Spanning the Boundaries of Data Visualization Work: An Exploration of Functional Affordances and Disciplinary Values

Jaime Snyder[1] and Katie Shilton[2]

[1] University of Washington, Seattle WA 98195, USA
jas1208@uw.edu
[2] University of Maryland, College Park MD 20742, USA
kshilton@umd.edu

Abstract. Creating data visualizations requires diverse skills including computer programming, statistics, and graphic design. Visualization practitioners, often formally trained in one but not all of these areas, increasingly face the challenge of reconciling, integrating and prioritizing competing disciplinary values, norms and priorities. To inform multidisciplinary visualization pedagogy, we analyze the negotiation of values in the rhetoric and affordances of two common tools for creating visual representations of data: R and Adobe Illustrator. Features of, and discourse around, these standard visualization tools illustrate both a convergence of values and priorities (clear, attractive, and communicative data-driven graphics) side-by-side with a retention of rhetorical divisions between disciplinary communities (statistical analysis in contrast to creative expression). We discuss implications for data-driven work and data science curricula within the current environment where data visualization *practice* is converging while values in *rhetoric* remain divided.

Keywords: Data visualization; Data science practice; Digital tool analysis; Materiality; Values.

1 Introduction

Creating visualizations to help humans understand complex data requires skills ranging from coding to statistics to graphic design. However, data visualization has traditionally drawn sharp disciplinary and professional boundaries between graphic and information *design* practitioners, typically trained in the arts and visual communication, and information *visualization* researchers, typically trained in computer and data science [1]. As Kosara points out,

> "Two cultures exist in visualization: very technical, analysis-oriented work in one, and artistic pieces on the other hand. … pragmatic visualization is mostly practiced by people in computer science with no background in art or design. A student once put it aptly: 'Computer graphics is mostly computers, but little graphics.' At the same

> time, artists and designers often work on visualizations without much knowledge of the technical work being done in computer science" [2, p. 631].

However, online tutorials and commentaries focused on data visualization suggest that, in practice, the boundaries of what are considered "core" visualization skills and tools are blurry [e.g., 3–6]. Further, industry and academic conferences such as OpenVis [7] and IEEE Vis [8] attract graphic artists and designers as well as computer scientists.

Attempts to delineate the skills, and practices of data visualization are especially complicated in the classroom. University curricula often echo siloed disciplinary boundaries, with departmental prerequisites, class size limits, and advising resources conspiring to keep graphics students in Departments of Art and Design and programming students in Departments of Computer Science. Because iSchools attract students with interests in both technical and design areas, our programs are in a position to lead more comprehensive data visualization training programs [9].

In order to probe implications for boundary-spanning visualization education, we performed an initial high-level analysis of two data visualization tools with strong connections to different communities of practice: R and Adobe Illustrator. We intentionally chose to begin with an analysis of tools for two reasons. First, we will be engaging directly with visualization designers in future work; there is such diversity among this community of practitioners that we felt it was advantageous to begin with a relatively straightforward analysis of functionality and technical specifications of common tools in order to inform decisions about how best to pursue more user-focused observations and interviews. Second, decisions regarding which software applications will be used in a course are often a strong influence on how the practice of data visualization will be taught. We wanted to understand these impacts at a high level before delving into other pedagogical influences in future work.

There are a host of tools currently available to generate data visualizations. Platforms focused on intelligence and other forms of business analytics, such as Power BI and Tableau, provide sophisticated dashboards for users to explore and annotate large datasets. These enterprise-level tools are typically web-based, carry expensive licensing agreements, and are often integrated in established work flows in large organizations. On the other end of the spectrum are open source tools, such as the very popular D3.js web-based platform, with ever-expanding sets of libraries, very active online user groups, and steep learning curves.

According to a series of recent surveys, both R and Adobe Illustrator are among the top five data visualization software applications used by professionals [10, 11]. R is an open source programming language and software environment designed to support complex computational statistics [12]. Adobe Illustrator was created as a tool for designers and artists to harness emerging digital frameworks to make precise graphic assets. These tools were originally designed to serve different communities, but our analysis shows that their functionality is moving towards becoming more similar , adapting to support a broadening range of practices associated with data visualization. Our analysis provides 1) examples of what it looks like when data practices *converge* from multiple work traditions and community values, and 2) implications for the boundary spanning curricular work and research that characterizes many data science programs fostered by iSchools.

