804::AID-ASI5>3.0.CO;2-V](https://doi.org/10.1002/(SICI)1097-4571(199709)48:9<804::AID-ASI5>3.0.CO;2-V). URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291097-4571%28199709%2948%3A9%3C804%3A%3AAID-ASI5%3E3.0.CO%3B2-V>.
- [40] Dan Buckley and Lex Wilson. *The Personalisation by Pieces Framework: A Framework for the Incremental Transformation of Pedagogy Towards Greater Learner Empowerment in Schools*.
- [41] Daniel Buschek, Martin Zürn, and Malin Eiband. "The Impact of Multiple Parallel Phrase Suggestions on Email Input and Composition Behaviour of Native and Non-Native English Writers." In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. CHI '21. New York, NY, USA: Association for Computing Machinery, May 2021, pp. 1–13. ISBN: 978-1-4503-8096-6. DOI: [10.1145/3411764.3445372](https://doi.org/10.1145/3411764.3445372). (Visited on 05/31/2022).

- [42] Kelly Caine. “Local Standards for Sample Size at CHI.” In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. San Jose California USA: ACM, May 2016, pp. 981–992. ISBN: 978-1-4503-3362-7. DOI: [10.1145/2858036.2858498](https://doi.org/10.1145/2858036.2858498). (Visited on 03/28/2023).
- [43] Alberto Cairo. *How Charts Lie: Getting Smarter about Visual Information*. Illustrated edition. New York: WW Norton, Oct. 2019. ISBN: 978-1-324-00156-0.
- [44] Robert J. Calin-Jageman and Geoff Cumming. “Estimation for Better Inference in Neuroscience.” In: *eNeuro* 6.4 (2019), ENEURO.0205-19.2019. ISSN: 2373-2822. DOI: [10.1523/ENEURO.0205-19.2019](https://doi.org/10.1523/ENEURO.0205-19.2019). (Visited on 01/12/2023).
- [45] Robert J. Calin-Jageman and Geoff Cumming. “The New Statistics for Better Science: Ask How Much, How Uncertain, and What Else Is Known.” In: *The American Statistician* 73.sup1 (Mar. 2019), pp. 271–280. ISSN: 0003-1305, 1537-2731. DOI: [10.1080/00031305.2018.1518266](https://doi.org/10.1080/00031305.2018.1518266). (Visited on 01/12/2023).
- [46] Russell N Carney and Joel R Levin. “Pictorial Illustrations Still Improve Students’ Learning from Text.” In: *Educational Psychology Review* (2001), p. 22.
- [47] L. F. Carter. “An Experiment on the Design of Tables and Graphs Used for Presenting Numerical Data.” In: *The Journal of Applied Psychology* 31.6 (Dec. 1947), pp. 640–650. ISSN: 0021-9010. DOI: [10.1037/h0054246](https://doi.org/10.1037/h0054246).
- [48] Paul Chandler and John Sweller. “Cognitive Load Theory and the Format of Instruction.” In: *Cognition and Instruction* 8.4 (Dec. 1991), pp. 293–332. ISSN: 0737-0008, 1532-690X. DOI: [10.1207/s1532690xci0804\\_2](https://doi.org/10.1207/s1532690xci0804_2). (Visited on 06/20/2023).
- [49] Olivier Chapuis and Nicolas Roussel. “Copy-and-Paste between Overlapping Windows.” In: *Proceedings of the 2007 CHI Conference on Human Factors in Computing Systems*. San Jose California USA: ACM, Apr. 2007, pp. 201–210. ISBN: 978-1-59593-593-9. DOI: [10.1145/1240624.1240657](https://doi.org/10.1145/1240624.1240657).
- [50] Xiang ‘Anthony’ Chen, Tovi Grossman, and George Fitzmaurice. “Swipeboard: A Text Entry Technique for Ultra-Small Interfaces That Supports Novice to Expert Transitions.” In: *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*. Honolulu Hawaii USA: ACM, Oct. 2014, pp. 615–620. ISBN: 978-1-4503-3069-5. DOI: [10.1145/2642918.2647354](https://doi.org/10.1145/2642918.2647354). (Visited on 04/04/2023).
- [51] Zhutian Chen and Haijun Xia. “CrossData: Leveraging Text-Data Connections for Authoring Data Documents.” In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. New Orleans LA USA: ACM, Apr. 2022, pp. 1–15. ISBN: 978-1-4503-9157-3. DOI: [10.1145/3491102.3517485](https://doi.org/10.1145/3491102.3517485).

- [52] Ed H. Chi, Lichan Hong, Michelle Gumbrecht, and Stuart K. Card. “ScentHighlights: Highlighting Conceptually-Related Sentences during Reading.” In: *Proceedings of the 10th International Conference on Intelligent User Interfaces*. IUI '05. New York, NY, USA: Association for Computing Machinery, Jan. 2005, pp. 272–274. ISBN: 978-1-58113-894-8. DOI: [10.1145/1040830.1040895](https://doi.org/10.1145/1040830.1040895).
- [53] Jinho Choi, Sanghun Jung, Deok Gun Park, Jaegul Choo, and Niklas Elmqvist. “Visualizing for the Non-Visual: Enabling the Visually Impaired to Use Visualization.” In: *Computer Graphics Forum* 38.3 (2019), pp. 249–260. ISSN: 1467-8659. DOI: [10.1111/cgf.13686](https://doi.org/10.1111/cgf.13686). (Visited on 06/15/2022).
- [54] Sagnik Ray Choudhury, Shuting Wang, and C. Lee. Giles. “Curve Separation for Line Graphs in Scholarly Documents.” In: *2016 IEEE/ACM Joint Conference on Digital Libraries (JCDL)*. Newark, NJ, USA: IEEE, June 2016, pp. 277–278.
- [55] Sagnik Ray Choudhury, Shuting Wang, and C. Lee. Giles. “Scalable Algorithms for Scholarly Figure Mining and Semantics.” In: *Proceedings of the International Workshop on Semantic Big Data - SBD '16*. San Francisco, California: ACM Press, 2016, pp. 1–6. ISBN: 978-1-4503-4299-5. DOI: [10.1145/2928294.2928305](https://doi.org/10.1145/2928294.2928305). (Visited on 05/26/2022).
- [56] Christopher Andreas Clark and Santosh Divvala. “Looking Beyond Text: Extracting Figures, Tables and Captions from Computer Science Papers.” In: *AAAI Workshops*. USA: AAAI, 2015, pp. 423–434. URL: <http://aaai.org/ocs/index.php/WS/AAAIW15/paper/view/10092>.
- [57] William S. Cleveland. *The Elements of Graphing Data*. 2 edition. Murray Hill, N.J.: Hobart Press, Dec. 2004. ISBN: 978-0-9634884-1-1.
- [58] Andy Cockburn, Carl Gutwin, and Alan Dix. “HARK No More: On the Preregistration of CHI Experiments.” In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. New York, NY, USA: Association for Computing Machinery, Apr. 2018, pp. 1–12. ISBN: 978-1-4503-5620-6. DOI: [10.1145/3173574.3173715](https://doi.org/10.1145/3173574.3173715). (Visited on 04/04/2023).
- [59] Ashley Colley, Sven Mayer, and Niels Henze. “Investigating the Effect of Orientation and Visual Style on Touchscreen Slider Performance.” In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. New York, NY, USA: Association for Computing Machinery, May 2019, pp. 1–9. ISBN: 978-1-4503-5970-2. DOI: [10.1145/3290605.3300419](https://doi.org/10.1145/3290605.3300419). (Visited on 05/31/2022).
- [60] Matthew Conlen and Jeffrey Heer. “Idyll: A Markup Language for Authoring and Publishing Interactive Articles on the Web.” In: *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. UIST '18. Berlin, Germany: ACM, Oct. 2018, pp. 977–989. ISBN: 978-1-4503-5948-1. DOI: [10.1145/3242587.3242600](https://doi.org/10.1145/3242587.3242600). URL: <http://doi.acm.org/10.1145/3242587.3242600>.

- [61] Matthew Conlen, Alex Kale, and Jeffrey Heer. "Capture & Analysis of Active Reading Behaviors for Interactive Articles on the Web." In: *Computer Graphics Forum* 38.3 (2019), pp. 687–698. ISSN: 1467-8659. DOI: [10.1111/cgf.13720](https://doi.org/10.1111/cgf.13720).
- [62] Harris Cooper and Larry V. Hedges. *The Handbook of Research Synthesis*. Russell Sage Foundation, Nov. 1993. ISBN: 978-1-61044-137-7.
- [63] Patricia Costigan-Eaves. "Edward R. Tufte *The Visual Display of Quantitative Information*." In: *Information Design Journal* 4.3 (Jan. 1986), pp. 235–236. ISSN: 0142-5471, 1569-979X. DOI: [10.1075/idj.4.3.12cos](https://doi.org/10.1075/idj.4.3.12cos). (Visited on 07/27/2022).
- [64] Walt Crawford. "Paper Persists: Why Physical Library Collections Still Matter." In: *Online* 22.1 (Jan/Feb 1998), pp. 42–48. ISSN: 01465422.
- [65] Anthony Creed, Ian Dennis, and Stephen Newstead. "Proof-Reading on VDUs." In: *Behaviour & Information Technology* 6.1 (Jan. 1987), pp. 3–13. ISSN: 0144-929X. DOI: [10.1080/01449298708901814](https://doi.org/10.1080/01449298708901814).
- [66] Mihaly Csikszentmihalyi. *Flow: The Psychology of Optimal Experience*. First Edition. New York: Harper Perennial Modern Classics, July 2008. ISBN: 978-0-06-133920-2.
- [67] Weiwei Cui, Xiaoyu Zhang, Yun Wang, He Huang, Bei Chen, Lei Fang, Haidong Zhang, Jian-Guan Lou, and Dongmei Zhang. "Text-to-Viz: Automatic Generation of Infographics from Proportion-Related Natural Language Statements." In: *IEEE Transactions on Visualization and Computer Graphics* 26.1 (Jan. 2020), pp. 906–916. ISSN: 1077-2626, 1941-0506, 2160-9306. DOI: [10.1109/TVCG.2019.2934785](https://doi.org/10.1109/TVCG.2019.2934785).
- [68] Zhe Cui, Sriram Karthik Badam, M Adil Yalçın, and Niklas Elmqvist. "DataSite: Proactive Visual Data Exploration with Computation of Insight-Based Recommendations." In: *Information Visualization* 18.2 (Apr. 2019), pp. 251–267. ISSN: 1473-8716. DOI: [10.1177/1473871618806555](https://doi.org/10.1177/1473871618806555).
- [69] Hugh M. Culbertson and Richard D. Powers. "A Study of Graph Comprehension Difficulties." In: *Audio Visual Communication Review* 7.2 (1959), pp. 97–110. ISSN: 0885-727X.
- [70] Geoff Cumming. "The New Statistics: Why and How." In: *Psychological Science* 25.1 (Jan. 2014), pp. 7–29. ISSN: 0956-7976. DOI: [10.1177/0956797613504966](https://doi.org/10.1177/0956797613504966). (Visited on 01/12/2023).
- [71] Geoff Cumming and Robert Calin-Jageman. *Introduction to the New Statistics: Estimation, Open Science, and Beyond*. 1st edition. London : New York: Routledge, Oct. 2016. ISBN: 978-1-138-82552-9.
- [72] Geoff Cumming, Fiona Fidler, and David L. Vaux. "Error Bars in Experimental Biology." In: *Journal of Cell Biology* 177.1 (Apr. 2007), pp. 7–11. ISSN: 0021-9525. DOI: [10.1083/jcb.200611141](https://doi.org/10.1083/jcb.200611141). (Visited on 01/12/2023).

- [73] Geoff Cumming and Sue Finch. "A Primer on the Understanding, Use, and Calculation of Confidence Intervals That Are Based on Central and Noncentral Distributions." In: *Educational and Psychological Measurement* 61.4 (Aug. 2001), pp. 532–574. ISSN: 0013-1644, 1552-3888. DOI: [10.1177/0013164401614002](https://doi.org/10.1177/0013164401614002). (Visited on 01/14/2023).
- [74] Geoff Cumming, Mark Zangari, and Neil Thomason. "Designing Software for Cognitive Change: StatPlay and Understanding Statistics." In: *World Conference on Computers in Education VI: WCCE '95 Liberating the Learner, Proceedings of the Sixth IFIP World Conference on Computers in Education, 1995*. Ed. by J. David Tinsley and Tom J. van Weert. IFIP — The International Federation for Information Processing. Boston, MA: Springer US, 1995, pp. 753–765. ISBN: 978-0-387-34844-5. DOI: [10.1007/978-0-387-34844-5\\_71](https://doi.org/10.1007/978-0-387-34844-5_71). (Visited on 04/04/2023).
- [75] William H. Cushman. "Reading from Microfiche, a VDT, and the Printed Page: Subjective Fatigue and Performance." In: *Human Factors* 28.1 (Feb. 1986), pp. 63–73. ISSN: 0018-7208. DOI: [10.1177/001872088602800107](https://doi.org/10.1177/001872088602800107).
- [76] Allen Cypher, Mira Dontcheva, Tessa Lau, and Jeffrey Nichols. *No Code Required: Giving Users Tools to Transform the Web*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2010. ISBN: 012381541X, 9780123815415.
- [77] Richard L. Daft and Robert H. Lengel. "Organizational Information Requirements, Media Richness and Structural Design." In: *Management Science* 32.5 (1986), pp. 554–571. ISSN: 0025-1909. JSTOR: [2631846](https://www.jstor.org/stable/2631846). (Visited on 06/20/2023).
- [78] Kenny Davila, Bhargava Urala Kota, Srirangaraj Setlur, Venu Govindaraju, Christopher Tensmeyer, Sumit Shekhar, and Ritwick Chaudhry. "ICDAR 2019 Competition on Harvesting Raw Tables from Infographics (CHART-Infographics)." In: *2019 International Conference on Document Analysis and Recognition (ICDAR)*. Sydney, NSW, Australia: IEEE, Sept. 2019, pp. 1594–1599. DOI: [10.1109/ICDAR.2019.00203](https://doi.org/10.1109/ICDAR.2019.00203).
- [79] Kenny Davila, Srirangaraj Setlur, David Doermann, Bhargava Urala Kota, and Venu Govindaraju. "Chart Mining: A Survey of Methods for Automated Chart Analysis." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.11 (Nov. 2021), pp. 3799–3819. ISSN: 1939-3539. DOI: [10.1109/TPAMI.2020.2992028](https://doi.org/10.1109/TPAMI.2020.2992028).
- [80] Kenny Davila, Chris Tensmeyer, Sumit Shekhar, Hrituraj Singh, Srirangaraj Setlur, and Venu Govindaraju. "ICPR 2020 - Competition on Harvesting Raw Tables from Infographics." In: *Pattern Recognition. ICPR International Workshops and Challenges*. Ed. by Alberto Del Bimbo, Rita Cucchiara, Stan Sclaroff, Giovanni Maria Farinella, Tao Mei, Marco Bertini, Hugo Jair Escalante, and Roberto Vezzani. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2021, pp. 361–380. ISBN: 978-3-030-68793-9. DOI: [10.1007/978-3-030-68793-9\\_27](https://doi.org/10.1007/978-3-030-68793-9_27).

- [81] Markus Demleitner. *Dexter*. 2010.
- [82] Thomas J. DiCiccio and Bradley Efron. “Bootstrap Confidence Intervals.” In: *Statistical Science* 11.3 (Sept. 1996), pp. 189–228. ISSN: 0883-4237, 2168-8745. DOI: [10 . 1214 / ss / 1032280214](https://doi.org/10.1214/ss/1032280214). (Visited on 04/05/2023).
- [83] Andreas Dieberger, Cameron Miner, and Dulce Ponceleon. “Supporting Narrative Flow in Presentation Software.” In: *CHI '01 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '01. Seattle, Washington: ACM, 2001, pp. 137–138. ISBN: 1-58113-340-5. DOI: [10.1145/634067.634151](https://doi.org/10.1145/634067.634151). URL: <http://doi.acm.org/10.1145/634067.634151>.
- [84] Andrew Dillon. “Reading from Paper versus Screens: A Critical Review of the Empirical Literature.” In: *Ergonomics* 35.10 (Oct. 1992), pp. 1297–1326. ISSN: 0014-0139, 1366-5847. DOI: [10 . 1080 / 00140139208967394](https://doi.org/10.1080/00140139208967394).
- [85] *Distill — Latest Articles about Machine Learning*. <http://distill.pub/>. 2016.
- [86] Morgan Dixon and James Fogarty. “Prefab: Implementing Advanced Behaviors Using Pixel-based Reverse Engineering of Interface Structure.” In: *Proceedings of the 2010 CHI Conference on Human Factors in Computing Systems*. CHI '10. Atlanta, Georgia, USA: ACM, 2010, pp. 1525–1534. ISBN: 978-1-60558-929-9. DOI: [10 . 1145 / 1753326 . 1753554](https://doi.org/10.1145/1753326.1753554). URL: <http://doi.acm.org.buproxy.univ-lille1.fr/10.1145/1753326.1753554>.
- [87] Longxu Dou, Guanghui Qin, Jinpeng Wang, Jin-Ge Yao, and Chin-Yew Lin. “Data2Text Studio: Automated Text Generation from Structured Data.” In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 13–18. DOI: [10.18653/v1/D18-2003](https://doi.org/10.18653/v1/D18-2003). (Visited on 06/15/2022).
- [88] Pierre Dragicevic. “Fair Statistical Communication in HCI.” In: *Modern Statistical Methods for HCI*. Ed. by Judy Robertson and Maurits Kaptein. Cham: Springer International Publishing, 2016, pp. 291–330. ISBN: 978-3-319-26631-2 978-3-319-26633-6. DOI: [10 . 1007 / 978 - 3 - 319 - 26633 - 6\\_13](https://doi.org/10.1007/978-3-319-26633-6_13). (Visited on 07/28/2022).
- [89] Pierre Dragicevic and Yvonne Jansen. “Blinded with Science or Informed by Charts? A Replication Study.” In: *IEEE Transactions on Visualization and Computer Graphics* 24.1 (Jan. 2018), pp. 781–790. ISSN: 1941-0506. DOI: [10.1109/TVCG.2017.2744298](https://doi.org/10.1109/TVCG.2017.2744298).
- [90] Pierre Dragicevic, Yvonne Jansen, Abhraneel Sarma, Matthew Kay, and Fanny Chevalier. *Explorable Multiverse Analyses*. 2018. URL: <https://explorablemultiverse.github.io/>.

## BIBLIOGRAPHY

- [91] Pierre Dragicevic, Yvonne Jansen, Abhraneel Sarma, Matthew Kay, and Fanny Chevalier. "Increasing the Transparency of Research Papers with Explorable Multiverse Analyses." In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. Glasgow, Scotland Uk: Association for Computing Machinery, May 2019, pp. 1–15. ISBN: 978-1-4503-5970-2. DOI: [10.1145/3290605.3300295](https://doi.org/10.1145/3290605.3300295). URL: <https://doi.org/10.1145/3290605.3300295> (visited on 06/15/2022).
- [92] Nell K. Duke and P. David Pearson. "Effective Practices for Developing Reading Comprehension." In: *Journal of Education* 189.1-2 (Jan. 2009), pp. 107–122. ISSN: 0022-0574. DOI: [10.1177/0022057409189001-208](https://doi.org/10.1177/0022057409189001-208).
- [93] Karolina Duklan and Michael A Martin. "Communicating Effectively with Words, Numbers, and Pictures: Drawing on Experience." In: *Journal of Actuarial Practice* 10 (2002), p. 59.
- [94] William P Dunlap. "Generalizing the Common Language Effect Size Indicator to Bivariate Normal Correlations." In: ().
- [95] Olive Jean Dunn. "Multiple Comparisons among Means." In: *Journal of the American Statistical Association* 56.293 (Mar. 1961), pp. 52–64. ISSN: 0162-1459, 1537-274X. DOI: [10.1080/01621459.1961.10482090](https://doi.org/10.1080/01621459.1961.10482090). (Visited on 04/03/2023).
- [96] Bradley Efron. *The Jackknife, the Bootstrap and Other Resampling Plans*. CBMS-NSF Regional Conference Series in Applied Mathematics. Society for Industrial and Applied Mathematics, Jan. 1982. ISBN: 978-0-89871-179-0. DOI: [10.1137/1.9781611970319](https://doi.org/10.1137/1.9781611970319). (Visited on 04/03/2023).
- [97] Bradley Efron. "Better Bootstrap Confidence Intervals." In: *Journal of the American Statistical Association* 82.397 (Mar. 1987), pp. 171–185. ISSN: 0162-1459. DOI: [10.1080/01621459.1987.10478410](https://doi.org/10.1080/01621459.1987.10478410). (Visited on 04/03/2023).
- [98] Bradley Efron. "The Bootstrap and Modern Statistics." In: *Journal of the American Statistical Association* 95.452 (2000), pp. 1293–1296. ISSN: 0162-1459. DOI: [10.2307/2669773](https://doi.org/10.2307/2669773). JSTOR: [2669773](https://www.jstor.org/stable/2669773). (Visited on 04/05/2023).
- [99] Lisa A. Elkin, Matthew Kay, James J. Higgins, and Jacob O. Wobbrock. "An Aligned Rank Transform Procedure for Multifactor Contrast Tests." In: *The 34th Annual ACM Symposium on User Interface Software and Technology*. UIST '21. New York, NY, USA: Association for Computing Machinery, Oct. 2021, pp. 754–768. ISBN: 978-1-4503-8635-7. DOI: [10.1145/3472749.3474784](https://doi.org/10.1145/3472749.3474784). (Visited on 03/23/2023).
- [100] Lei Fang and Jian-Guang Lou. *Part & Whole Extraction: Towards A Deep Understanding of Quantitative Facts for Percentages in Text*. Oct. 2021. arXiv: [2110.13505](https://arxiv.org/abs/2110.13505) [cs].
- [101] Gloria D. Feliciano, Richard D. Powers, and Bryant E. Kearl. "The Presentation of Statistical Information." In: *Audio Visual Communication Review* 11.3 (1963), pp. 32–39. ISSN: 0885-727X.
- [102] Fewer and Faster. *Speaker Deck*. 2010. URL: <https://speakerdeck.com/>.

- [103] Fiona Fidler, Geoff Cumming, Neil Thomason, Dominique Pannuzzo, Julian Smith, Penny Fyffe, Holly Edmonds, Claire Harrington, and Rachel Schmitt. "Toward Improved Statistical Reporting in the Journal of Consulting and Clinical Psychology." In: *Journal of Consulting and Clinical Psychology* 73.1 (Feb. 2005), pp. 136–143. ISSN: 0022-006X. DOI: [10.1037/0022-006X.73.1.136](https://doi.org/10.1037/0022-006X.73.1.136).
- [104] Fiona Fidler and Geoffrey R. Loftus. "Why Figures with Error Bars Should Replace p Values: Some Conceptual Arguments and Empirical Demonstrations." In: *Zeitschrift für Psychologie/Journal of Psychology* 217.1 (2009), pp. 27–37. ISSN: 0044-3409. DOI: [10.1027/0044-3409.217.1.27](https://doi.org/10.1027/0044-3409.217.1.27).
- [105] Hans Fischer. *A History of the Central Limit Theorem: From Classical to Modern Probability Theory*. New York, NY: Springer New York, 2011. ISBN: 978-0-387-87856-0 978-0-387-87857-7. DOI: [10.1007/978-0-387-87857-7](https://doi.org/10.1007/978-0-387-87857-7). (Visited on 04/05/2023).
- [106] Martin A. Fischler and Robert C. Bolles. "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography." In: *Readings in Computer Vision*. Ed. by Martin Fischler and Oscar Firschein. San Francisco (CA): Morgan Kaufmann, 1987, pp. 726–740. ISBN: 978-0-08-051581-6. DOI: [10.1016/B978-0-08-051581-6.50070-2](https://doi.org/10.1016/B978-0-08-051581-6.50070-2). URL: <https://www.sciencedirect.com/science/article/pii/B9780080515816500702>.
- [107] David R. Flatla, Alan R. Andrade, Ross D. Teviotdale, Dylan L. Knowles, and Craig Stewart. "ColourID: Improving Colour Identification for People with Impaired Colour Vision." In: *Proceedings of the 2015 CHI Conference on Human Factors in Computing Systems*. CHI '15. New York, NY, USA: Association for Computing Machinery, Apr. 2015, pp. 3543–3552. ISBN: 978-1-4503-3145-6. DOI: [10.1145/2702123.2702578](https://doi.org/10.1145/2702123.2702578). (Visited on 05/31/2022).
- [108] James Fogarty. "Code and Contribution in Interactive Systems Research." In: ().
- [109] Raymond Fok, Andrew Head, Jonathan Bragg, Kyle Lo, Marti A. Hearst, and Daniel S. Weld. *Scim: Intelligent Faceted Highlights for Interactive, Multi-Pass Skimming of Scientific Papers*. May 2022. arXiv: [2205.04561 \[cs\]](https://arxiv.org/abs/2205.04561).
- [110] P. E. Forssen. "Maximally Stable Colour Regions for Recognition and Matching." In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. USA: IEEE, June 2007, pp. 1–8. DOI: [10.1109/CVPR.2007.383120](https://doi.org/10.1109/CVPR.2007.383120).
- [111] Erin D. Foster and Ariel Deardorff. "Open Science Framework (OSF)." In: *Journal of the Medical Library Association : JMLA* 105.2 (Apr. 2017), pp. 203–206. ISSN: 1536-5050. DOI: [10.5195/jmla.2017.88](https://doi.org/10.5195/jmla.2017.88).
- [112] Herbert Freeman. "On the Encoding of Arbitrary Geometric Configurations." In: *IRE Transactions on Electronic Computers* EC-10.2 (1961), pp. 260–268. DOI: [10.1109/TEC.1961.5219197](https://doi.org/10.1109/TEC.1961.5219197).



- [113] Jérémy Frey, Maxime Daniel, Julien Castet, Martin Hachet, and Fabien Lotte. “Framework for Electroencephalography-based Evaluation of User Experience.” In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. CHI ’16. New York, NY, USA: Association for Computing Machinery, May 2016, pp. 2283–2294. ISBN: 978-1-4503-3362-7. DOI: [10.1145/2858036.2858525](https://doi.org/10.1145/2858036.2858525). (Visited on 06/17/2022).
- [114] Jun Fujima, Aran Lunzer, Kasper Hornbæk, and Yuzuru Tanaka. “Clip, Connect, Clone: Combining Application Elements to Build Custom Interfaces for Information Access.” In: *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*. UIST ’04. Santa Fe, NM, USA: ACM, 2004, pp. 175–184. ISBN: 1-58113-957-8. DOI: [10.1145/1029632.1029664](https://doi.org/10.1145/1029632.1029664). URL: <http://doi.acm.org/10.1145/1029632.1029664>.
- [115] GNOME. *Assistive Technology Service Provider Interface*. 2011. URL: <https://github.com/GNOME/at-spi2-atk>.
- [116] Jinglun Gao, Yin Zhou, and Kenneth E. Barner. “View: Visual Information Extraction Widget for Improving Chart Images Accessibility.” In: *2012 19th IEEE International Conference on Image Processing*. Orlando, FL, USA: IEEE, Sept. 2012, pp. 2865–2868. DOI: [10.1109/ICIP.2012.6467497](https://doi.org/10.1109/ICIP.2012.6467497).
- [117] Tong Gao, Mira Dontcheva, Eytan Adar, Zhicheng Liu, and Karrie G. Karahalios. “DataTone: Managing Ambiguity in Natural Language Interfaces for Data Visualization.” In: *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. Charlotte NC USA: ACM, Nov. 2015, pp. 489–500. ISBN: 978-1-4503-3779-3. DOI: [10.1145/2807442.2807478](https://doi.org/10.1145/2807442.2807478).
- [118] Tong Gao, Jessica R. Hullman, Eytan Adar, Brent Hecht, and Nicholas Diakopoulos. “NewsViews: An Automated Pipeline for Creating Custom Geovisualizations for News.” In: *Proceedings of the 2014 CHI Conference on Human Factors in Computing Systems*. CHI ’14. New York, NY, USA: Association for Computing Machinery, Apr. 2014, pp. 3005–3014. ISBN: 978-1-4503-2473-1. DOI: [10.1145/2556288.2557228](https://doi.org/10.1145/2556288.2557228). (Visited on 07/07/2022).
- [119] Tong Ge, Yue Zhao, Bongshin Lee, Donghao Ren, Baoquan Chen, and Yunhai Wang. “Canis: A High-Level Language for Data-Driven Chart Animations.” In: *Computer Graphics Forum* 39.3 (June 2020), pp. 607–617. ISSN: 0167-7055, 1467-8659. DOI: [10.1111/cgf.14005](https://doi.org/10.1111/cgf.14005). (Visited on 01/23/2023).
- [120] Andrew Gelman, Cristian Pasarica, and Rahul Dodhia. “Let’s Practice What We Preach.” In: *The American Statistician* 56.2 (May 2002), pp. 121–130. ISSN: 0003-1305. DOI: [10.1198/000313002317572790](https://doi.org/10.1198/000313002317572790). (Visited on 08/04/2022).

- [121] Emmanouil Giannidakis, Gilles Bailly, Sylvain Malacria, and Fanny Chevalier. "IconHK: Using Toolbar Button Icons to Communicate Keyboard Shortcuts." In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI '17. Denver, Colorado, USA: ACM, May 2017, pp. 4715–4726. ISBN: 978-1-4503-4655-9. DOI: [10.1145/3025453.3025595](https://doi.org/10.1145/3025453.3025595). URL: <http://doi.acm.org/10.1145/3025453.3025595>.
- [122] Douglas J. Gillan, Christopher D. Wickens, J. G. Hollands, and C. Melody Carswell. "Guidelines for Presenting Quantitative Data in HFES Publications." In: *Human Factors: The Journal of the Human Factors and Ergonomics Society* 40.1 (Mar. 1998), pp. 28–41. ISSN: 0018-7208, 1547-8181. DOI: [10.1518/001872098779480640](https://doi.org/10.1518/001872098779480640).
- [123] David Giofrè, Ingrid Boedker, Geoff Cumming, Carlotta Rivella, and Patrizio Tressoldi. "The Influence of Journal Submission Guidelines on Authors' Reporting of Statistics and Use of Open Research Practices: Five Years Later." In: *Behavior Research Methods* (Oct. 2022). ISSN: 1554-3528. DOI: [10.3758/s13428-022-01993-3](https://doi.org/10.3758/s13428-022-01993-3). (Visited on 03/31/2023).
- [124] Arthur M Glenberg and William E Langston. "Comprehension of Illustrated Text: Pictures Help to Build Mental Models." In: *Journal of Memory and Language* 31.2 (Apr. 1992), pp. 129–151. ISSN: 0749-596X. DOI: [10.1016/0749-596X\(92\)90008-L](https://doi.org/10.1016/0749-596X(92)90008-L).
- [125] Pascal Goffin, Tanja Blascheck, Petra Isenberg, and Wesley Willett. "Interaction Techniques for Visual Exploration Using Embedded Word-Scale Visualizations." In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. Honolulu HI USA: ACM, Apr. 2020, pp. 1–13. ISBN: 978-1-4503-6708-0. DOI: [10.1145/3313831.3376842](https://doi.org/10.1145/3313831.3376842).
- [126] Pascal Goffin, Jeremy Boy, Wesley Willett, and Petra Isenberg. "An Exploratory Study of Word-Scale Graphics in Data-Rich Text Documents." In: *IEEE Transactions on Visualization and Computer Graphics* 23.10 (Oct. 2017), pp. 2275–2287. ISSN: 1941-0506. DOI: [10.1109/TVCG.2016.2618797](https://doi.org/10.1109/TVCG.2016.2618797).
- [127] Pascal Goffin, Wesley Willett, Anastasia Bezerianos, and Petra Isenberg. "Exploring the Effect of Word-Scale Visualizations on Reading Behavior." In: *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*. CHI EA '15. New York, NY, USA: Association for Computing Machinery, Apr. 2015, pp. 1827–1832. ISBN: 978-1-4503-3146-3. DOI: [10.1145/2702613.2732778](https://doi.org/10.1145/2702613.2732778).
- [128] Pascal Goffin, Wesley Willett, Jean-Daniel Fekete, and Petra Isenberg. "Exploring the Placement and Design of Word-Scale Visualizations." In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (Dec. 2014), pp. 2291–2300. ISSN: 1941-0506. DOI: [10.1109/TVCG.2014.2346435](https://doi.org/10.1109/TVCG.2014.2346435).

- [129] Pascal Goffin, Wesley Willett, Jean-Daniel Fekete, and Petra Isenberg. “Design Considerations for Enhancing Word-Scale Visualizations with Interaction.” In: *Posters of the Conference on Information Visualization (InfoVis)*. Chicago, Illinois, USA: IEEE, Oct. 2015, p. 2.
- [130] Pascal Goffin, Wesley Willett, and Petra Isenberg. “Sharing Information from Personal Digital Notes Using Word-Scale Visualizations.” In: *Proceedings of the IEEE VIS Workshop on Personal Visualization: Exploring Data in Everyday Life*. Chicago Illinois USA: IEEE, Oct. 2015, p. 5.
- [131] Alyssa Goodman et al. *The “Paper” of the Future*. Preprint. Preprints, Jan. 2021. DOI: [10.22541/au.148769949.92783646/v2](https://doi.org/10.22541/au.148769949.92783646/v2).
- [132] Steven Goodman. “A Dirty Dozen: Twelve P-Value Misconceptions.” In: *Seminars in Hematology* 45.3 (July 2008), pp. 135–140. ISSN: 00371963. DOI: [10.1053/j.seminhematol.2008.04.003](https://doi.org/10.1053/j.seminhematol.2008.04.003). (Visited on 03/22/2023).
- [133] Rupert Goodwins. *Firefox Points the Way to Eradicating PDFs*. [https://www.theregister.com/2022/10/24/firefox\\_pdf\\_column/](https://www.theregister.com/2022/10/24/firefox_pdf_column/).
- [134] Mitchell Gordon, Tom Ouyang, and Shumin Zhai. “WatchWriter: Tap and Gesture Typing on a Smartwatch Miniature Keyboard with Statistical Decoding.” In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. San Jose California USA: ACM, May 2016, pp. 3817–3821. ISBN: 978-1-4503-3362-7. DOI: [10.1145/2858036.2858242](https://doi.org/10.1145/2858036.2858242). (Visited on 04/04/2023).
- [135] John D. Gould and Nancy Grischkowsky. “Doing the Same Work with Hard Copy and with Cathode-Ray Tube (CRT) Computer Terminals.” In: *Human Factors* 26.3 (June 1984), pp. 323–337. ISSN: 0018-7208. DOI: [10.1177/001872088402600308](https://doi.org/10.1177/001872088402600308).
- [136] John Gramlich. *What the 2020 Electorate Looks like by Party, Race and Ethnicity, Age, Education and Religion*. 2020.
- [137] Arnd Gross, Sibylle Schirm, and Markus Scholz. “Ycasd— a Tool for Capturing and Scaling Data from Graphical Representations.” In: *BMC Bioinformatics* 15.1 (June 2014), p. 219. ISSN: 1471-2105. DOI: [10.1186/1471-2105-15-219](https://doi.org/10.1186/1471-2105-15-219). (Visited on 05/26/2022).
- [138] Tovi Grossman and Ravin Balakrishnan. “The Bubble Cursor: Enhancing Target Acquisition by Dynamic Resizing of the Cursor’s Activation Area.” In: *Proceedings of the 2005 CHI Conference on Human Factors in Computing Systems*. CHI ’05. Portland, Oregon, USA: ACM, 2005, pp. 281–290. ISBN: 1-58113-998-5. DOI: [10.1145/1054972.1055012](https://doi.org/10.1145/1054972.1055012). URL: <http://doi.acm.org/10.1145/1054972.1055012>.
- [139] Tovi Grossman, Fanny Chevalier, and Rubaiat Habib Kazi. “Your Paper Is Dead! Bringing Life to Research Articles with Animated Figures.” In: *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*. CHI EA ’15. Seoul, Republic of Korea: Association for Computing Machinery, Apr. 2015, pp. 461–475. ISBN: 978-1-4503-3146-3. DOI: [10.1145/2702613.2732501](https://doi.org/10.1145/2702613.2732501). URL: <http://doi.acm.org.buproxy.univ-lille1.fr/10.1145/2702613.2732501>.