2 Background: The Data Visualization Process

Phrases like "data visualization," "information visualization," and "information design" are often used interchangeably. In this paper, we use *data visualization* to refer to the process of conceptualizing, designing, and building visual representations of data. Data visualization is practiced by a wide range of people, from data scientists to graphic designers, and generates artifacts as diverse as information graphics, interactive maps, and dynamic charts. In this sense, data visualization provides an umbrella term for more specialized disciplines. For example, *information visualization*, in its most precise definition, refers to the creation of interactive systems for representing abstract data (i.e., data that has no inherent spatial relations, unlike GIS or data derived from observations of physical phenomena) [13]. Information visualization systems are often built by computer scientists, data scientists and statisticians striving for objective and comprehensive representations of data for the purpose of analysis [14]. Information visualization training is often focused on quantitative methods and algorithms. The field's orientation to visual perception is typically cognitive rather than aesthetic. *Information design*, on the other hand, is frequently used to describe the work of graphic designers, information architects, and user experience specialists who create information-driven visual artifacts that help people communicate and explain things [15]. Information design artifacts include wayfinding signs, user interfaces for web and mobile applications, and information graphics. Practitioners often have formal training in visual communication and graphic design, and they tend to prioritize clarity of presentation and aesthetics. Technology skills in this community are more frequently self-taught [16]. In spite of similar sounding names, the differences between these approaches to data visualization are important. These groupings are the basis for professional and philosophical identities, and they reflect values differences that we discuss in more depth below.

Despite these distinctions, a generalized data visualization design process is shared by those who create visual representations of data. Ben Fry [17], one of the creators of Processing, a computer programming environment created for artists and designers which has become a popular tool for data art projects, outlines seven stages of the data visualization process [17, p. 5]:

- *Acquire*: Obtain data
- *Parse*: Structure data's meaning
- *Filter*: Remove all but core data
- *Mine*: Find patterns
- *Represent*: Choose a visual model
- *Refine*: Improve representation
- *Interact*: Add methods for manipulating or interacting with data

These steps echo other descriptions of the visualization process [cf.: 13, 14, 18] and provide a vocabulary to analyze R and Illustrator: tools designed to serve information visualization and information design communities, respectively. The values associated with these tools, and functionality built to support those values, reflect the concerns and practices of the people who design and use them. The following section describes how we decomposed these tools to interpret their supported values and practices.

3 Method

In order to better understand connections among technology, practices, and values, software applications and tools can be decomposed to identify functional elements and affordances, followed by consideration of what practices those affordances support [19–21]. We first performed a comparison of R and Adobe Illustrator as products by juxtaposing their respective audiences, support structures, licensing, purpose, and extensibility. Next, we unpacked the data visualization tools in each package, comparing graphic functionality, how data is entered or imported, the graphic outputs available, design patterns, features for manipulation of aesthetics and layout, and features for optimization. To trace change over time, we created a timeline of features and functionality for each of the tools. The timeline drew from current and historical documents, including archived web pages, that provided detailed descriptions of software features and functionality. These included software documentation (e.g., release notes, user guides), training materials (online tutorials, videos, demos), and marketing and media materials (e.g., new feature announcements, advertisements, reviews, blog posts). Finally, we examined public discourse around each product's visualization features, as well as the current state of both tools as evidenced by materials on their websites. To present our analysis, we begin with descriptions of R and Illustrator as products (Appendix: Table 1[1]), then describe the chronology of feature updates most relevant to their capacity to support data visualization (Appendix: Table 2). We close with a discussion of public discourse and rhetoric surrounding each product.

4 Findings

4.1 Adobe Illustrator

Adobe Illustrator, a core component of the Adobe Creative Suite, is a vector-based graphics tool. Adobe, Inc. was founded in 1982, and Adobe Illustrator was introduced in 1986 as a digital tool for graphic and typographic design. According to an early marketing video for Adobe Illustrator 88:

> "Now anyone can create high quality graphics and illustrations quickly and easily because today there's a revolution taking place in graphic design and production, a revolution born of computers and laser technology but with its roots in the tradition of quality and creativity, a revolution based on new tools, tools which free the imagination and eliminate drudgery" [22].

As a vector-based system, Illustrator renders line art using mathematical calculations called Bezier curves, generating smoothly arcing lines that can be scaled to any size without degradation or loss of sharpness. The scalability offered by Illustrator's underlying mathematics catalyzed the accessibility of digitized font and the evolution of typography in the digital age [23]. For decades, Adobe products have remained industry standard despite expensive annual licensing fees. Currently, the Adobe collection of

[1] See appendix at https://evidlab.umd.edu/2019/02/**iconference-2019-appendix**/

software applications is bundled as a cloud-based suite of tools with updates delivered automatically to users in return for a monthly subscription fee.

4.2 Adobe Illustrator's data visualization arc

When Adobe Illustrator launched in 1986, its primary purpose was to give users creative and artistic control of vector-based graphic output. Illustrator's ability to scale an image without quality loss provided significant benefits over raster-based images built from pixels (such as the jpeg file format) that pixelated and degraded when enlarged.

Illustrator has gradually adapted to support elements of Fry's data visualization process, particularly *represent* and *refine*. The earliest functionality directly related to data-driven graph creation was a chart-building tool incorporated into Illustrator 3.0 in 1990 [24]. Originally this tool included a spreadsheet-like interface for adding a dataset and a set of six chart templates to represent that data (grouped column, stacked column, line chart, pie chart, area graph and scatter plot). Three additional chart templates were added around 1997 (bar graph, stacked bar graph, and radar graph) [25]. To create a visualization, users entered their data, made adjustments as needed through the spreadsheet interface, and selected a chart template using a graphical user interface (GUI) tool window. Illustrator then generated a vector image. That image could be refined using Illustrator's extensive graphic design tools, including changing color, typography, replacing the shapes of plot points, or altering scale or orientation.

Today, users can use data to create and customize nine different types of graphs [26]. Using the Graph Data window, users can enter data into a spreadsheet-like interface by hand, copy and paste data from a spreadsheet application such as Microsoft Excel, or import data in a tab delineated file format. The resulting editable, vector-based graph is generated within the document art board and can be manipulated using the full suite of editing tools native to Illustrator.

Historically, one of the primary limitations of using the Illustrator chart-building tool was that once a chart is generated, connections with underlying data are lost. In exchange for the ability to manually make creative and artistic refinements to a chart, designers had to give up a degree of automaticity. However, with Illustrator 8.0, launched in 1998, the introduction of action scripting enabled users to routinize specific tasks using a set of common scripting languages. This change automated at least some parts of the tedious process of updating charts when data changed. The motivation for adding this functionality came from Adobe's primary audience of digital publishing designers [23]. However, Illustrator was still marketed primarily as a generalized graphics application as indicated by the Adobe Creative Suite reference book:

> "…the purpose of a chart or graph is clear communication of numeric information as a visual aid. A properly selected chart design will accomplish this. If you create a lot of charts or graphs, look into a specialty graphics application" [27].

In 2015, Adobe made further steps to support dynamic data—steps towards Fry's *interact* stage of visualization—with Creative Cloud Charts [28]. Creative Cloud Charts enabled a constant connection between streaming data and Illustrator visualizations. Adobe introduced this feature as a "Technology Preview": experimental, non-permanent beta functionality enabled by cloud-based software updates. Just a few months

after making the Creative Cloud Charts preview available, Adobe announced that it would "pause" development based on user feedback [29]. However, its temporary inclusion reflected growing interest in data-driven graphic design tools and hinted at the challenges of integrating robust graphics and dynamic data functionality into a single tool. Most recently, Adobe researchers, arguing that "graphic designers approach visualization authoring differently from computer scientists," have prototyped the new (not yet available) Data Illustrator, tailored specifically for data visualization [30].

4.3 R

The open source statistical software R, a GNU project, first appeared in 1993 [12]. R was developed to support both statistical computing and graphics. In addition to a command line interface, freely available development environments such as RStudio enable users to interact with the R environment through a GUI. Data visualizations such as scatterplots and bar charts can be exported as PDFs and viewed on the web or printed at any scale. By making adjustments to default parameters included in the R library, users can customize graphic elements including labels, colors, size, transparency and orientation. Users can also add legends and other explanatory text directly through commands in the R code, rather than adding these details using editing software after the plot has been generated. R software is currently supported by the R Foundation for Statistical Computing and is freely available for download and development by users, primarily through the creation of libraries or packages. Because of its open licensing, its extensibility, and the ability for users to control and generate print-quality graphic output, R has become an increasingly popular choice with statisticians, data scientists and others for performing quantitative analysis, data mining and visualization [12].