- [140] Tovi Grossman and George Fitzmaurice. “ToolClips: An Investigation of Contextual Video Assistance for Functionality Understanding.” In: *Proceedings of the 2010 CHI Conference on Human Factors in Computing Systems*. CHI '10. Atlanta, Georgia, USA: Association for Computing Machinery, 2010, 1515–1524. ISBN: 9781605589299. DOI: [10.1145/1753326.1753552](https://doi.org/10.1145/1753326.1753552). URL: <https://doi.org/10.1145/1753326.1753552>.
- [141] Tovi Grossman, Nicholas Kong, and Ravin Balakrishnan. “Modeling Pointing at Targets of Arbitrary Shapes.” In: *Proceedings of the 2007 CHI Conference on Human Factors in Computing Systems*. San Jose California USA: ACM, Apr. 2007, pp. 463–472. ISBN: 978-1-59593-593-9. DOI: [10.1145/1240624.1240700](https://doi.org/10.1145/1240624.1240700). (Visited on 06/13/2022).
- [142] Transparent Statistics in Human-Computer Interaction Working Group. *Transparent Statistics Guidelines*. June 2019. DOI: [10.5281/zenodo.1186169](https://doi.org/10.5281/zenodo.1186169).
- [143] Ken Gu, Eunice Jun, and Tim Althoff. *Understanding and Supporting Debugging Workflows in Multiverse Analysis*. Feb. 2023. arXiv: [arXiv: 2210.03804](https://arxiv.org/abs/2210.03804). (Visited on 04/05/2023).
- [144] François Guimbretière. “Paper Augmented Digital Documents.” In: *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology*. UIST '03. Vancouver, Canada: ACM, 2003, pp. 51–60. ISBN: 1-58113-636-6. DOI: [10.1145/964696.964702](https://doi.org/10.1145/964696.964702). URL: <http://doi.acm.org/10.1145/964696.964702>.
- [145] Brian D. Hall, Yang Liu, Yvonne Jansen, Pierre Dragicevic, Fanny Chevalier, and Matthew Kay. “A Survey of Tasks and Visualizations in Multiverse Analysis Reports.” In: *Computer Graphics Forum* 41.1 (2022), pp. 402–426. ISSN: 1467-8659. DOI: [10.1111/cgf.14443](https://doi.org/10.1111/cgf.14443). (Visited on 02/19/2023).
- [146] Lisa Lavoie Harlow, Stanley A. Mulaik, and James H. Steiger, eds. *What If There Were No Significance Tests? What If There Were No Significance Tests?* Mahwah, NJ, US: Lawrence Erlbaum Associates Publishers, 1997, pp. xviii, 446. ISBN: 978-0-8058-2634-0.
- [147] Jonathan Harper and Maneesh Agrawala. “Deconstructing and Restyling D3 Visualizations.” In: *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*. UIST '14. New York, NY, USA: Association for Computing Machinery, Oct. 2014, pp. 253–262. ISBN: 978-1-4503-3069-5. DOI: [10.1145/2642918.2647411](https://doi.org/10.1145/2642918.2647411). (Visited on 05/26/2022).
- [148] Jonathan Harper and Maneesh Agrawala. “Converting Basic D3 Charts into Reusable Style Templates.” In: *IEEE Transactions on Visualization and Computer Graphics* 24.3 (Mar. 2018), pp. 1274–1286. ISSN: 1941-0506. DOI: [10.1109/TVCG.2017.2659744](https://doi.org/10.1109/TVCG.2017.2659744).
- [149] Charles R. Harris et al. “Array Programming with NumPy.” In: *Nature* 585:7825 (Sept. 2020), pp. 357–362. ISSN: 1476-4687. DOI: [10.1038/s41586-020-2649-2](https://doi.org/10.1038/s41586-020-2649-2). (Visited on 04/03/2023).

- [150] Chris Harris and Mike Stephens. “A combined corner and edge detector.” In: *In Proc. of Fourth Alvey Vision Conference*. USA: IEEE, 1988, pp. 147–151.
- [151] Björn Hartmann, Leslie Wu, Kevin Collins, and Scott R. Klemmer. “Programming by a Sample: Rapidly Creating Web Applications with D.Mix.” In: *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*. UIST ’07. Newport, Rhode Island, USA: ACM, 2007, pp. 241–250. ISBN: 978-1-59593-679-0. DOI: [10.1145/1294211.1294254](https://doi.org/10.1145/1294211.1294254). URL: <http://doi.acm.org/10.1145/1294211.1294254>.
- [152] S I Hayakawa. *Language In Thought And Action*. 1949.
- [153] Andrew Head, Kyle Lo, Dongyeop Kang, Raymond Fok, Sam Skjonsberg, Daniel S. Weld, and Marti A. Hearst. “Augmenting Scientific Papers with Just-in-Time, Position-Sensitive Definitions of Terms and Symbols.” In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. Yokohama Japan: ACM, May 2021, pp. 1–18. ISBN: 978-1-4503-8096-6. DOI: [10.1145/3411764.3445648](https://doi.org/10.1145/3411764.3445648).
- [154] Andrew Head, Amber Xie, and Marti A. Hearst. “Math Augmentation: How Authors Enhance the Readability of Formulas Using Novel Visual Design Practices.” In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. New Orleans LA USA: ACM, Apr. 2022, pp. 1–18. ISBN: 978-1-4503-9157-3. DOI: [10.1145/3491102.3501932](https://doi.org/10.1145/3491102.3501932).
- [155] Jeffrey Heer and George Robertson. “Animated Transitions in Statistical Data Graphics.” In: *IEEE Transactions on Visualization and Computer Graphics* 13.6 (Nov. 2007), pp. 1240–1247. ISSN: 1941-0506. DOI: [10.1109/TVCG.2007.70539](https://doi.org/10.1109/TVCG.2007.70539).
- [156] Jouni Helske, Satu Helske, Matthew Cooper, Anders Ynnerman, and Lonni Besançon. “Can Visualization Alleviate Dichotomous Thinking? Effects of Visual Representations on the Cliff Effect.” In: *IEEE Transactions on Visualization and Computer Graphics* 27.8 (Aug. 2021), pp. 3397–3409. ISSN: 1941-0506. DOI: [10.1109/TVCG.2021.3073466](https://doi.org/10.1109/TVCG.2021.3073466). arXiv: [2002.07671 \[cs, stat\]](https://arxiv.org/abs/2002.07671). (Visited on 03/23/2023).
- [157] Brian Hempel, Justin Lubin, and Ravi Chugh. “Sketch-n-Sketch: Output-Directed Programming for SVG.” In: *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. Oct. 2019, pp. 281–292. DOI: [10.1145/3332165.3347925](https://doi.org/10.1145/3332165.3347925). arXiv: [1907.10699 \[cs\]](https://arxiv.org/abs/1907.10699). (Visited on 06/25/2023).
- [158] R.N. Henson. “Analysis of Variance (ANOVA).” In: *Brain Mapping*. Elsevier, 2015, pp. 477–481. ISBN: 978-0-12-397316-0. DOI: [10.1016/B978-0-12-397025-1.00319-5](https://doi.org/10.1016/B978-0-12-397025-1.00319-5). (Visited on 04/04/2023).

- [159] Michael H. Herzog, Gregory Francis, and Aaron Clarke. “ANOVA.” In: *Understanding Statistics and Experimental Design : How to Not Lie with Statistics*. Ed. by Michael H. Herzog, Gregory Francis, and Aaron Clarke. Learning Materials in Biosciences. Cham: Springer International Publishing, 2019, pp. 67–82. ISBN: 978-3-030-03499-3. DOI: [10.1007/978-3-030-03499-3\\_6](https://doi.org/10.1007/978-3-030-03499-3_6). (Visited on 04/04/2023).
- [160] Ken Hinckley, Shengdong Zhao, Raman Sarin, Patrick Baudisch, Edward Cutrell, Michael Shilman, and Desney Tan. “InkSeine: In Situ Search for Active Note Taking.” In: *Proceedings of the 2007 CHI Conference on Human Factors in Computing Systems*. CHI '07. New York, NY, USA: Association for Computing Machinery, Apr. 2007, pp. 251–260. ISBN: 978-1-59593-593-9. DOI: [10.1145/1240624.1240666](https://doi.org/10.1145/1240624.1240666).
- [161] William E. Hockley. “The Picture Superiority Effect in Associative Recognition.” In: *Memory & Cognition* 36.7 (Oct. 2008), pp. 1351–1359. ISSN: 1532-5946. DOI: [10.3758/MC.36.7.1351](https://doi.org/10.3758/MC.36.7.1351).
- [162] Jake M. Hofman, Daniel G. Goldstein, and Jessica Hullman. “How Visualizing Inferential Uncertainty Can Mislead Readers About Treatment Effects in Scientific Results.” In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. New York, NY, USA: Association for Computing Machinery, Apr. 2020, pp. 1–12. ISBN: 978-1-4503-6708-0. DOI: [10.1145/3313831.3376454](https://doi.org/10.1145/3313831.3376454). (Visited on 01/11/2023).
- [163] Fred Hohman, Matthew Conlen, Jeffrey Heer, and Duen Horng (Polo) Chau. “Communicating with Interactive Articles.” In: *Distill* 5.9 (Sept. 2020), e28. ISSN: 2476-0757. DOI: [10.23915/distill.00028](https://doi.org/10.23915/distill.00028).
- [164] Sture Holm. “A Simple Sequentially Rejective Multiple Test Procedure.” In: *Scandinavian Journal of Statistics* 6.2 (1979), pp. 65–70. ISSN: 0303-6898. JSTOR: [4615733](https://www.jstor.org/stable/4615733). (Visited on 04/03/2023).
- [165] Jonggi Hong, Seongkook Heo, Poika Isokoski, and Geehyuk Lee. “SplitBoard: A Simple Split Soft Keyboard for Wristwatch-sized Touch Screens.” In: *Proceedings of the 2015 CHI Conference on Human Factors in Computing Systems*. Seoul Republic of Korea: ACM, Apr. 2015, pp. 1233–1236. ISBN: 978-1-4503-3145-6. DOI: [10.1145/2702123.2702273](https://doi.org/10.1145/2702123.2702273). (Visited on 04/04/2023).
- [166] Enamul Hoque and Maneesh Agrawala. “Searching the Visual Style and Structure of D3 Visualizations.” In: *IEEE Transactions on Visualization and Computer Graphics* 26.1 (Jan. 2020), pp. 1236–1245. ISSN: 1941-0506. DOI: [10.1109/TVCG.2019.2934431](https://doi.org/10.1109/TVCG.2019.2934431).
- [167] Kasper Hornb and Erik Frøkj. “Reading Patterns and Usability in Visualizations of Electronic Documents.” In: 10.2 (2003), p. 31.
- [168] Eric Horvitz. “Principles of Mixed-Initiative User Interfaces.” In: *Proceedings of the 1999 CHI Conference on Human Factors in Computing Systems*. Pittsburgh, Pennsylvania, United States: ACM Press, 1999, pp. 159–166. ISBN: 978-0-201-48559-2. DOI: [10.1145/302979.303030](https://doi.org/10.1145/302979.303030). (Visited on 06/20/2022).

- [169] Kevin Hu, Michiel A. Bakker, Stephen Li, Tim Kraska, and César Hidalgo. “VizML: A Machine Learning Approach to Visualization Recommendation.” In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. Glasgow Scotland Uk: ACM, May 2019, pp. 1–12. ISBN: 978-1-4503-5970-2. DOI: [10.1145/3290605.3300358](https://doi.org/10.1145/3290605.3300358).
- [170] Jessica Hullman, Nicholas Diakopoulos, and Eytan Adar. “Contextifier: Automatic Generation of Annotated Stock Visualizations.” In: *Proceedings of the 2013 CHI Conference on Human Factors in Computing Systems*. CHI '13. New York, NY, USA: Association for Computing Machinery, Apr. 2013, pp. 2707–2716. ISBN: 978-1-4503-1899-0. DOI: [10.1145/2470654.2481374](https://doi.org/10.1145/2470654.2481374). (Visited on 07/07/2022).
- [171] Jessica Hullman, Yea-Seul Kim, Francis Nguyen, Lauren Speers, and Maneesh Agrawala. “Improving Comprehension of Measurements Using Concrete Re-expression Strategies.” In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. Montreal QC Canada: ACM, Apr. 2018, pp. 1–12. ISBN: 978-1-4503-5620-6. DOI: [10.1145/3173574.3173608](https://doi.org/10.1145/3173574.3173608). (Visited on 11/24/2022).
- [172] Jessica Hullman, Paul Resnick, and Eytan Adar. “Hypothetical Outcome Plots Outperform Error Bars and Violin Plots for Inferences about Reliability of Variable Ordering.” In: *PLOS ONE* 10.11 (Nov. 2015), e0142444. ISSN: 1932-6203. DOI: [10.1371/journal.pone.0142444](https://doi.org/10.1371/journal.pone.0142444). (Visited on 01/12/2023).
- [173] Joseph A. Huwaldt. *Plot Digitizer*. <http://plotdigitizer.sourceforge.net/>. 2001. (Visited on 05/26/2022).
- [174] IEEE. *Resolution and Size*. <https://journals.ieeeauthorcenter.ieee.org/create-your-ieee-journal-article/create-graphics-for-your-article/resolution-and-size/>. 2022. (Visited on 06/14/2022).
- [175] IPFS. *IPFS is the Distributed Web*. 2015. URL: <https://ipfs.io>.
- [176] Junyong In and Sangseok Lee. “Statistical Data Presentation.” In: *Korean Journal of Anesthesiology* 70.3 (May 2017), p. 267. ISSN: 2005-6419, 2005-7563. DOI: [10.4097/kjae.2017.70.3.267](https://doi.org/10.4097/kjae.2017.70.3.267). (Visited on 03/23/2023).
- [177] Marcin Iwanowski, Arkadiusz Cacko, and Grzegorz Sarwas. “Comparing Images for Document Plagiarism Detection.” In: *Computer Vision and Graphics*. Ed. by Leszek J. Chmielewski, Amitava Datta, Ryszard Kozera, and Konrad Wojciechowski. Cham: Springer International Publishing, 2016, pp. 532–543. ISBN: 978-3-319-46418-3.
- [178] Daniel Johnson, Lennart E. Nacke, and Peta Wyeth. “All about That Base: Differing Player Experiences in Video Game Genres and the Unique Case of MOBA Games.” In: *Proceedings of the 2015 CHI Conference on Human Factors in Computing Systems*. CHI '15. New York, NY, USA: Association for Computing Machinery, Apr. 2015, pp. 2265–2274. ISBN: 978-1-4503-3145-6. DOI: [10.1145/2702123.2702447](https://doi.org/10.1145/2702123.2702447). (Visited on 05/31/2022).

- [179] Patrick W. Jordan, B. Thomas, Ian Lyall McClelland, and Bernard Weerdmeester, eds. *Usability Evaluation In Industry*. 1st edition. London ; Bristol, Pa: CRC Press, June 1996. ISBN: 978-0-7484-0460-5.
- [180] René Jorna. "A Comparison of Presentation and Representation: Linguistic and Pictorial." In: *Human-Computer Interaction*. Ed. by Gerrit C. van der Veer and Gijsbertus Mulder. Berlin, Heidelberg: Springer, 1988, pp. 172–185. ISBN: 978-3-642-73402-1. DOI: [10.1007/978-3-642-73402-1\\_11](https://doi.org/10.1007/978-3-642-73402-1_11).
- [181] Eunice Jun, Maureen Daum, Jared Roesch, Sarah Chasins, Emery Berger, Rene Just, and Katharina Reinecke. "Tea: A High-level Language and Runtime System for Automating Statistical Analysis." In: *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. UIST '19. New York, NY, USA: Association for Computing Machinery, Oct. 2019, pp. 591–603. ISBN: 978-1-4503-6816-2. DOI: [10.1145/3332165.3347940](https://doi.org/10.1145/3332165.3347940). (Visited on 03/31/2023).
- [182] Daekyoung Jung, Wonjae Kim, Hyunjoo Song, Jeong-in Hwang, Bongshin Lee, Bohyoung Kim, and Jinwook Seo. "ChartSense: Interactive Data Extraction from Chart Images." In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI '17. Denver, Colorado, USA: Association for Computing Machinery, 2017, pp. 6706–6717. ISBN: 978-1-4503-4655-9. DOI: [10.1145/3025453.3025957](https://doi.org/10.1145/3025453.3025957). URL: <http://doi.acm.org/10.1145/3025453.3025957>.
- [183] Hernisa Kacorri, Kris M. Kitani, Jeffrey P. Bigham, and Chieko Asakawa. "People with Visual Impairment Training Personal Object Recognizers: Feasibility and Challenges." In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI '17. New York, NY, USA: Association for Computing Machinery, May 2017, pp. 5839–5849. ISBN: 978-1-4503-4655-9. DOI: [10.1145/3025453.3025899](https://doi.org/10.1145/3025453.3025899). (Visited on 05/31/2022).
- [184] Kushal Kafle, Brian Price, Scott Cohen, and Christopher Kanan. "DVQA: Understanding Data Visualizations via Question Answering." In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT: IEEE, June 2018, pp. 5648–5656. ISBN: 978-1-5386-6420-9. DOI: [10.1109/CVPR.2018.00592](https://doi.org/10.1109/CVPR.2018.00592). (Visited on 06/15/2022).
- [185] Alex Kale, Francis Nguyen, Matthew Kay, and Jessica Hullman. "Hypothetical Outcome Plots Help Untrained Observers Judge Trends in Ambiguous Data." In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (Jan. 2019), pp. 892–902. ISSN: 1941-0506. DOI: [10.1109/TVCG.2018.2864909](https://doi.org/10.1109/TVCG.2018.2864909).
- [186] Seokbin Kang, Ekta Shokeen, Virginia L. Byrne, Leyla Norooz, Elizabeth Bonsignore, Caro Williams-Pierce, and Jon E. Froehlich. "ARMath: Augmenting Everyday Life with Math Learning." In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. New York, NY, USA: Association for Computing Machinery, Apr. 2020, pp. 1–15. ISBN: 978-1-4503-6708-0. DOI: [10.1145/3313831.3376252](https://doi.org/10.1145/3313831.3376252).



- [187] Jonathan P. Kastellec and Eduardo L. Leoni. "Using Graphs Instead of Tables in Political Science." In: *Perspectives on Politics* 5.4 (Dec. 2007), pp. 755–771. ISSN: 1541-0986, 1537-5927. DOI: [10.1017/S1537592707072209](https://doi.org/10.1017/S1537592707072209). (Visited on 03/23/2023).
- [188] Matthew Kay, Gregory L. Nelson, and Eric B. Hekler. "Researcher-Centered Design of Statistics: Why Bayesian Statistics Better Fit the Culture and Incentives of HCI." In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. San Jose California USA: ACM, May 2016, pp. 4521–4532. ISBN: 978-1-4503-3362-7. DOI: [10.1145/2858036.2858465](https://doi.org/10.1145/2858036.2858465). (Visited on 03/22/2023).
- [189] Rubaiat Habib Kazi, Fanny Chevalier, Tovi Grossman, Shengdong Zhao, and George Fitzmaurice. "Draco: Bringing Life to Illustrations with Kinetic Textures." In: *Proceedings of the 2014 CHI Conference on Human Factors in Computing Systems*. Toronto Ontario Canada: ACM, Apr. 2014, pp. 351–360. ISBN: 978-1-4503-2473-1. DOI: [10.1145/2556288.2556987](https://doi.org/10.1145/2556288.2556987).
- [190] Alice Keller. "In Print or On Screen? Investigating the Reading Habits of Undergraduate Students Using Photo-Diaries and Photo-Interviews." In: *Libri* 62.1 (Mar. 2012), pp. 1–18. ISSN: 1865-8423. DOI: [10.1515/libri-2012-0001](https://doi.org/10.1515/libri-2012-0001).
- [191] Dave Kelly, Jaap Jasperse, and Ian Westbrooke. "Designing Science Graphs for Data Analysis and Presentation." In: *Science & Technical Publishing, Department of Conservation, Wellington, New Zealand* 32 (2005), p. 68.
- [192] Alison Kidd. "The Marks Are on the Knowledge Worker." In: *Proceedings of the 1994 CHI Conference on Human Factors in Computing Systems*. CHI '94. New York, NY, USA: Association for Computing Machinery, Apr. 1994, pp. 186–191. ISBN: 978-0-89791-650-9. DOI: [10.1145/191666.191740](https://doi.org/10.1145/191666.191740).
- [193] Dae Hyun Kim, Enamul Hoque, and Maneesh Agrawala. "Answering Questions about Charts and Generating Visual Explanations." In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. Honolulu, HI, USA: Association for Computing Machinery, Apr. 2020, pp. 1–13. ISBN: 978-1-4503-6708-0. DOI: [10.1145/3313831.3376467](https://doi.org/10.1145/3313831.3376467). URL: <https://doi.org/10.1145/3313831.3376467> (visited on 05/30/2022).
- [194] N. W. Kim, S. C. Joyner, A. Riegelhuth, and Y. Kim. "Accessible Visualization: Design Space, Opportunities, and Challenges." In: *Computer Graphics Forum* 40.3 (2021), pp. 173–188. ISSN: 1467-8659. DOI: [10.1111/cgf.14298](https://doi.org/10.1111/cgf.14298). (Visited on 09/08/2022).
- [195] Sunjung Kim and Linda J. Lombardino. "Comparing Graphs and Text: Effects of Complexity and Task." In: *Journal of Eye Movement Research* 8.3 (Oct. 2015), p. 17. ISSN: 1995-8692. DOI: [10.16910/jemr.8.3.2](https://doi.org/10.16910/jemr.8.3.2).

- [196] Yea-Seul Kim, Jake M Hofman, and Daniel G Goldstein. “Putting Scientific Results in Perspective: Improving the Communication of Standardized Effect Sizes.” In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. New Orleans LA USA: ACM, Apr. 2022, pp. 1–14. ISBN: 978-1-4503-9157-3. DOI: [10.1145/3491102.3502053](https://doi.org/10.1145/3491102.3502053). (Visited on 01/09/2023).
- [197] Yea-Seul Kim, Jessica Hullman, and Maneesh Agrawala. “Generating Personalized Spatial Analogies for Distances and Areas.” In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. San Jose California USA: ACM, May 2016, pp. 38–48. ISBN: 978-1-4503-3362-7. DOI: [10.1145/2858036.2858440](https://doi.org/10.1145/2858036.2858440). (Visited on 11/24/2022).
- [198] Torsten L Klein. “Communicating Quantitative Information: Tables vs Graphs.” In: *MPRA 60514* (2014), p. 20.
- [199] Rex B. Kline. *Beyond Significance Testing: Reforming Data Analysis Methods in Behavioral Research*. Beyond Significance Testing: Reforming Data Analysis Methods in Behavioral Research. Washington, DC, US: American Psychological Association, 2004, pp. xii, 325. ISBN: 978-1-59147-118-9. DOI: [10.1037/10693-000](https://doi.org/10.1037/10693-000).
- [200] Clemens N. Klokmoose, James R. Eagan, Siemen Baader, Wendy Mackay, and Michel Beaudouin-Lafon. “Webstrates: Shareable Dynamic Media.” In: *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. UIST '15. Charlotte, NC, USA: ACM, 2015, pp. 280–290. ISBN: 978-1-4503-3779-3. DOI: [10.1145/2807442.2807446](https://doi.org/10.1145/2807442.2807446). URL: <http://doi.acm.org/10.1145/2807442.2807446>.
- [201] Nicholas Kong and Maneesh Agrawala. “Graphical Overlays: Using Layered Elements to Aid Chart Reading.” In: *IEEE Transactions on Visualization and Computer Graphics* 18.12 (Dec. 2012), pp. 2631–2638. ISSN: 1941-0506. DOI: [10.1109/TVCG.2012.229](https://doi.org/10.1109/TVCG.2012.229).
- [202] Nicholas Kong, Marti A. Hearst, and Maneesh Agrawala. “Extracting References between Text and Charts via Crowdsourcing.” In: *Proceedings of the 2014 CHI Conference on Human Factors in Computing Systems*. CHI '14. Toronto Ontario Canada: ACM, Apr. 2014, pp. 31–40. ISBN: 978-1-4503-2473-1. DOI: [10.1145/2556288.2557241](https://doi.org/10.1145/2556288.2557241). (Visited on 06/15/2022).
- [203] Markus Konkol, Christian Kray, and Jan Suleiman. “Creating Interactive Scientific Publications Using Bindings.” In: *Proceedings of the ACM on Human-Computer Interaction* 3.EICS (June 2019), pp. 1–18. ISSN: 2573-0142. DOI: [10.1145/3331158](https://doi.org/10.1145/3331158).
- [204] Miha Kovač and Adriaan van der Weel. “Reading in a Post-Textual Era.” In: *First Monday* (Sept. 2018). ISSN: 1396-0466. DOI: [10.5210/fm.v23i10.9416](https://doi.org/10.5210/fm.v23i10.9416).
- [205] Martin Krzywinski and Naomi Altman. “Error Bars.” In: *Nature Methods* 10.10 (Oct. 2013), pp. 921–922. ISSN: 1548-7105. DOI: [10.1038/nmeth.2659](https://doi.org/10.1038/nmeth.2659). (Visited on 03/23/2023).

- [206] Sri H Kurniawan and Panayiotis Zaphiris. "Reading Online or on Paper: Which Is Faster?" In: (2001), p. 3.
- [207] Bum Chul Kwon, Florian Stoffel, and Dominik Jäckle. "VisJockey: Enriching Data Stories through Orchestrated Interactive Visualization." In: *Computation + Journalism Symposium*. New York, NY, USA, 2014, p. 5.
- [208] U.S. Bureau of Labor Statistics. *Employment Situation Summary - 2022 Mo8 Results*. <https://www.bls.gov/news.release/empsit.nro.htm>. 2022.
- [209] Chufan Lai, Zhixian Lin, Ruike Jiang, Yun Han, Can Liu, and Xiaoru Yuan. "Automatic Annotation Synchronizing with Textual Description for Visualization." In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. Honolulu, HI, USA: Association for Computing Machinery, Apr. 2020, pp. 1–13. ISBN: 978-1-4503-6708-0. DOI: [10.1145/3313831.3376443](https://doi.org/10.1145/3313831.3376443). URL: <https://doi.org/10.1145/3313831.3376443>.
- [210] Daniel Lakens. "Calculating and Reporting Effect Sizes to Facilitate Cumulative Science: A Practical Primer for t-Tests and ANOVAs." In: *Frontiers in Psychology* 4 (2013). ISSN: 1664-1078. (Visited on 01/19/2023).
- [211] Daniel Lakens. *The Practical Alternative to the P-Value Is the Correctly Used p-Value*. Apr. 2019. DOI: [10.31234/osf.io/shm8v](https://doi.org/10.31234/osf.io/shm8v). (Visited on 01/28/2023).
- [212] Matthew Lamm, Arun Tejasvi Chaganty, Christopher D. Manning, Dan Jurafsky, and Percy Liang. *Textual Analogy Parsing: What's Shared and What's Compared among Analogous Facts*. Sept. 2018. arXiv: [1809.02700 \[cs\]](https://arxiv.org/abs/1809.02700).
- [213] Jill H. Larkin and Herbert A. Simon. "Why a Diagram Is (Sometimes) Worth Ten Thousand Words." In: *Cognitive Science* 11.1 (1987), pp. 65–100. ISSN: 1551-6709. DOI: [10.1111/j.1551-6708.1987.tb00863.x](https://doi.org/10.1111/j.1551-6708.1987.tb00863.x). (Visited on 07/21/2022).
- [214] Elina Late, Carol Tenopir, Sanna Talja, and Lisa Christian. "Reading Practices in Scholarly Work: From Articles and Books to Blogs." In: *Journal of Documentation* 75.3 (May 2019), pp. 478–499. ISSN: 0022-0418. DOI: [10.1108/JD-11-2018-0178](https://doi.org/10.1108/JD-11-2018-0178).
- [215] Shahid Latif, Diao Liu, and Fabian Beck. "Exploring Interactive Linking Between Text and Visualization." In: *EuroVis 2018 - Short Papers*. Brno Czech Republic: The Eurographics Association, 2018, 4 pages. ISBN: 9783038680604. DOI: [10.2312/EUROVISSHORT.20181084](https://doi.org/10.2312/EUROVISSHORT.20181084).
- [216] Shahid Latif, Kaidie Su, and Fabian Beck. "Authoring Combined Textual and Visual Descriptions of Graph Data." In: *EuroVis*. Porto, Portugal: IEEE, 2019, p. 5.

- [217] Shahid Latif, Zheng Zhou, Yoon Kim, Fabian Beck, and Nam Wook Kim. “Kori: Interactive Synthesis of Text and Charts in Data Documents.” In: *IEEE Transactions on Visualization and Computer Graphics* 28.1 (Jan. 2022), pp. 184–194. ISSN: 1941-0506. DOI: [10.1109/TVCG.2021.3114802](https://doi.org/10.1109/TVCG.2021.3114802).
- [218] Marie-Paule Lecoutre, Jacques Poitevineau, and Bruno Lecoutre. “Even statisticians are not immune to misinterpretations of Null Hypothesis Significance Tests.” In: *International Journal of Psychology* 38.1 (2003), pp. 37–45. ISSN: 1464-066X. DOI: [10.1080/00207590244000250](https://doi.org/10.1080/00207590244000250). (Visited on 03/22/2023).
- [219] Elsie Lee-Robbins, Shiqing He, and Eytan Adar. “Learning Objectives, Insights, and Assessments: How Specification Formats Impact Design.” In: *IEEE Transactions on Visualization and Computer Graphics* 28.1 (Jan. 2022), pp. 676–685. ISSN: 1941-0506. DOI: [10.1109/TVCG.2021.3114811](https://doi.org/10.1109/TVCG.2021.3114811).
- [220] Yuxuan Lei, Qi Lu, and Yingqing Xu. “O&O: A DIY Toolkit for Designing and Rapid Prototyping Olfactory Interfaces.” In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. New Orleans LA USA: ACM, Apr. 2022, pp. 1–21. ISBN: 978-1-4503-9157-3. DOI: [10.1145/3491102.3502033](https://doi.org/10.1145/3491102.3502033).
- [221] Luis A. Leiva, Daniel Martín-Albo, Réjean Plamondon, and Radu-Daniel Vatavu. “KeyTime: Super-Accurate Prediction of Stroke Gesture Production Times.” In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. Montreal QC Canada: ACM, Apr. 2018, pp. 1–12. ISBN: 978-1-4503-5620-6. DOI: [10.1145/3173574.3173813](https://doi.org/10.1145/3173574.3173813). (Visited on 06/17/2022).
- [222] Robert H. Lengel and Richard L. Daft. “The Selection of Communication Media as an Executive Skill.” In: *The Academy of Management Executive* (1987-1989) 2.3 (1988), pp. 225–232. ISSN: 0896-3789. JSTOR: [4164833](https://www.jstor.org/stable/4164833). (Visited on 06/20/2023).
- [223] Bo Leuf and Ward Cunningham. *The Wiki Way: Collaboration and Sharing on the Internet: Quick Collaboration on the Web*. 1st edition. Boston: Addison-Wesley Professional, Apr. 2001. ISBN: 978-0-201-71499-9.
- [224] Stefan Leutenegger, Margarita Chli, and Roland Y. Siegwart. “BRISK: Binary Robust Invariant Scalable Keypoints.” In: *Proceedings of the 2011 International Conference on Computer Vision. ICCV '11*. Washington, DC, USA: IEEE Computer Society, 2011, pp. 2548–2555. ISBN: 978-1-4577-1101-5. DOI: [10.1109/ICCV.2011.6126542](https://doi.org/10.1109/ICCV.2011.6126542). URL: <http://dx.doi.org/10.1109/ICCV.2011.6126542>.
- [225] Christophe Leys. “Why Psychologists Should by Default Use Welch’s t-Test Instead of Student’s t-Test.” In: 30.1 (Apr. 2017), p. 92. ISSN: 2397-8570. DOI: [10.5334/irsp.82](https://doi.org/10.5334/irsp.82). (Visited on 04/03/2023).

- [226] Deqing Li, Honghui Mei, Yi Shen, Shuang Su, Wenli Zhang, Junting Wang, Ming Zu, and Wei Chen. "ECharts: A Declarative Framework for Rapid Construction of Web-Based Visualization." In: *Visual Informatics* 2.2 (June 2018), pp. 136–146. ISSN: 2468-502X. DOI: [10.1016/j.visinf.2018.04.011](https://doi.org/10.1016/j.visinf.2018.04.011).
- [227] Haotian Li, Yong Wang, Aoyu Wu, Huan Wei, and Huamin Qu. "Structure-Aware Visualization Retrieval." In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. CHI '22. New York, NY, USA: Association for Computing Machinery, Apr. 2022, pp. 1–14. ISBN: 978-1-4503-9157-3. DOI: [10.1145/3491102.3502048](https://doi.org/10.1145/3491102.3502048). (Visited on 05/27/2022).
- [228] Tzu-Mao Li, Michal Lukáč, Michaël Gharbi, and Jonathan Ragan-Kelley. "Differentiable Vector Graphics Rasterization for Editing and Learning." In: *ACM Transactions on Graphics* 39.6 (Nov. 2020), 193:1–193:15. ISSN: 0730-0301. DOI: [10.1145/3414685.3417871](https://doi.org/10.1145/3414685.3417871). (Visited on 01/23/2023).
- [229] Guangping Liang, Wenliang Fu, and Kaifa Wang. "Analysis of T-Test Misuses and SPSS Operations in Medical Research Papers." In: *Burns & Trauma* 7 (Jan. 2019), s41038–019–0170–3. ISSN: 2321-3876. DOI: [10.1186/s41038-019-0170-3](https://doi.org/10.1186/s41038-019-0170-3). (Visited on 04/04/2023).
- [230] Chunyuan Liao, François Guimbretière, Ken Hinckley, Ken Hinckley, and Jim Hollan. "Papiercraft: A Gesture-based Command System for Interactive Paper." In: *ACM Trans. Comput.-Hum. Interact.* 14.4 (Jan. 2008), 18:1–18:27. ISSN: 1073-0516. DOI: [10.1145/1314683.1314686](https://doi.org/10.1145/1314683.1314686). URL: <http://doi.acm.org/10.1145/1314683.1314686>.
- [231] Can Liu, Olivier Chapuis, Michel Beaudouin-Lafon, and Eric Lecolinet. "Shared Interaction on a Wall-Sized Display in a Data Manipulation Task." In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. CHI '16. New York, NY, USA: Association for Computing Machinery, May 2016, pp. 2075–2086. ISBN: 978-1-4503-3362-7. DOI: [10.1145/2858036.2858039](https://doi.org/10.1145/2858036.2858039). (Visited on 06/17/2022).
- [232] Yang Liu, Alex Kale, Tim Althoff, and Jeffrey Heer. "Boba: Authoring and Visualizing Multiverse Analyses." In: *IEEE Transactions on Visualization and Computer Graphics* 27.2 (Feb. 2021), pp. 1753–1763. ISSN: 1941-0506. DOI: [10.1109/TVCG.2020.3028985](https://doi.org/10.1109/TVCG.2020.3028985).
- [233] Ziming Liu. "Reading Behavior in the Digital Environment: Changes in Reading Behavior over the Past Ten Years." In: *Journal of Documentation* 61.6 (Jan. 2005), pp. 700–712. ISSN: 0022-0418. DOI: [10.1108/00220410510632040](https://doi.org/10.1108/00220410510632040).
- [234] Ziming Liu. "Print vs. Electronic Resources: A Study of User Perceptions, Preferences, and Use." In: *Information Processing & Management* 42.2 (Mar. 2006), pp. 583–592. ISSN: 0306-4573. DOI: [10.1016/j.ipm.2004.12.002](https://doi.org/10.1016/j.ipm.2004.12.002).