4.4 R's data visualization arc

Early in its development, data visualization in R benefitted from the introduction of the grid package, created in 2001 to aid in the creation of graphical output. The grid package marked a transition from the original lattice package that provided users with an array of options for displaying multivariate data [31]. Grid built on lattice by providing an expanded set of low-level commands to control the appearance and arrangement of vector-based graphic output [32]. For example, grid added the functionality to open a view window and to position elements within that space. The base graphics in R rely on the templates included with the grid package, enabling users to generate approximately seven different types of standardized charts including density plots, dot plots, bar charts, line charts, pie charts, boxplots and scatter plots [33].

Grid also provided the framework necessary for ggplot2, launched in 2005 and currently the second most popular R package [34]. ggpplot2 retains many of the same features and functionality as R's base graphics [35], and also extends R's capabilities for *representing* and *refining* visualizations through approximately 20 additional functions for generating data visualizations (or plots, as they are called in R) [36]. One of the key advantages to using ggplot2 is the method by which vector-based graphics are layered when exported. Rather than grouping graphic elements (such as plot points, axes, fit

lines, labels, titles, legends, etc.) arbitrarily or in an order determined by the sequence in which code is compiled, ggplot2 writes visual elements into semantic components based on Leland Wilkinson's *Grammar of Graphics*, a structural framework for identifying the meaning-making components of a graphic [32]. When ggplot2 writes the graphic output file (typically a PDF), related elements are on the same or adjacent layers (ie., axis labels and tick marks are grouped together, grid lines are separate from plot points). Because adjustments to the aesthetics and layout of a graphic are conducted by changing parameters of specific elements in the code, the grouping of elements according to semantic (rather than algorithmic) proximity makes it easier to refine graphics in ggplot2 with minimal additional code. This enables users to add, remove or change visual components of a graphic more easily [37].

While ggplot2 remains a highly popular package for R users, a series of new web-based data visualization tools have been introduced in recent years, such as D3.js and Tableau. The value of these newer tools lies in their ability to create and display visual representations of streaming data through a web browser. D3.js, in particular, supports streaming data through its JavaScript-based open source framework, making it highly accessible and flexible. However, much like the tradeoffs made by users of Illustrator, these tools do not do everything. While D3.js allows for highly nuanced refinement of graphic output, it does not have the analytic power of R.

In response to tools like D3.js, the Shiny package was introduced to the R suite of resources in 2012. Shiny provides a two-way interface between a web-based database and a dynamic webpage [38]. Using Shiny, a data visualization can be updated as its database changes, and, conversely, changes made to a visualization can be reflected in the database in real time. As a result, R can now interact with D3.js and visualization designers benefit from features of each to present a more fluid, refined and responsive representation of data.

R, combined with its extensions through ggplot and Shiny, emphasizes Fry's *parse* and *mine* stages of visualization, while also supporting *represent*, *refine*, and *interact*. Its built-in templates and structural frameworks illustrate how it relies on systematic rules for the structured representation and standardized output of data.

4.5 Converging material practices, distinct rhetoric

As noted above, R and Illustrator have evolved from different origins; however, growing connectivity, cloud-based computing, and data-driven work practices have influenced both applications. Both tools enable users to import or create a dataset within the application, then use that dataset to represent and refine a visualization. Each application facilitates parsing and representing data by providing users with a set of standardized design patterns to generate data visualizations. Both tools have also introduced recent functionality for interaction, enabling visualizations to respond dynamically to changes in the underlying dataset. And documentation for each tool boasts of high quality, professional graphic output. As a blogger writes of Illustrator:

> "Illustrator is pretty much the only [graphic design] application I know that can build live graphs so the numbers actually are well represented in proportion in your graph and can make them press ready and … really beautiful" [39].

R's website claims:

> "One of R's strengths is the ease with which well-designed publication-quality plots can be produced ... Great care has been taken over the defaults for the minor design choices in graphics, but the user retains full control" [12].