- [235] Geoffrey R. Loftus. "A Picture Is Worth a Thousandp Values: On the Irrelevance of Hypothesis Testing in the Microcomputer Age." In: *Behavior Research Methods, Instruments, & Computers* 25.2 (June 1993), pp. 250–256. ISSN: 0743-3808, 1532-5970. DOI: [10.3758/BF03204506](https://doi.org/10.3758/BF03204506).
- [236] David G. Lowe. "Distinctive Image Features from Scale-Invariant Keypoints." In: *Int. J. Comput. Vision* 60.2 (Nov. 2004), pp. 91–110. ISSN: 0920-5691. DOI: [10.1023/B:VISI.0000029664.99615.94](https://doi.org/10.1023/B:VISI.0000029664.99615.94). URL: <https://doi.org/10.1023/B:VISI.0000029664.99615.94>.
- [237] Min Lu, Jie Liang, Yu Zhang, Guozheng Li, Siming Chen, Zongru Li, and Xiaoru Yuan. "Interaction+: Interaction Enhancement for Web-Based Visualizations." In: *2017 IEEE Pacific Visualization Symposium (PacificVis)*. Seoul Republic of Korea: IEEE, Apr. 2017, pp. 61–70. DOI: [10.1109/PACIFICVIS.2017.8031580](https://doi.org/10.1109/PACIFICVIS.2017.8031580).
- [238] Yuyu Luo, Xuedi Qin, Nan Tang, and Guoliang Li. "DeepEye: Towards Automatic Data Visualization." In: *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. Paris: IEEE, Apr. 2018, pp. 101–112. ISBN: 978-1-5386-5520-7. DOI: [10.1109/ICDE.2018.00019](https://doi.org/10.1109/ICDE.2018.00019).
- [239] Yuyu Luo, Nan Tang, Guoliang Li, Jiawei Tang, Chengliang Chai, and Xuedi Qin. "Natural Language to Visualization by Neural Machine Translation." In: *IEEE Transactions on Visualization and Computer Graphics* 28.1 (Jan. 2022), pp. 217–226. ISSN: 1941-0506. DOI: [10.1109/TVCG.2021.3114848](https://doi.org/10.1109/TVCG.2021.3114848).
- [240] Shuai Ma, Taichang Zhou, Fei Nie, and Xiaojuan Ma. "Glancee: An Adaptable System for Instructors to Grasp Student Learning Status in Synchronous Online Classes." In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. New Orleans LA USA: ACM, Apr. 2022, pp. 1–25. ISBN: 978-1-4503-9157-3. DOI: [10.1145/3491102.3517482](https://doi.org/10.1145/3491102.3517482).
- [241] Weihong Ma, Hesuo Zhang, Shuang Yan, Guangshun Yao, Yichao Huang, Hui Li, Yaqiang Wu, and Lianwen Jin. *Towards an Efficient Framework for Data Extraction from Chart Images*. May 2021. arXiv: [arXiv: 2105.02039](https://arxiv.org/abs/2105.02039). (Visited on 08/17/2022).
- [242] Michael MacDonald-Ross. "How Numbers Are Shown: A Review of Research on the Presentation of Quantitative Data in Texts." In: *AV Communication Review* 25.4 (1977), pp. 359–409. ISSN: 0001-2890.
- [243] Eva Mackamul. "Improving the Discoverability of Interactions in Interactive Systems." In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems Extended Abstracts*. CHI EA '22. New Orleans, LA, USA: Association for Computing Machinery, 2022. ISBN: 9781450391566. DOI: [10.1145/3491101.3503813](https://doi.org/10.1145/3491101.3503813). URL: <https://doi.org/10.1145/3491101.3503813>.
- [244] Heinz Mandl and Joel R. Levin. *Knowledge Acquisition from Text and Pictures*. Amsterdam ; New York : New York, N.Y., U.S.A: North-Holland, June 1989. ISBN: 978-0-444-87353-8.

- [245] Baurzhan Muftakhidinov Mark Mitchell and Tobias Winchen et al. *Engauge Digitizer*. <http://markummittchell.github.io/engauge-digitizer/>. 2015. (Visited on 05/26/2022).
- [246] Damien Masson, Sylvain Malacria, Géry Casiez, and Daniel Vogel. "Charagraph: Interactive Generation of Charts for Realtime Annotation of Data-Rich Paragraphs." In: *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. CHI '23. Hamburg, Germany: Association for Computing Machinery, 2023. ISBN: 978-1-4503-9421-5/23/04. DOI: [10.1145/3544548.3581091](https://doi.org/10.1145/3544548.3581091). URL: <https://doi.org/10.1145/3544548.3581091>.
- [247] Damien Masson, Sylvain Malacria, Edward Lank, and Géry Casiez. "Chameleon: Bringing Interactivity to Static Digital Documents." In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. Honolulu HI USA: ACM, Apr. 2020, pp. 1–13. ISBN: 978-1-4503-6708-0. DOI: [10.1145/3313831.3376559](https://doi.org/10.1145/3313831.3376559). URL: <https://doi.org/10.1145/3313831.3376559> (visited on 03/22/2023).
- [248] Damien Masson, Sylvain Malacria, Daniel Vogel, Edward Lank, and Géry Casiez. "ChartDetective: Easy and Accurate Interactive Data Extraction from Complex Vector Charts." In: *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. CHI '23. Hamburg, Germany: Association for Computing Machinery, 2023. ISBN: 978-1-4503-9421-5/23/04. DOI: [10.1145/3544548.3581113](https://doi.org/10.1145/3544548.3581113). URL: <https://doi.org/10.1145/3544548.3581113>.
- [249] Damien Masson, Jo Vermeulen, George Fitzmaurice, and Justin Matejka. "Supercharging Trial-and-Error for Learning Complex Software Applications." In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. CHI '22. New Orleans, LA, USA: Association for Computing Machinery, 2022. ISBN: 9781450391573. DOI: [10.1145/3491102.3501895](https://doi.org/10.1145/3491102.3501895). URL: <https://doi.org/10.1145/3491102.3501895>.
- [250] J Matas, O Chum, M Urban, and T Pajdla. "Robust wide-baseline stereo from maximally stable extremal regions." In: *Image and Vision Computing* 22.10 (2004). British Machine Vision Computing 2002, pp. 761–767. ISSN: 0262-8856. DOI: [10.1016/j.imavis.2004.02.006](http://www.sciencedirect.com/science/article/pii/S0262885604000435). URL: <http://www.sciencedirect.com/science/article/pii/S0262885604000435>.
- [251] A. Mathisen, T. Horak, C. N. Klokmose, K. Grønbaek, and N. Elmqvist. "InsideInsights: Integrating Data-Driven Reporting in Collaborative Visual Analytics." In: *Computer Graphics Forum* 38.3 (2019), pp. 649–661. DOI: <https://doi.org/10.1111/cgf.13717>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13717>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13717>.
- [252] J. Nathan Matias, Sayamindu Dasgupta, and Benjamin Mako Hill. "Skill Progression in Scratch Revisited." In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. CHI '16. New York, NY, USA: Association for Computing Machinery, May 2016, pp. 1486–

1490. ISBN: 978-1-4503-3362-7. DOI: [10.1145/2858036.2858349](https://doi.org/10.1145/2858036.2858349). (Visited on 06/17/2022).
- [253] Richard E. Mayer. "Multimedia Learning." In: *Psychology of Learning and Motivation*. Vol. 41. Academic Press, Jan. 2002, pp. 85–139. DOI: [10.1016/S0079-7421\(02\)80005-6](https://doi.org/10.1016/S0079-7421(02)80005-6).
- [254] Andrew McNutt, Gordon Kindlmann, and Michael Correll. "Surfacing Visualization Mirages." In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. Honolulu HI USA: ACM, Apr. 2020, pp. 1–16. ISBN: 978-1-4503-6708-0. DOI: [10.1145/3313831.3376420](https://doi.org/10.1145/3313831.3376420). (Visited on 11/20/2022).
- [255] Blakeley B. McShane and David Gal. "Blinding Us to the Obvious? The Effect of Statistical Training on the Evaluation of Evidence." In: *Management Science* 62.6 (June 2016), pp. 1707–1718. ISSN: 0025-1909. DOI: [10.1287/mnsc.2015.2212](https://doi.org/10.1287/mnsc.2015.2212). (Visited on 03/22/2023).
- [256] Gonzalo Gabriel Méndez, Miguel A. Nacenta, and Sebastien Vandenhete. "iVoLVER: Interactive Visual Language for Visualization Extraction and Reconstruction." In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. CHI '16. New York, NY, USA: Association for Computing Machinery, May 2016, pp. 4073–4085. ISBN: 978-1-4503-3362-7. DOI: [10.1145/2858036.2858435](https://doi.org/10.1145/2858036.2858435). (Visited on 05/26/2022).
- [257] Ronald Metoyer, Qiyu Zhi, Bart Janczuk, and Walter Scheirer. "Coupling Story to Visualization: Using Textual Analysis as a Bridge Between Data and Interpretation." In: *23rd International Conference on Intelligent User Interfaces*. Tokyo Japan: ACM, Mar. 2018, pp. 503–507. ISBN: 978-1-4503-4945-1. DOI: [10.1145/3172944.3173007](https://doi.org/10.1145/3172944.3173007).
- [258] Norman Meuschke, Christopher Gondek, Daniel Seebacher, Corinna Breitingner, Daniel Keim, and Bela Gipp. "An Adaptive Image-based Plagiarism Detection Approach." In: *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries*. JCDL '18. Fort Worth, Texas, USA: ACM, 2018, pp. 131–140. ISBN: 978-1-4503-5178-2. DOI: [10.1145/3197026.3197042](https://doi.org/10.1145/3197026.3197042). URL: <http://doi.acm.org/10.1145/3197026.3197042>.
- [259] Joachim Meyer, Marcia Kuskin Shamo, and Daniel Gopher. "Information Structure and the Relative Efficacy of Tables and Graphs." In: *Human Factors* 41.4 (Dec. 1999), pp. 570–587. ISSN: 0018-7208. DOI: [10.1518/001872099779656707](https://doi.org/10.1518/001872099779656707).
- [260] K. Mikolajczyk and C. Schmid. "A performance evaluation of local descriptors." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.10 (Oct. 2005), pp. 1615–1630. ISSN: 0162-8828. DOI: [10.1109/TPAMI.2005.188](https://doi.org/10.1109/TPAMI.2005.188).



## BIBLIOGRAPHY

- [261] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. "A Comparison of Affine Region Detectors." In: *Int. J. Comput. Vision* 65.1-2 (Nov. 2005), pp. 43–72. ISSN: 0920-5691. DOI: [10.1007/s11263-005-3848-x](https://doi.org/10.1007/s11263-005-3848-x). URL: <http://dx.doi.org/10.1007/s11263-005-3848-x>.
- [262] Rupert G. Miller. *Simultaneous Statistical Inference*. Springer Series in Statistics. New York, NY: Springer, 1981. ISBN: 978-1-4613-8124-2 978-1-4613-8122-8. DOI: [10.1007/978-1-4613-8122-8](https://doi.org/10.1007/978-1-4613-8122-8). (Visited on 04/03/2023).
- [263] Toshiya Miyatsu, Khuyen Nguyen, and Mark A. McDaniel. "Five Popular Study Strategies: Their Pitfalls and Optimal Implementations." In: *Perspectives on Psychological Science* 13.3 (May 2018), pp. 390–407. ISSN: 1745-6916. DOI: [10.1177/1745691617710510](https://doi.org/10.1177/1745691617710510). (Visited on 05/03/2023).
- [264] Jennifer C. Molloy. "The Open Knowledge Foundation: Open Data Means Better Science." In: *PLoS Biology* 9.12 (Dec. 2011), e1001195. ISSN: 1544-9173. DOI: [10.1371/journal.pbio.1001195](https://doi.org/10.1371/journal.pbio.1001195).
- [265] Pam A. Mueller and Daniel M. Oppenheimer. "The Pen Is Mightier Than the Keyboard: Advantages of Longhand Over Laptop Note Taking." In: *Psychological Science* 25.6 (June 2014), pp. 1159–1168. ISSN: 0956-7976. DOI: [10.1177/0956797614524581](https://doi.org/10.1177/0956797614524581).
- [266] Marcus R. Munafò, Brian A. Nosek, Dorothy V. M. Bishop, Katherine S. Button, Christopher D. Chambers, Nathalie Percie du Sert, Uri Simonsohn, Eric-Jan Wagenmakers, Jennifer J. Ware, and John P. A. Ioannidis. "A Manifesto for Reproducible Science." In: *Nature human behaviour* 1 (Jan. 2017), p. 0021. ISSN: 2397-3374. DOI: [10.1038/s41562-016-0021](https://doi.org/10.1038/s41562-016-0021).
- [267] Tamara Munzner. *Visualization Analysis and Design*. 1st edition. Boca Raton: A K Peters/CRC Press, Dec. 2014. ISBN: 978-1-4665-0891-0.
- [268] Paul Muter, Susane A. Latrémouille, William C. Treurniet, and Paul Beam. "Extended Reading of Continuous Text on Television Screens." In: *Human Factors* 24.5 (Oct. 1982), pp. 501–508. ISSN: 0018-7208. DOI: [10.1177/001872088202400501](https://doi.org/10.1177/001872088202400501).
- [269] Paul Muter and Paula Maurutto. "Reading and Skimming from Computer Screens and Books: The Paperless Office Revisited?" In: *Behaviour & Information Technology* 10.4 (July 1991), pp. 257–266. ISSN: 0144-929X, 1362-3001. DOI: [10.1080/01449299108924288](https://doi.org/10.1080/01449299108924288).
- [270] Arpit Narechania, Arjun Srinivasan, and John Stasko. "NL4DV: A Toolkit for Generating Analytic Specifications for Data Visualization from Natural Language Queries." In: *IEEE Transactions on Visualization and Computer Graphics* 27.2 (Feb. 2021), pp. 369–379. ISSN: 1077-2626, 1941-0506, 2160-9306. DOI: [10.1109/TVCG.2020.3030378](https://doi.org/10.1109/TVCG.2020.3030378).
- [271] Theodor Holm Nelson. *Literary Machines : Edition 87.1*. (S.l. : published by the author), 1987. ISBN: 978-0-89347-056-2.

- [272] Theodor Holm Nelson. "The Heart of Connection: Hypermedia Unified by Transclusion." In: *Commun. ACM* 38.8 (Aug. 1995), pp. 31–33. ISSN: 0001-0782. DOI: [10.1145/208344.208353](https://doi.org/10.1145/208344.208353). URL: <http://doi.acm.org/10.1145/208344.208353>.
- [273] Donald A. Norman. "How Might People Interact with Agents." In: *Communications of the ACM* 37.7 (July 1994), pp. 68–71. ISSN: 0001-0782, 1557-7317. DOI: [10.1145/176789.176796](https://doi.org/10.1145/176789.176796). (Visited on 06/20/2022).
- [274] C. North. "Toward Measuring Visualization Insight." In: *IEEE Computer Graphics and Applications* 26.3 (May 2006), pp. 6–9. ISSN: 1558-1756. DOI: [10.1109/MCG.2006.70](https://doi.org/10.1109/MCG.2006.70).
- [275] Michèle B. Nuijten, Marcel A. L. M. van Assen, C. H. J. Hartgerink, Sacha Epskamp, and Jelte Wicherts. *The Validity of the Tool "Statcheck" in Discovering Statistical Reporting Inconsistencies*. Nov. 2017. DOI: [10.31234/osf.io/tcxaj](https://doi.org/10.31234/osf.io/tcxaj). (Visited on 02/20/2023).
- [276] Michèle B. Nuijten and Joshua R. Polanin. "'Statcheck': Automatically Detect Statistical Reporting Inconsistencies to Increase Reproducibility of Meta-Analyses." In: *Research Synthesis Methods* 11.5 (2020), pp. 574–579. ISSN: 1759-2887. DOI: [10.1002/jrsm.1408](https://doi.org/10.1002/jrsm.1408). (Visited on 02/20/2023).
- [277] Daniel Nüst, Markus Konkol, Edzer Pebesma, Christian Kray, Marc Schutzeichel, Holger Przibytzin, and Jörg Lorenz. "Opening the Publication Process with Executable Research Compendia." In: *D-Lib Magazine* 23 (Jan. 2017). DOI: [10.1045/january2017-nuest](https://doi.org/10.1045/january2017-nuest).
- [278] Kenton O'Hara. *Towards a Typology of Reading Goals*. 1996.
- [279] Kenton O'Hara and Abigail Sellen. "A Comparison of Reading Paper and On-Line Documents." In: *Proceedings of the 1997 CHI Conference on Human Factors in Computing Systems*. Atlanta Georgia USA: ACM, Mar. 1997, pp. 335–342. ISBN: 978-0-89791-802-2. DOI: [10.1145/258549.258787](https://doi.org/10.1145/258549.258787).
- [280] Jason Obeid and Enamul Hoque. "Chart-to-Text: Generating Natural Language Descriptions for Charts by Adapting the Transformer Model." In: *Proceedings of the 13th International Conference on Natural Language Generation*. Dublin, Ireland: Association for Computational Linguistics, Dec. 2020, pp. 138–147. (Visited on 06/15/2022).
- [281] *Observable - Explore, Analyze, and Explain Data. As a Team*. <https://observablehq.com/>. 2018.
- [282] Dan R. Olsen. "Evaluating User Interface Systems Research." In: *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*. UIST '07. New York, NY, USA: Association for Computing Machinery, Oct. 2007, pp. 251–258. ISBN: 978-1-59593-679-0. DOI: [10.1145/1294211.1294256](https://doi.org/10.1145/1294211.1294256). (Visited on 06/22/2023).

- [283] Stephen Oney, Chris Harrison, Amy Ogan, and Jason Wiese. “Zoom-Board: A Diminutive Qwerty Soft Keyboard Using Iterative Zooming for Ultra-Small Devices.” In: *Proceedings of the 2013 CHI Conference on Human Factors in Computing Systems*. CHI '13. New York, NY, USA: Association for Computing Machinery, Apr. 2013, pp. 2799–2802. ISBN: 978-1-4503-1899-0. DOI: [10.1145/2470654.2481387](https://doi.org/10.1145/2470654.2481387). (Visited on 04/03/2023).
- [284] World Health Organization. *Weekly Epidemiological Update on COVID-19 - 22 June 2021*. <https://www.who.int/publications/m/item/weekly-epidemiological-update-on-covid-19-22-june-2021>. 2021.
- [285] World Health Organization. *Weekly Epidemiological Update on COVID-19 - 29 June 2021*. <https://www.who.int/publications/m/item/weekly-epidemiological-update-on-covid-19-29-june-2021>. 2021.
- [286] Magnus Österholm. “Characterizing Reading Comprehension of Mathematical Texts.” In: *Educational Studies in Mathematics* 63.3 (Nov. 2006), pp. 325–346. ISSN: 1573-0816. DOI: [10.1007/s10649-005-9016-y](https://doi.org/10.1007/s10649-005-9016-y).
- [287] Alvitta Ottley, Aleksandra Kaszowska, R. Jordan Crouser, and Evan M. Peck. “The Curious Case of Combining Text and Visualization.” In: *EuroVis*. Porto, Portugal: The Eurographics Association, 2019, p. 5. ISBN: 978-3-03868-090-1. DOI: [10.2312/evs.20191181](https://doi.org/10.2312/evs.20191181).
- [288] *PDF.js*. Mozilla, 2011. URL: <https://mozilla.github.io/pdf.js/>.
- [289] Allan Paivio. *Mental Representations: A Dual Coding Approach*. Oxford Psychology Series no. 9. New York : Oxford [Oxfordshire]: Oxford University Press ; Clarendon Press, 1986. ISBN: 978-0-19-503936-8.
- [290] Allan Paivio, Mary Walsh, and Trudy Bons. “Concreteness Effects on Memory: When and Why?” In: *Journal of Experimental Psychology: Learning, Memory, and Cognition* 20 (1994), pp. 1196–1204. ISSN: 1939-1285. DOI: [10.1037/0278-7393.20.5.1196](https://doi.org/10.1037/0278-7393.20.5.1196).
- [291] Allan Paivio, John C. Yuille, and Stephen A. Madigan. “Concreteness, Imagery, and Meaningfulness Values for 925 Nouns.” In: *Journal of Experimental Psychology* 76 (1968), pp. 1–25. ISSN: 0022-1015. DOI: [10.1037/h0025327](https://doi.org/10.1037/h0025327).
- [292] Anshul Vikram Pandey, Katharina Rall, Margaret L. Satterthwaite, Oded Nov, and Enrico Bertini. “How Deceptive Are Deceptive Visualizations? An Empirical Analysis of Common Distortion Techniques.” In: *Proceedings of the 2015 CHI Conference on Human Factors in Computing Systems*. CHI '15. New York, NY, USA: Association for Computing Machinery, Apr. 2015, pp. 1469–1478. ISBN: 978-1-4503-3145-6. DOI: [10.1145/2702123.2702608](https://doi.org/10.1145/2702123.2702608). (Visited on 12/11/2022).
- [293] Deokgun Park, Mohamed Suhail, Minsheng Zheng, Cody Dunne, Eric Ragan, and Niklas Elmquist. “StoryFacets: A design study on storytelling with visualizations for collaborative data analysis.” In: *Information Visualization* 21.1 (2022), pp. 3–16. DOI: [10.1177/14738716211032653](https://doi.org/10.1177/14738716211032653). eprint: <https://doi.org/10.1177/14738716211032653>.

- 1177 / 14738716211032653. URL: <https://doi.org/10.1177/14738716211032653>.
- [294] Jonas Parnow and Marian Dörk. “Micro Visualizations: Data-driven Typography and Graphical Text Enhancement.” In: *InfoVis*. Chicago, Illinois USA: IEEE, 2015, p. 2.
- [295] Erik Pasternak, Rachel Fenichel, and Andrew N. Marshall. “Tips for Creating a Block Language with Blockly.” In: *2017 IEEE Blocks and Beyond Workshop (B&B)*. Oct. 2017, pp. 21–24. DOI: [10.1109/BLOCKS.2017.8120404](https://doi.org/10.1109/BLOCKS.2017.8120404).
- [296] Panupong Pasupat and Percy Liang. “Compositional Semantic Parsing on Semi-Structured Tables.” In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, July 2015, pp. 1470–1480. DOI: [10.3115/v1/P15-1142](https://doi.org/10.3115/v1/P15-1142). (Visited on 06/15/2022).
- [297] Jennifer Pearson, George Buchanan, and Harold Thimbleby. *Designing for Digital Reading*. Cham: Springer International Publishing, 2014. ISBN: 978-3-031-01202-0 978-3-031-02330-9. DOI: [10.1007/978-3-031-02330-9](https://doi.org/10.1007/978-3-031-02330-9).
- [298] S. Pettifer, P. McDERMOTT, J. Marsh, D. Thorne, A. Villegger, and T.k. Attwood. “Ceci n’est Pas Un Hamburger: Modelling and Representing the Scholarly Article.” In: *Learned Publishing* 24.3 (2011), pp. 207–220. ISSN: 1741-4857. DOI: [10.1087/20110309](https://doi.org/10.1087/20110309).
- [299] Chanda Phelan, Jessica Hullman, Matthew Kay, and Paul Resnick. “Some Prior(s) Experience Necessary: Templates for Getting Started With Bayesian Analysis.” In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI ’19. New York, NY, USA: Association for Computing Machinery, May 2019, pp. 1–12. ISBN: 978-1-4503-5970-2. DOI: [10.1145/3290605.3300709](https://doi.org/10.1145/3290605.3300709). (Visited on 03/23/2023).
- [300] Joao Pinheiro and Jorge Poco. *ChartText: Linking Text with Charts in Documents*. Jan. 2022. arXiv: [2201.05043 \[cs\]](https://arxiv.org/abs/2201.05043).
- [301] Plotly. *Plotly*. 2012. URL: <https://plot.ly/>.
- [302] Jorge Poco and Jeffrey Heer. “Reverse-Engineering Visualizations: Recovering Visual Encodings from Chart Images.” In: *Comput. Graph. Forum* 36.3 (June 2017), pp. 353–363. ISSN: 0167-7055. DOI: [10.1111/cgf.13193](https://doi.org/10.1111/cgf.13193). URL: <https://doi.org/10.1111/cgf.13193>.
- [303] Jorge Poco, Angela Mayhua, and Jeffrey Heer. “Extracting and Retargeting Color Mappings from Bitmap Images of Visualizations.” In: *IEEE Transactions on Visualization and Computer Graphics* 24.1 (Jan. 2018), pp. 637–646. ISSN: 1941-0506. DOI: [10.1109/TVCG.2017.2744320](https://doi.org/10.1109/TVCG.2017.2744320).
- [304] T. Poisot. “The Digitize Package: Extracting Numerical Data from Scatterplots.” In: *The R Journal* 3.1 (2011), pp. 25–26.

- [305] Nadja Popovich, Blacki Migliozi, Rumsey Taylor, Josh Williams, and Derek Watkins. “How Much Hotter Is Your Hometown Than When You Were Born?” In: *The New York Times* (Aug. 2018). ISSN: 0362-4331. (Visited on 06/21/2023).
- [306] Zachary Pousman, John Stasko, and Michael Mateas. “Casual Information Visualization: Depictions of Data in Everyday Life.” In: *IEEE Transactions on Visualization and Computer Graphics* 13.6 (Nov. 2007), pp. 1145–1152. ISSN: 1941-0506. DOI: [10.1109/TVCG.2007.70541](https://doi.org/10.1109/TVCG.2007.70541).
- [307] Christopher Power. “Guidelines Are Only Half of the Story: Accessibility Problems Encountered by Blind Users on the Web.” In: (2012), p. 10.
- [308] Morgan N. Price, Bill N. Schilit, and Gene Golovchinsky. “XLibris: The Active Reading Machine.” In: *CHI 98 Conference Summary on Human Factors in Computing Systems*. CHI '98. New York, NY, USA: Association for Computing Machinery, Apr. 1998, pp. 22–23. ISBN: 978-1-58113-028-7. DOI: [10.1145/286498.286510](https://doi.org/10.1145/286498.286510).
- [309] *PrimeReact | React UI Component Library*. PrimeTek Informatics, 2017. URL: <https://www.primefaces.org/primereact>.
- [310] Chunyao Qian, Shizhao Sun, Weiwei Cui, Jian-Guang Lou, Haidong Zhang, and Dongmei Zhang. “Retrieve-Then-Adapt: Example-based Automatic Generation for Proportion-related Infographics.” In: *IEEE Transactions on Visualization and Computer Graphics* 27.2 (Feb. 2021), pp. 443–452. ISSN: 1941-0506. DOI: [10.1109/TVCG.2020.3030448](https://doi.org/10.1109/TVCG.2020.3030448).
- [311] Jing Qian, Qi Sun, Curtis Wigington, Han L. Han, Tong Sun, Jennifer Healey, James Tompkin, and Jeff Huang. “Dually Noted: Layout-Aware Annotations with Smartphone Augmented Reality.” In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. New Orleans LA USA: ACM, Apr. 2022, pp. 1–15. ISBN: 978-1-4503-9157-3. DOI: [10.1145/3491102.3502026](https://doi.org/10.1145/3491102.3502026).
- [312] Qt. *QWebEngineView*. 2008. URL: <https://doc.qt.io/qt-5/qwebengineview.html>.
- [313] Ghulam Jilani Abdul Rahim Quadri. “Constructing Frameworks for Task-Optimized Visualizations.” PhD thesis. United States – Florida: University of South Florida, 2021. ISBN: 9798496510165.
- [314] Ghulam Jilani Quadri and Paul Rosen. “A Survey of Perception-Based Visualization Studies by Task.” In: *IEEE Transactions on Visualization and Computer Graphics* 28.12 (Dec. 2022), pp. 5026–5048. ISSN: 1077-2626. DOI: [10.1109/TVCG.2021.3098240](https://doi.org/10.1109/TVCG.2021.3098240). (Visited on 12/12/2022).
- [315] Napol Rachatasumrit, Jonathan Bragg, Amy X. Zhang, and Daniel S Weld. “CiteRead: Integrating Localized Citation Contexts into Scientific Paper Reading.” In: *27th International Conference on Intelligent User Interfaces*. IUI '22. New York, NY, USA: Association for Computing Machinery, Mar. 2022, pp. 707–719. ISBN: 978-1-4503-9144-3. DOI: [10.1145/3490099.3511162](https://doi.org/10.1145/3490099.3511162).

- [316] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. “Improving Language Understanding by Generative Pre-Training.” In: ().
- [317] *React – A JavaScript Library for Building User Interfaces*. Meta, 2013. URL: <https://reactjs.org/>.
- [318] Donghao Ren, Matthew Brehmer, Bongshin Lee, Tobias Höllerer, and Eun Kyoung Choe. “ChartAccent: Annotation for Data-Driven Storytelling.” In: *2017 IEEE Pacific Visualization Symposium (PacificVis)*. Seoul Republic of Korea: IEEE, Apr. 2017, pp. 230–239. DOI: [10.1109/PACIFICVIS.2017.8031599](https://doi.org/10.1109/PACIFICVIS.2017.8031599).
- [319] Donghao Ren, Tobias Höllerer, and Xiaoru Yuan. “iVisDesigner: Expressive Interactive Design of Information Visualizations.” In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (Dec. 2014), pp. 2092–2101. ISSN: 1941-0506. DOI: [10.1109/TVCG.2014.2346291](https://doi.org/10.1109/TVCG.2014.2346291).
- [320] Grégoire Richard, Thomas Pietrzak, Ferran Argelaguet Sanz, Anatole Lécuyer, and Géry Casiez. “Within or Between? Comparing Experimental Designs for Virtual Embodiment Studies.” In: *Proceedings of the IEEE Conference on Virtual Reality (VR 2022)*. IEEE, Mar. 2022, p. 1. DOI: [10.1109/VR51125.2022.00037](https://doi.org/10.1109/VR51125.2022.00037). (Visited on 04/05/2023).
- [321] Nathalie Henry Riche, Christophe Hurter, Nicholas Diakopoulos, and Sheelagh Carpendale. *Data-Driven Storytelling*. Natick, Massachusetts, USA: CRC Press, Mar. 2018. ISBN: 978-1-315-28155-1.
- [322] Yann Riche, Nathalie Henry Riche, Ken Hinckley, Sheri Panabaker, Sarah Fuelling, and Sarah Williams. “As We May Ink?: Learning from Everyday Analog Pen Use to Improve Digital Ink Experiences.” In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. Denver Colorado USA: ACM, May 2017, pp. 3241–3253. ISBN: 978-1-4503-4655-9. DOI: [10.1145/3025453.3025716](https://doi.org/10.1145/3025453.3025716).
- [323] Stuart Ritchie. “The Big Idea: Should We Get Rid of the Scientific Paper?” In: *The Guardian* (Apr. 2022). ISSN: 0261-3077. (Visited on 05/14/2023).
- [324] Ankit Rohatgi. *Webplotdigitizer: Version 4.5*. 2021.
- [325] Hugo Romat, Nathalie Henry Riche, Ken Hinckley, Bongshin Lee, Caroline Appert, Emmanuel Pietriga, and Christopher Collins. “ActiveInk: (Th)Inking with Data.” In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. Glasgow Scotland Uk: ACM, May 2019, pp. 1–13. ISBN: 978-1-4503-5970-2. DOI: [10.1145/3290605.3300272](https://doi.org/10.1145/3290605.3300272).
- [326] Hugo Romat, Emmanuel Pietriga, Nathalie Henry-Riche, Ken Hinckley, and Caroline Appert. “SpaceInk: Making Space for In-Context Annotations.” In: *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. UIST ’19. New York, NY, USA: Association for Computing Machinery, Oct. 2019, pp. 871–882. ISBN: 978-1-4503-6816-2. DOI: [10.1145/3332165.3347934](https://doi.org/10.1145/3332165.3347934).

## BIBLIOGRAPHY

- [327] E. Rosten, R. Porter, and T. Drummond. “Faster and Better: A Machine Learning Approach to Corner Detection.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.1 (Jan. 2010), pp. 105–119. ISSN: 0162-8828. DOI: [10.1109/TPAMI.2008.275](https://doi.org/10.1109/TPAMI.2008.275).
- [328] Jeffrey N. Rouder, Paul L. Speckman, Dongchu Sun, Richard D. Morey, and Geoffrey Iverson. “Bayesian t Tests for Accepting and Rejecting the Null Hypothesis.” In: *Psychonomic Bulletin & Review* 16.2 (Apr. 2009), pp. 225–237. ISSN: 1531-5320. DOI: [10.3758/PBR.16.2.225](https://doi.org/10.3758/PBR.16.2.225). (Visited on 04/04/2023).
- [329] Quentin Roy, Simon Tangi Perrault, Katherine Fennedy, Thomas Pietrzak, and Anne Roudaut. “Understanding User Strategies When Touching Arbitrary Shaped Objects.” In: *Proceedings of the 23rd International Conference on Mobile Human-Computer Interaction*. MobileHCI '21. New York, NY, USA: Association for Computing Machinery, Sept. 2021, pp. 1–11. ISBN: 978-1-4503-8328-8. DOI: [10.1145/3447526.3472038](https://doi.org/10.1145/3447526.3472038). (Visited on 01/23/2023).
- [330] Quentin Roy, Futian Zhang, and Daniel Vogel. “Automation Accuracy Is Good, but High Controllability May Be Better.” In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. New York, NY, USA: Association for Computing Machinery, May 2019, pp. 1–8. ISBN: 978-1-4503-5970-2. DOI: [10.1145/3290605.3300750](https://doi.org/10.1145/3290605.3300750). (Visited on 05/30/2022).
- [331] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. “ORB: An Efficient Alternative to SIFT or SURF.” In: *Proceedings of the 2011 International Conference on Computer Vision*. ICCV '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 2564–2571. ISBN: 978-1-4577-1101-5. DOI: [10.1109/ICCV.2011.6126544](https://doi.org/10.1109/ICCV.2011.6126544). URL: <http://dx.doi.org/10.1109/ICCV.2011.6126544>.
- [332] Richard M. Ryan and Edward L. Deci. “Self-Determination Theory and the Facilitation of Intrinsic Motivation, Social Development, and Well-Being.” In: *American Psychologist* 55.1 (20000301), p. 68. ISSN: 1935-990X. DOI: [10.1037/0003-066X.55.1.68](https://doi.org/10.1037/0003-066X.55.1.68). (Visited on 06/20/2023).
- [333] Bahador Saket, Alex Endert, and Cagatay Demiralp. *Task-Based Effectiveness of Basic Visualizations*. Apr. 2018. arXiv: [1709.08546 \[cs\]](https://arxiv.org/abs/1709.08546).
- [334] Nazmus Saquib, Rubaiat Habib Kazi, Li-yi Wei, Gloria Mark, and Deb Roy. “Constructing Embodied Algebra by Sketching.” In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. CHI '21. New York, NY, USA: Association for Computing Machinery, May 2021, pp. 1–16. ISBN: 978-1-4503-8096-6. DOI: [10.1145/3411764.3445460](https://doi.org/10.1145/3411764.3445460).
- [335] Abhraneel Sarma, Alex Kale, Michael Jongho Moon, Nathan Taback, Fanny Chevalier, Jessica Hullman, and Matthew Kay. *Multiverse: Multiplexing Alternative Data Analyses in R Notebooks*. Preprint. Open Science Framework, Apr. 2021. DOI: [10.31219/osf.io/yfbwm](https://doi.org/10.31219/osf.io/yfbwm). (Visited on 04/05/2023).