4.6 Language, values, and communities of practice

Closer inspection of the iterative development of data visualization features over time in both R and Illustrator shows a steady convergence of the practices supported by each application, as the tools' affordances have expanded to address a majority of the data visualization process outlined by Fry and others [13, 14, 17] (Appendix: Table 3). Although the functionality of these tools has converged over time, each has retained a strong identity within distinct communities of practice. The iterative alignment of each tool's material practices around the data visualization process (shown in Appendix: Table 3) contrasts with the distinct language used to talk about these tools in documentation, marketing materials, and popular media. As a tutorial for working with data in Illustrator asks:

> "What if you want to make a graphic for a publication or a presentation that's polished and fully customized? Adobe Illustrator gives you the control you need to do this...It's not graphing software. It's illustration software, but once you get the hang of things, Adobe Illustrator can be a valuable tool in your visualization arsenal" [6].

In contrast, as an R-Bloggers tutorial explains, R's output has been optimized for analytic tasks rather than communication:

> "Even for beginners, it's pretty easy to get out a basic plot to learn something about a data set. The trouble starts, however, when you want to show the plot to someone else. And, that is not the documentation's "fault". The documentation is all there. Typing `help(package="graphics")` at the command line will put most everything you need to know about base graphics at your reach. However, I think that until a person gets used to thinking of R as a system of interacting functions it takes a bit of ingenuity for a beginner to figure out how to accomplish a basic "functional" task like produce a plot that you can show around" [40].

The rhetoric used to describe these tools on their respective support websites also reflects their intended target audiences. R is described as primarily statistical, extensible, and having formulaic defaults:

> "R provides a wide variety of statistical (linear and nonlinear modeling, classical statistical tests, time-series analysis, classification, clustering, ….) and graphical techniques, and is highly extensible…One of R's strengths is the ease with which well-designed publication-quality plots can be produced, including mathematical symbols and formulae where needed. Great care has been taken over the defaults for the minor design choices in graphics, but the user retains full control" [12].

Illustrator evokes aesthetics and creative expression:

> "Artists use Illustrator for creating clean visual compositions that can be scaled infinitely without losing quality, creating free hand drawings in illustrations, tracing and recoloring scanned in artwork and also creating wireframes from which to create digital paintings" [41].

Illustrator's tools for customization are described in terms of values associated with creativity, such as flexibility and self-expression. For example, Illustrator's help documents reiterate the many ways you can change the template options:

> "For example, you can change the appearance and position of the graph's axes, add drop shadows, move the legend, and combine different graph types… You can also manually customize your graph in numerous ways. You can change the colors of shading; change the typeface and type style; move, reflect, shear, rotate, or scale any or all parts of the graph; and customize column and marker designs. You can apply transparency, gradients, blends, brush strokes, graphic styles, and other effects to graphs" [42].

The rhetoric around customization in statements like this is much more strongly associated with subjective expression than with rule-based objectivity. Attention to appearance and position, shading, typeface, and substantial customization features all support values of flexibility and the pursuit of craft. Illustrator's long history of supporting customization through its vector-based framework and explicit marketing to artists has led to strong associations with values of creativity, flexibility, and aesthetics.

Even as R packages have been introduced that enable increasingly refined representation and refinement of data visualizations (such as ggplot2), the grid framework at the heart of the platform anchors these packages in standardized (rather than customized) methods for image construction. As the *R for Data Science* online tutorial describes: "ggplot2 implements the grammar of graphics, a coherent system for describing and building graphs" [43]. While it is possible to make adjustments to individual graphic elements, defaults built into the package are emphasized in practice: "Once you set an aesthetic, ggplot2 takes care of the rest. It selects a pleasing set of levels to use for the aesthetic, and it constructs a legend that explains the mapping" [43]. Although R does allow for customization, it assumes that users will value the systematic application of rules over the need to make creative decisions themselves: "In practice, you rarely need to supply all seven parameters because ggplot2 will provide useful defaults for everything except the data, the mappings, and the geom function" [43]. Discourse around R strongly evokes values of standardization, automation and quantification, contributing to an analytic identity for the data visualization features of the R software.

5 Discussion

Comparing values evident in both the functionality of and the discourse surrounding R and Illustrator reveals a gap between converging material practices and divergent rhetoric. In previous work, we have explored the many ways researchers can observe values in practices, design, and language [44], and this study illustrates the importance of examining all three. While R and Illustrator currently lack *rhetorical values convergence*, we have seen a *convergence of values in design and practice*. Major recent developments in the visualization trajectories of each tool (Creative Cloud Charts for Illustrator and Shiny for R) are observable indicators of values convergence in practice among traditionally disconnected communities. Data-driven work combined with increasing

emphasis on sharing and publication act as a catalyst for this convergence, spurred by the expansion of data-driven work practices.