- [336] Arvind Satyanarayan and Jeffrey Heer. “Authoring Narrative Visualizations with Ellipsis.” In: *Computer Graphics Forum* 33.3 (2014), pp. 361–370. ISSN: 1467-8659. DOI: [10.1111/cgf.12392](https://doi.org/10.1111/cgf.12392).
- [337] Arvind Satyanarayan and Jeffrey Heer. “Lyra: An Interactive Visualization Design Environment: Lyra: An Interactive Visualization Design Environment.” In: *Computer Graphics Forum* 33.3 (June 2014), pp. 351–360. ISSN: 01677055. DOI: [10.1111/cgf.12391](https://doi.org/10.1111/cgf.12391). (Visited on 06/15/2022).
- [338] Arvind Satyanarayan, Dominik Moritz, Kanit Wongsuphasawat, and Jeffrey Heer. “Vega-Lite: A Grammar of Interactive Graphics.” In: *IEEE Transactions on Visualization and Computer Graphics* 23.1 (Jan. 2017), pp. 341–350. ISSN: 1941-0506. DOI: [10.1109/TVCG.2016.2599030](https://doi.org/10.1109/TVCG.2016.2599030).
- [339] Eric Saund, David Fleet, Daniel Larner, and James Mahoney. “Perceptually-supported Image Editing of Text and Graphics.” In: *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology*. UIST ’03. Vancouver, Canada: ACM, 2003, pp. 183–192. ISBN: 1-58113-636-6. DOI: [10.1145/964696.964717](https://doi.org/10.1145/964696.964717). URL: <http://doi.acm.org/10.1145/964696.964717>.
- [340] Manolis Savva, Nicholas Kong, Arti Chhajta, Li Fei-Fei, Maneesh Agrawala, and Jeffrey Heer. “ReVision: Automated Classification, Analysis and Redesign of Chart Images.” In: *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*. UIST ’11. Santa Barbara, California, USA: Association for Computing Machinery, 2011, pp. 393–402. ISBN: 978-1-4503-0716-1. DOI: [10.1145/2047196.2047247](https://doi.org/10.1145/2047196.2047247). URL: <http://doi.acm.org/10.1145/2047196.2047247>.
- [341] Mike Scaife and Yvonne Rogers. “External Cognition: How Do Graphical Representations Work?” In: *International Journal of Human-Computer Studies* 45.2 (Aug. 1996), pp. 185–213. ISSN: 1071-5819. DOI: [10.1006/ijhc.1996.0048](https://doi.org/10.1006/ijhc.1996.0048).
- [342] Bill N. Schilit, Gene Golovchinsky, and Morgan N. Price. “Beyond Paper: Supporting Active Reading with Free Form Digital Ink Annotations.” In: *Proceedings of the 1998 CHI Conference on Human Factors in Computing Systems*. CHI ’98. USA: ACM Press/Addison-Wesley Publishing Co., Jan. 1998, pp. 249–256. ISBN: 978-0-201-30987-4. DOI: [10.1145/274644.274680](https://doi.org/10.1145/274644.274680).
- [343] Thomas R Schnell and Daniel J Rocchio. “A Comparison of Underlying Strategies for Improving Reading Comprehension and Retention.” In: (1979).
- [344] Wolfgang Schnotz and Maria Bannert. “Construction and Interference in Learning from Multiple Representation.” In: *Learning and Instruction*. External and Internal Representations in Multimedia Learning 13.2 (Apr. 2003), pp. 141–156. ISSN: 0959-4752. DOI: [10.1016/S0959-4752\(02\)00017-8](https://doi.org/10.1016/S0959-4752(02)00017-8).



## BIBLIOGRAPHY

- [345] Edward Segel and Jeffrey Heer. “Narrative Visualization: Telling Stories with Data.” In: *IEEE Transactions on Visualization and Computer Graphics* 16.6 (Nov. 2010), pp. 1139–1148. ISSN: 1941-0506. DOI: [10.1109/TVCG.2010.179](https://doi.org/10.1109/TVCG.2010.179).
- [346] Abigail Sellen and Richard Harper. *The Myth of the Paperless Office*. Feb. 2002.
- [347] Vidya Setlur, Sarah E. Battersby, Melanie Tory, Rich Gossweiler, and Angel X. Chang. “Eviza: A Natural Language Interface for Visual Analysis.” In: *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. UIST ’16. New York, NY, USA: Association for Computing Machinery, Oct. 2016, pp. 365–377. ISBN: 978-1-4503-4189-9. DOI: [10.1145/2984511.2984588](https://doi.org/10.1145/2984511.2984588).
- [348] Mingyan Shao and Robert P. Futrelle. “Recognition and Classification of Figures in PDF Documents.” In: *Graphics Recognition. Ten Years Review and Future Perspectives*. Ed. by Wenyin Liu and Josep Lladós. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 231–242. ISBN: 978-3-540-34712-5. DOI: [10.1007/11767978\\_21](https://doi.org/10.1007/11767978_21).
- [349] Leixian Shen, Enya Shen, Yuyu Luo, Xiaocong Yang, Xuming Hu, Xiongshuai Zhang, Zhiwei Tai, and Jianmin Wang. “Towards Natural Language Interfaces for Data Visualization: A Survey.” In: *IEEE Transactions on Visualization and Computer Graphics* 1 (2022), pp. 1–1. ISSN: 1077-2626, 1941-0506, 2160-9306. DOI: [10.1109/TVCG.2022.3148007](https://doi.org/10.1109/TVCG.2022.3148007). arXiv: [2109.03506 \[cs\]](https://arxiv.org/abs/2109.03506).
- [350] Roger N. Shepard. “Recognition Memory for Words, Sentences, and Pictures.” In: *Journal of Verbal Learning and Verbal Behavior* 6.1 (Feb. 1967), pp. 156–163. ISSN: 0022-5371. DOI: [10.1016/S0022-5371\(67\)80067-7](https://doi.org/10.1016/S0022-5371(67)80067-7). (Visited on 06/24/2023).
- [351] Mary D. Shepherd and Carla C. van de Sande. “Reading Mathematics for Understanding—From Novice to Expert.” In: *The Journal of Mathematical Behavior* 35 (Sept. 2014), pp. 74–86. ISSN: 0732-3123. DOI: [10.1016/j.jmathb.2014.06.003](https://doi.org/10.1016/j.jmathb.2014.06.003).
- [352] Kevin Sheppard et al. *bashtage/arch: Release 5.3.1*. Version v5.3.1. June 2022. DOI: [10.5281/zenodo.6684078](https://doi.org/10.5281/zenodo.6684078). URL: <https://doi.org/10.5281/zenodo.6684078>.
- [353] Danqing Shi, Xinyue Xu, Fuling Sun, Yang Shi, and Nan Cao. “Calliope: Automatic Visual Data Story Generation from a Spreadsheet.” In: *IEEE Transactions on Visualization and Computer Graphics* 27.2 (Feb. 2021), pp. 453–463. ISSN: 1941-0506. DOI: [10.1109/TVCG.2020.3030403](https://doi.org/10.1109/TVCG.2020.3030403).
- [354] Jianbo Shi and C. Tomasi. “Good features to track.” In: *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. USA: IEEE, June 1994, pp. 593–600. DOI: [10.1109/CVPR.1994.323794](https://doi.org/10.1109/CVPR.1994.323794).

- [355] Zbyněk Šidák. "Rectangular Confidence Regions for the Means of Multivariate Normal Distributions." In: *Journal of the American Statistical Association* 62.318 (June 1967), pp. 626–633. ISSN: 0162-1459. DOI: [10.1080/01621459.1967.10482935](https://doi.org/10.1080/01621459.1967.10482935). (Visited on 04/03/2023).
- [356] Kurex Sidik and Jeffrey N. Jonkman. "A Simple Confidence Interval for Meta-Analysis." In: *Statistics in Medicine* 21.21 (2002), pp. 3153–3159. ISSN: 1097-0258. DOI: [10.1002/sim.1262](https://doi.org/10.1002/sim.1262). (Visited on 04/05/2023).
- [357] Herbert A Simon. "On the Forms of Mental Representation." In: *University of Minnesota Press, Minneapolis* (1978), p. 16.
- [358] Lauren M. Singer and Patricia A. Alexander. "Reading on Paper and Digitally: What the Past Decades of Empirical Research Reveal." In: *Review of Educational Research* 87.6 (Dec. 2017), pp. 1007–1041. ISSN: 0034-6543. DOI: [10.3102/0034654317722961](https://doi.org/10.3102/0034654317722961).
- [359] Ian Spence. "Visual Psychophysics of Simple Graphical Elements." In: *Journal of Experimental Psychology: Human Perception and Performance* 16.4 (1990), pp. 683–692. ISSN: 1939-1277, 0096-1523. DOI: [10.1037/0096-1523.16.4.683](https://doi.org/10.1037/0096-1523.16.4.683).
- [360] Ian Spence and Stephan Lewandowsky. "Displaying Proportions and Percentages." In: *Applied Cognitive Psychology* 5.1 (1991), pp. 61–77. ISSN: 1099-0720. DOI: [10.1002/acp.2350050106](https://doi.org/10.1002/acp.2350050106).
- [361] Springer. *Conference Proceedings Guidelines | Springer*. <https://www.springer.com/gp/computer-science/lncs/conference-proceedings-guidelines>. 2022. (Visited on 06/14/2022).
- [362] Arjun Srinivasan, Steven M. Drucker, Alex Endert, and John Stasko. "Augmenting Visualizations with Interactive Data Facts to Facilitate Interpretation and Communication." In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (Jan. 2019), pp. 672–681. ISSN: 1077-2626, 1941-0506, 2160-9306. DOI: [10.1109/TVCG.2018.2865145](https://doi.org/10.1109/TVCG.2018.2865145).
- [363] Arjun Srinivasan, Nikhila Nyapathy, Bongshin Lee, Steven M. Drucker, and John Stasko. "Collecting and Characterizing Natural Language Utterances for Specifying Data Visualizations." In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. Yokohama Japan: ACM, May 2021, pp. 1–10. ISBN: 978-1-4503-8096-6. DOI: [10.1145/3411764.3445400](https://doi.org/10.1145/3411764.3445400).
- [364] UNESCO Institute for Statistics. *Literacy Rates Continue to Rise from One Generation to the next - UNESCO Digital Library*. <https://unesdoc.unesco.org/ark:/48223/pfo000258942?posInSet=1&queryId=826eb5c2-2660-4792-a3c8-83848b42df94>. 2017.
- [365] Sara Steegen, Francis Tuerlinckx, Andrew Gelman, and Wolf Vanpaemel. "Increasing Transparency Through a Multiverse Analysis." In: *Perspectives on Psychological Science* 11.5 (Sept. 2016), pp. 702–712. ISSN: 1745-6916. DOI: [10.1177/1745691616658637](https://doi.org/10.1177/1745691616658637). (Visited on 02/23/2023).

- [366] James H. Steiger. “Beyond the F Test: Effect Size Confidence Intervals and Tests of Close Fit in the Analysis of Variance and Contrast Analysis.” In: *Psychological Methods* 9.2 (20040510), p. 164. ISSN: 1939-1463. DOI: [10.1037/1082-989X.9.2.164](https://doi.org/10.1037/1082-989X.9.2.164). (Visited on 04/04/2023).
- [367] Charles D. Stolper, Bongshin Lee, Nathalie Henry Riche, and John T. Stasko. “Emerging and Recurring Data-Driven Storytelling Techniques: Analysis of a Curated Collection of Recent Stories.” In: 2016. URL: <https://api.semanticscholar.org/CorpusID:17894163>.
- [368] Michael Strickland, Archie Tse, Matthew Ericson, and Tom Giratikanon. *Archie Markup Language (ArchieML)*. <http://archieml.org/>. 2015.
- [369] Hendrik Strobelt, Daniela Oelke, Christian Rohrdantz, Andreas Stoffel, Daniel A. Keim, and Oliver Deussen. “Document Cards: A Top Trumps Visualization for Documents.” In: *IEEE Transactions on Visualization and Computer Graphics* 15.6 (Nov. 2009), pp. 1145–1152. ISSN: 1941-0506. DOI: [10.1109/TVCG.2009.139](https://doi.org/10.1109/TVCG.2009.139).
- [370] Hariharan Subramonyam, Colleen Seifert, Priti Shah, and Eytan Adar. “texSketch: Active Diagramming through Pen-and-Ink Annotations.” In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. Honolulu HI USA: ACM, Apr. 2020, pp. 1–13. ISBN: 978-1-4503-6708-0. DOI: [10.1145/3313831.3376155](https://doi.org/10.1145/3313831.3376155).
- [371] Sangho Suh, Martinet Lee, Gracie Xia, and Edith Law. “Coding Strip: A Pedagogical Tool for Teaching and Learning Programming Concepts through Comics.” In: *2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. Dunedin, New Zealand: IEEE, Aug. 2020, pp. 1–10. ISBN: 978-1-72816-901-9. DOI: [10.1109/VL/HCC50065.2020.9127262](https://doi.org/10.1109/VL/HCC50065.2020.9127262). (Visited on 06/25/2023).
- [372] Sangho Suh, Jian Zhao, and Edith Law. “CodeToon: Story Ideation, Auto Comic Generation, and Structure Mapping for Code-Driven Storytelling.” In: *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. Bend OR USA: ACM, Oct. 2022, pp. 1–16. ISBN: 978-1-4503-9320-1. DOI: [10.1145/3526113.3545617](https://doi.org/10.1145/3526113.3545617). (Visited on 06/25/2023).
- [373] Gail M. Sullivan and Richard Feinn. “Using Effect Size—or Why the P Value Is Not Enough.” In: *Journal of Graduate Medical Education* 4.3 (Sept. 2012), pp. 279–282. ISSN: 1949-8349. DOI: [10.4300/JGME-D-12-00156.1](https://doi.org/10.4300/JGME-D-12-00156.1). (Visited on 01/12/2023).
- [374] Nicole Sultanum, Fanny Chevalier, Zoya Bylinskii, and Zhicheng Liu. “Leveraging Text-Chart Links to Support Authoring of Data-Driven Articles with VizFlow.” In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. CHI '21. New York, NY, USA: Association for Computing Machinery, May 2021, pp. 1–17. ISBN: 978-1-4503-8096-6. DOI: [10.1145/3411764.3445354](https://doi.org/10.1145/3411764.3445354).

- [375] Kaiwen Sun, Carlo Sugatan, Tanisha Afnan, Hayley Simon, Susan A. Gelman, Jenny Radesky, and Florian Schaub. ““They See You’re a Girl If You Pick a Pink Robot with a Skirt”: A Qualitative Study of How Children Conceptualize Data Processing and Digital Privacy Risks.” In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. Yokohama Japan: ACM, May 2021, pp. 1–34. ISBN: 978-1-4503-8096-6. DOI: [10.1145/3411764.3445333](https://doi.org/10.1145/3411764.3445333).
- [376] Desney S. Tan, Brian Meyers, and Mary Czerwinski. “WinCuts: Manipulating Arbitrary Window Regions for More Effective Use of Screen Space.” In: *CHI '04 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '04. Vienna, Austria: ACM, 2004, pp. 1525–1528. ISBN: 1-58113-703-6. DOI: [10.1145/985921.986106](https://doi.org/10.1145/985921.986106). URL: <http://doi.acm.org/10.1145/985921.986106>.
- [377] Bo Tang, Shi Han, Man Lung Yiu, Rui Ding, and Dongmei Zhang. “Extracting Top-K Insights from Multi-dimensional Data.” In: *Proceedings of the 2017 ACM International Conference on Management of Data*. Chicago Illinois USA: ACM, May 2017, pp. 1509–1524. ISBN: 978-1-4503-4197-4. DOI: [10.1145/3035918.3035922](https://doi.org/10.1145/3035918.3035922).
- [378] Craig S. Tashman and W. Keith Edwards. “Active Reading and Its Discontents: The Situations, Problems and Ideas of Readers.” In: *Proceedings of the 2011 CHI Conference on Human Factors in Computing Systems*. CHI '11. New York, NY, USA: Association for Computing Machinery, May 2011, pp. 2927–2936. ISBN: 978-1-4503-0228-9. DOI: [10.1145/1978942.1979376](https://doi.org/10.1145/1978942.1979376).
- [379] Craig S. Tashman and W. Keith Edwards. “LiquidText: A Flexible, Multitouch Environment to Support Active Reading.” In: *Proceedings of the 2011 CHI Conference on Human Factors in Computing Systems*. Vancouver BC Canada: ACM, May 2011, pp. 3285–3294. ISBN: 978-1-4503-0228-9. DOI: [10.1145/1978942.1979430](https://doi.org/10.1145/1978942.1979430).
- [380] Carol Tenopir, Donald W. King, Lisa Christian, and Rachel Volentine. “Scholarly Article Seeking, Reading, and Use: A Continuing Evolution from Print to Electronic in the Sciences and Social Sciences.” In: *Learned Publishing* 28.2 (2015), pp. 93–105. ISSN: 1741-4857. DOI: [10.1087/20150203](https://doi.org/10.1087/20150203).
- [381] Carol Tenopir, Rachel Volentine, and Donald W King. “Scholarly Reading and the Value of Academic Library Collections: Results of a Study in Six UK Universities: Based on a Study Carried out for JISC Collections and Presented by Carol Tenopir at the 35th UKSG Conference, Glasgow, March 2012.” In: *Insights: the UKSG journal* 25.2 (July 2012), pp. 130–149. ISSN: 2048-7754. DOI: [10.1629/2048-7754.25.2.130](https://doi.org/10.1629/2048-7754.25.2.130).
- [382] Carol Tenopir, Concepción S. Wilson, Pertti Vakkari, Sanna Talja, and Donald W. King. “Cross Country Comparison of Scholarly E-Reading Patterns in Australia, Finland, and the United States.” In: *Australian*

## BIBLIOGRAPHY

- Academic & Research Libraries* 41.1 (Mar. 2010), pp. 26–41. ISSN: 0004-8623. DOI: [10.1080/00048623.2010.10721432](https://doi.org/10.1080/00048623.2010.10721432).
- [383] *The ACM Publishing System (TAPS) Best Practices*. <https://www.acm.org/publications/taps/taps-best-practices>. (Visited on 06/23/2023).
- [384] *Towards a Typology of Reading Goals*.
- [385] Edward Tufte. *Beautiful Evidence*. Cheshire, Conn: Graphics Press, Jan. 2006. ISBN: 978-1-930824-16-4.
- [386] John Tukey. *Exploratory Data Analysis*. 1st edition. Reading, Mass: Pearson, Jan. 1977. ISBN: 978-0-201-07616-5.
- [387] B. Tummers. *DataThief III*. <https://www.datathief.org/>. 2006. (Visited on 05/26/2022).
- [388] Raphael Vallat. “Pingouin: Statistics in Python.” In: *Journal of Open Source Software* 3.31 (Nov. 2018), p. 1026. ISSN: 2475-9066. DOI: [10.21105/joss.01026](https://doi.org/10.21105/joss.01026). (Visited on 04/03/2023).
- [389] Aditya Vashistha, Pooja Sethi, and Richard Anderson. “Respeak: A Voice-based, Crowd-powered Speech Transcription System.” In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI '17. New York, NY, USA: Association for Computing Machinery, May 2017, pp. 1855–1866. ISBN: 978-1-4503-4655-9. DOI: [10.1145/3025453.3025640](https://doi.org/10.1145/3025453.3025640). (Visited on 06/17/2022).
- [390] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. “Attention Is All You Need.” In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017. (Visited on 04/05/2023).
- [391] Victor. *Up and Down the Ladder of Abstraction*. <http://worrydream.com/LadderOfAbstraction/>. 2011. (Visited on 06/21/2023).
- [392] Bret Victor. *Tangle: A JavaScript Library for Reactive Documents*. <http://worrydream.com/Tangle/>. 2010.
- [393] Bret Victor. *Explorable Explanations*. 2011. URL: <http://worrydream.com/ExplorableExplanations/>.
- [394] Bret Victor. *Kill Math*. <http://worrydream.com/KillMath/>. 2011.
- [395] Bret Victor. “Humane Representation of Thought: A Trail Map for the 21st Century.” In: *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*. UIST '14. Honolulu, Hawaii, USA: ACM, 2014, pp. 699–699. ISBN: 978-1-4503-3069-5. DOI: [10.1145/2642918.2642920](https://doi.org/10.1145/2642918.2642920). URL: <http://doi.acm.org/10.1145/2642918.2642920>.
- [396] Pauli Virtanen et al. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python.” In: *Nature Methods* 17 (2020), pp. 261–272. DOI: [10.1038/s41592-019-0686-2](https://doi.org/10.1038/s41592-019-0686-2).

- [397] Jan B. Vornhagen, April Tyack, and Elisa D. Mekler. "Statistical Significance Testing at CHI PLAY: Challenges and Opportunities for More Transparency." In: *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*. Virtual Event Canada: ACM, Nov. 2020, pp. 4–18. ISBN: 978-1-4503-8074-4. DOI: [10.1145/3410404.3414229](https://doi.org/10.1145/3410404.3414229). (Visited on 03/23/2023).
- [398] Chat Wacharamanotham, Lukas Eisenring, Steve Haroz, and Florian Echtler. "Transparency of CHI Research Artifacts: Results of a Self-Reported Survey." In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. Honolulu, HI, USA: Association for Computing Machinery, 2020, 1–14. ISBN: 9781450367080. DOI: [10.1145/3313831.3376448](https://doi.org/10.1145/3313831.3376448). URL: <https://doi.org/10.1145/3313831.3376448>.
- [399] Chat Wacharamanotham, Krishna Subramanian, Sarah Theres Völkel, and Jan Borchers. "Statsplorer: Guiding Novices in Statistical Analysis." In: *Proceedings of the 2015 CHI Conference on Human Factors in Computing Systems*. CHI '15. New York, NY, USA: Association for Computing Machinery, Apr. 2015, pp. 2693–2702. ISBN: 978-1-4503-3145-6. DOI: [10.1145/2702123.2702347](https://doi.org/10.1145/2702123.2702347). (Visited on 03/31/2023).
- [400] Michelle Q. Wang Baldonado, Allison Woodruff, and Allan Kuchinsky. "Guidelines for Using Multiple Views in Information Visualization." In: *Proceedings of the Working Conference on Advanced Visual Interfaces*. AVI '00. Palermo, Italy: ACM, 2000, pp. 110–119. ISBN: 1-58113-252-2. DOI: [10.1145/345513.345271](https://doi.org/10.1145/345513.345271). URL: <http://doi.acm.org/10.1145/345513.345271>.
- [401] John Warnock. *The Camelot Project*. 1991.
- [402] J. N. Washburne. "An Experimental Study of Various Graphic, Tabular, and Textual Methods of Presenting Quantitative Material." In: *Journal of Educational Psychology* 18.7 (1927), pp. 465–476. ISSN: 1939-2176. DOI: [10.1037/h0070054](https://doi.org/10.1037/h0070054).
- [403] Alan J. Wecker, Joel Lanir, Osnat Mokryn, Einat Minkov, and Tsvi Kuflik. "Semantize: Visualizing the Sentiment of Individual Document." In: *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces - AVI '14*. Como, Italy: ACM Press, 2014, pp. 385–386. ISBN: 978-1-4503-2775-6. DOI: [10.1145/2598153.2600056](https://doi.org/10.1145/2598153.2600056).
- [404] M Whitehouse and M Rojanasakul. "Find Out If Your Job Will Be Automated." In: *Bloomberg.com* (July 2017). (Visited on 06/21/2023).
- [405] Wikipedia. *DTrace*. 2005. URL: <https://en.wikipedia.org/wiki/DTrace>.
- [406] Wikipedia. *EPUB*. 2007. URL: <https://en.wikipedia.org/wiki/EPUB>.
- [407] R. T. Wilkinson\* and Helen M. Robinshaw. "Proof-Reading: VDU and Paper Text Compared for Speed, Accuracy and Fatigue." In: *Behaviour & Information Technology* 6.2 (Apr. 1987), pp. 125–133. ISSN: 0144-929X. DOI: [10.1080/01449298708901822](https://doi.org/10.1080/01449298708901822).

- [408] Jacob O. Wobbrock, Leah Findlater, Darren Gergle, and James J. Higgins. "The Aligned Rank Transform for Nonparametric Factorial Analyses Using Only Anova Procedures." In: *Proceedings of the 2011 CHI Conference on Human Factors in Computing Systems*. CHI '11. New York, NY, USA: Association for Computing Machinery, May 2011, pp. 143–146. ISBN: 978-1-4503-0228-9. DOI: [10.1145/1978942.1978963](https://doi.org/10.1145/1978942.1978963). (Visited on 03/23/2023).
- [409] Jacob O. Wobbrock and Julie A. Kientz. "Research Contributions in Human-Computer Interaction." In: *Interactions* 23.3 (Apr. 2016), pp. 38–44. ISSN: 1072-5520, 1558-3449. DOI: [10.1145/2907069](https://doi.org/10.1145/2907069). (Visited on 06/22/2023).
- [410] Kanit Wongsuphasawat, Dominik Moritz, Anushka Anand, Jock Mackinlay, Bill Howe, and Jeffrey Heer. "Voyager: Exploratory Analysis via Faceted Browsing of Visualization Recommendations." In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (Jan. 2016), pp. 649–658. ISSN: 1941-0506. DOI: [10.1109/TVCG.2015.2467191](https://doi.org/10.1109/TVCG.2015.2467191).
- [411] P. Wright and A. Lickorish. "Proof-Reading Texts on Screen and Paper." In: *Behaviour & Information Technology* 2 (1983), pp. 227–235. ISSN: 1362-3001. DOI: [10.1080/01449298308914479](https://doi.org/10.1080/01449298308914479).
- [412] Keke Wu, Emma Petersen, Tahmina Ahmad, David Burlinson, Shea Tanis, and Danielle Albers Szafir. "Understanding Data Accessibility for People with Intellectual and Developmental Disabilities." In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. CHI '21. New York, NY, USA: Association for Computing Machinery, May 2021, pp. 1–16. ISBN: 978-1-4503-8096-6. DOI: [10.1145/3411764.3445743](https://doi.org/10.1145/3411764.3445743). (Visited on 09/08/2022).
- [413] Sally P. W. Wu and Martina A. Rau. "How Students Learn Content in Science, Technology, Engineering, and Mathematics (STEM) Through Drawing Activities." In: *Educational Psychology Review* 31.1 (Mar. 2019), pp. 87–120. ISSN: 1573-336X. DOI: [10.1007/s10648-019-09467-3](https://doi.org/10.1007/s10648-019-09467-3).
- [414] Qian Yang, Gerard de Melo, Yong Cheng, and Sen Wang. "HiText: Text Reading with Dynamic Saliency Marking." In: *Proceedings of the 26th International Conference on World Wide Web Companion*. WWW '17 Companion. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, Apr. 2017, pp. 311–319. ISBN: 978-1-4503-4914-7. DOI: [10.1145/3041021.3054168](https://doi.org/10.1145/3041021.3054168).
- [415] Ron Yeh, Chunyuan Liao, Scott Klemmer, François Guimbretière, Brian Lee, Boyko Kakaradov, Jeannie Stamberger, and Andreas Paepcke. "ButterflyNet: A Mobile Capture and Access System for Field Biology Research." In: *Proceedings of the 2006 CHI Conference on Human Factors in Computing Systems*. CHI '06. Montréal, Québec, Canada: ACM, 2006, pp. 571–580. ISBN: 1-59593-372-7. DOI: [10.1145/1124772.1124859](https://doi.org/10.1145/1124772.1124859). URL: <http://doi.acm.org/10.1145/1124772.1124859>.

- [416] Tom Yeh, Tsung-Hsiang Chang, and Robert C. Miller. "Sikuli: Using GUI Screenshots for Search and Automation." In: *Proceedings of the 22Nd Annual ACM Symposium on User Interface Software and Technology*. UIST '09. Victoria, BC, Canada: ACM, 2009, pp. 183–192. ISBN: 978-1-60558-745-5. DOI: [10.1145/1622176.1622213](https://doi.org/10.1145/1622176.1622213). URL: <http://doi.acm.org/10.1145/1622176.1622213>.
- [417] Ji Soo Yi, Youn ah Kang, John Stasko, and J.A. Jacko. "Toward a Deeper Understanding of the Role of Interaction in Information Visualization." In: *IEEE Transactions on Visualization and Computer Graphics* 13.6 (Nov. 2007), pp. 1224–1231. ISSN: 1941-0506. DOI: [10.1109/TVCG.2007.70515](https://doi.org/10.1109/TVCG.2007.70515).
- [418] Chun Yu, Yizheng Gu, Zhican Yang, Xin Yi, Hengliang Luo, and Yuanchun Shi. "Tap, Dwell or Gesture? Exploring Head-Based Text Entry Techniques for HMDs." In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI '17. New York, NY, USA: Association for Computing Machinery, May 2017, pp. 4479–4488. ISBN: 978-1-4503-4655-9. DOI: [10.1145/3025453.3025964](https://doi.org/10.1145/3025453.3025964). (Visited on 06/17/2022).
- [419] Luke S. Zettlemoyer and Robert St. Amant. "A Visual Medium for Programmatic Control of Interactive Applications." In: *Proceedings of the 1999 CHI Conference on Human Factors in Computing Systems*. CHI '99. Pittsburgh, Pennsylvania, USA: ACM, 1999, pp. 199–206. ISBN: 0-201-48559-1. DOI: [10.1145/302979.303039](https://doi.org/10.1145/302979.303039). URL: <http://doi.acm.org/10.1145/302979.303039>.
- [420] Qiyu Zhi, Alvitta Ottley, and Ronald Metoyer. "Linking and Layout: Exploring the Integration of Text and Visualization in Storytelling." In: *Computer Graphics Forum* 38.3 (2019), pp. 675–685. ISSN: 1467-8659. DOI: [10.1111/cgf.13719](https://doi.org/10.1111/cgf.13719).
- [421] Yaqian Zhu and John Kolassa. "Assessing and Comparing the Accuracy of Various Bootstrap Methods." In: *Communications in Statistics - Simulation and Computation* 47.8 (Sept. 2018), pp. 2436–2453. ISSN: 0361-0918. DOI: [10.1080/03610918.2017.1348516](https://doi.org/10.1080/03610918.2017.1348516). (Visited on 03/07/2023).
- [422] die.net. *XGetImage*. 1987. URL: <https://linux.die.net/man/3/xgetimage>.



## APPENDICES

## EXTRACTION ALGORITHMS USED IN CHARTDETECTIVE

All the extraction algorithms work by taking as input a user selection (i.e., a list of shapes) and outputting an array of coordinates that should be added to the extracted data. A shape is defined by a list of points (corners). To give consistent specifications to extractors, all shapes are pre-processed to be subdivided into smaller units everytime they use the “moveto” feature (C2), see Algorithm 7. For clarity reasons, the pseudo-code focuses on one specific orientation: vertical bars, box plots, and axes and horizontal lines.

## A.1 EXTRACTION OF BOX PLOTS

**Algorithm 2:** Extraction of box plots

---

**Input:** A user selection of shapes *shapes*  
**Output:** An array of 2D coordinates

```

points ← []
groups ← group shapes with equal horizontal positions
foreach group ∈ groups do
  min ← y of group[0]
  max ← top of group[0]
  q1 ← min
  q3 ← max
  medians ← []
  foreach shape ∈ group do
    if shape is horizontal line then
      | add vertical centre of shape to medians
    if shape is rectangle then
      | q1 ← y of shape
      | q3 ← top of shape
  foreach median ∈ medians do
    if median > q1 and median < q3 then
      | add (middle of group, median) to points
  add (middle of group, min) to points
  add (middle of group, max) to points
  add (middle of group, q1) to points
  add (middle of group, q3) to points
return points

```

---

## A.2 EXTRACTION OF BARS

---

**Algorithm 3:** Extraction of bars

---

**Input:** A user selection of shapes *shapes***Output:** An array of 2D coordinates*points*  $\leftarrow$  []**foreach**  $s \in shapes$  **do**  | add (middle of  $s$ , top of  $s$ ) to *points***return** *points*

---

## A.3 EXTRACTION OF SCATTER PLOTS

---

**Algorithm 4:** Extraction of scatters

---

**Input:** A user selection of shapes *shapes***Output:** An array of 2D coordinates*points*  $\leftarrow$  []**foreach**  $s \in shapes$  **do**  | add centre of  $s$  to *points***return** *points*

---

## A.4 EXTRACTION OF AXES

---

**Algorithm 5:** Extraction of axis

---

**Input:** A user selection of shapes *shapes***Output:** An array of 2D coordinates*points*  $\leftarrow$  []**foreach**  $s \in shapes$  **do**  | **if**  $s$  is text **then**    | add (text of  $s$ , centre of  $s$ ) to *points***return** *points*

---

## A.5 EXTRACTION OF LINES

---

**Algorithm 6:** Extraction of lines

---

**Input:** A user selection of shapes *shapes***Output:** An array of 2D coordinates*points*  $\leftarrow$  []**foreach**  $s \in shapes$  **do**  | **foreach**  $pt \in s$  **do**    | add  $pt$  to *points***return** *points*

---

## A.6 SPLITTING COMPOUNDED SHAPES

---

**Algorithm 7:** Split a compound shape

---

**Input:** A shape  $s$ **Output:** An array of shapes $shapes \leftarrow []$  $path \leftarrow []$ **foreach**  $ope \in$  path of  $s$  **do**    **if**  $ope$  is moveto **then**        add shape formed from  $path$  to  $shapes$          $path \leftarrow []$     add  $ope$  to  $path$ **return**  $shapes$ 

---

## GLOSSARY

---

- active reading** act of annotating a document and interacting with its content.. 8, 9, 22, 37
- cognitive load theory** a theory that suggests working memory has limited capacity and that the cognitive load affects the learning process.. 2, 9
- dual-coding theory** a theory that suggests the human mind processes and represents information using a verbal and visual system.. 2, 10, 11, 118
- extraneous load** refers to the added complexity due to the presentation of the information rather than the information itself.. 9, 24, 41
- extrinsic motivation** motivation due to the task leading to some external gain.. 9
- intrinsic load** refers to the inherent complexity of the information.. 9, 11
- intrinsic motivation** motivation due to the task being personally rewarding.. 9
- ladder of abstraction** a metaphorical ladder to represent the different levels of specificity (bottom of the ladder) or generality (top of the ladder) when describing concepts.. 2, 11, 15, 118
- media richness theory** a theory to describe how different media vary in their ability to convey information effectively.. 1, 9, 94, 117, 118
- multimedia principle** principle explaining that people learn better from text and images combined than from text alone.. 2, 10, 15, 17
- personalized learning** efforts to tailor learning materials and education to meet the needs of individual students.. 2, 9
- self-determination theory** a theory of human motivation that identifies autonomy, competence, and relatedness as needs to become self-determined.. 1, 9