We argue that this indicates that data visualization is increasingly *both* information visualization and information design. Practitioners are blending traditionally separate values as they balance standardized visual representation techniques for analysis with the creativity expected of graphical communication. In response, developers are adapting their tools to support this convergence. R has added features that enable users to consider aesthetics when making data visually clear and compelling. At the same time, Illustrator has added support for designing with data, acknowledging that an increasing number of artists and graphic designers work with quantitative source materials. As data-driven work expands beyond practices associated primarily with statistics or computer science, values associated with quantification will be brought to creative practice, and creativity and expression become values in data science.

6 Conclusion

Research has demonstrated that values convergence facilitates group work and collaboration [45, 46]. Shared values encourage trust between parties and smooth technology acceptance [45, 47]. Convergence of values within the domain of data visualization signals an opportunity for iSchools to play an important role in facilitating the productive and constructive merging of disciplinary expertise. Our analysis of two tools from previously distinct communities of practice indicates that data-driven work may encourage new kinds of collaborations, as artists and analysts find themselves sharing interests, tools, and practices, if not yet language to describe this work.

However, boundary spanning of this nature requires scaffolding. To date, little research has examined situations where values in *practice* are converging while values in *rhetoric* remain divided. Within the domain of data visualization, a convergence of values across practitioner groups has the potential to create opportunities for knowledge sharing, but also carries the threat of backlash, provoking divisive discourse among disciplines as stakeholders perceive a need to protect their positions of expertise. Gatherings such as OpenVis and recent workshops being offered through the IEEE Vis conference [i.e., 48] encourage convergence by providing opportunities for cross-pollination and the creation of new professional identities for those embracing the converged values of quantification and creative expression. However, higher education institutional structures (e.g., departments, curricular requirements, and hiring practices) more typically reflect the siloed values of design and computer science. It remains to be seen, however, how data-driven workers who must navigate values in both practice and rhetoric will balance these tensions to engage in productive collaboration.

This research prompts us to turn attention to the educational implications of values and practice convergence paired with rhetorical divergence. As we develop data science curricula, we have a unique opportunity to introduce new pedagogical approaches to bridging technical and communicative facets of many of the professions for which we are training our students. We believe that further work in this area can expand our understanding of other emerging and converging multidisciplinary data practices.

References

1. Ziemkiewicz C, Kosara R (2009) Embedding information visualization within visual representation. In: Advances in Information and Intelligent Systems. Springer Berlin, Heidelberg, pp 307–326.
2. Kosara R (2007) Visualization Criticism: The Missing Link Between Information Visualization and Art. In: Proc IEEE Info Vis. IEEE, pp 631–636.
3. Garrett J (2014) How to Use Adobe Illustrator Variable Data with XML. In: Hypertransitory.com. http://hypertransitory.com/blog/2014/05/27/use-adobe-illustrator-variable-data-xml/. Accessed 12 Dec 2018.
4. Gordon M (2013) Exporting nice plots in R. In: R-Bloggers. http://www.r-bloggers.com/exporting-nice-plots-in-r/. Accessed 12 Dec 2018.
5. Smith D (2009) 10 tips for making your R graphics look their best. In: Revolutions. http://blog.revolutionanalytics.com/2009/01/10-tips-for-making-your-r-graphics-look-their-best.html. Accessed 12 Dec 2018.
6. Yau N (2008) How to Make a Graph in Adobe Illustrator. In: FlowingData. https://flowingdata.com/2008/12/16/how-to-make-a-graph-in-adobe-illustrator/. Accessed 12 Dec 2018.
7. OpenVisConf OpenVisConf. http://openvisconf.com. Accessed 12 Dec 2018.
8. IEEE VIS Conference. http://ieeevis.org/year/2018/welcome. Accessed 24 Aug 2018.
9. Hemsley J, Snyder J, Cottam J, Fisher B, Lemieux V, Stanton J, Wang Y (2015) Visualization Pedagogy in iSchools. Proceedings of the iConference 2015.
10. Meeks E (2018) 2018 Data Visualization Survey Results. In: Elijah Meeks. https://medium.com/@Elijah_Meeks/2018-data-visualization-survey-results-26a90856476b. Accessed 24 Aug 2018.
11. Meeks E (2017) 2017 Data Visualization Survey Results. In: Elijah Meeks. https://medium.com/@Elijah_Meeks/2017-data-visualization-survey-results-40688830b9f2. Accessed 24 Aug 2018.
12. R: What is R? https://www.r-project.org/about.html. Accessed 12 Dec 2018.
13. Card SK, Mackinlay JD, Shneiderman B (1999) Readings in Information Visualization: Using Vision to Think. Morgan Kaufmann, San Francisco, CA.
14. Few S (2009) Now You See It: Simple Visualization Techniques for Quantitative Analysis, 1st ed. Analytics Press, USA.
15. Jacobson R Information design. MIT Press, Cambridge, MA.
16. Yau N (2012) Visualize This! John Wiley & Sons.
17. Fry B (2008) Visualizing Data: Exploring and explaining data with the Processing Environment. O'Reilly Media, Sebastopol, CA.
18. Tufte ER (1990) Envisioning Information. Graphics Press, Cheshire, CT.
19. Brey P (2000) Disclosive Computer Ethics. SIGCAS Comput Soc 30:10–16 . doi: 10.1145/572260.572264.
20. Flanagan M, Howe DC, Nissenbaum H (2005) Values at Play: Design Tradeoffs in Socially-oriented Game Design. In: Proc CHI. New York, NY, USA, pp 751–760.
21. Friedman B, Kahn Jr PH, Borning A (2008) Value sensitive design and information systems. In: Himma KE, Tavani HT (eds) The Handbook of Information and Computer Ethics. Wiley, Hoboken, NJ, pp 69–102.
22. slomacuser Adobe Illustrator 88. Retrieved from https://www.youtube.com/watch?v=eNLFXKyCy0A. Accessed 12 Dec 2018.
23. Dill E The History of Adobe Illustrator. In: Vecteezy. http://www.vecteezy.com/blog/2010/5/24/the-history-of-adobe-illustrator. Accessed 12 Dec 2018.
24. Inc IMG (1990) InfoWorld. InfoWorld Media Group, Inc.
25. Adobe (1997) Adobe Illustrator 7.0 for the Macintosh New Features (web archive). https://web.archive.org/web/19970726030201/http://www3.adobe.com/prodindex/illustrator/PDFS/ai7newfeatures.pdf

26. Adobe Illustrator website. http://www.adobe.com/products/illustrator.html?sdid=KKQML&mv=search&s_kwcid=AL!3085!3!81143514756!e!!g!!adobe%20illustrator&ef_id=VOOJXgAABfd1VP64:20160523174240:s. Accessed 12 Dec 2018.
27. Steuer S (2004) The Adobe Illustrator CS Wow! Book. Peachpit Press.
28. Creative Cloud Charts (Technology Preview). https://helpx.adobe.com/illustrator/using/creative-cloud-charts-graphs-infographics.html. Accessed 12 Dec 2018.
29. RoyCreative.com Pausing Creative Cloud Charts (Preview). In: Adobe Illus. Blog. http://blogs.adobe.com/adobeillustrator/2015/11/pausing-creative-cloud-charts-preview.html. Accessed 12 Dec 2018.
30. Liu Z, Thompson J, Wilson A, Dontcheva M, Delorey J, Grigg S, Kerr B, Stasko J (2018) Data Illustrator: Augmenting Vector Design Tools with Lazy Data Binding for Expressive Visualization Authoring. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. ACM, New York, NY, USA, pp 123:1–123:13
31. Sarkar D (2015) Package "lattice." https://cran.r-project.org/web/packages/lattice/lattice.pdf. Accessed 12 Dec 2018.
32. Murrell P R graphics, 2nd ed. CRC Press, Boca Raton, FL.
33. Quick-R: Basic Graphs. http://www.statmethods.net/graphs/. Accessed 12 Dec 2018.
34. Top 20 R packages by popularity. http://www.kdnuggets.com/2015/06/top-20-r-packages.html. Accessed 12 Dec 2018.
35. Yau N (2016) Comparing ggplot2 and R Base Graphics. In: FlowingData. http://flowingdata.com/2016/03/22/comparing-ggplot2-and-r-base-graphics/. Accessed 12 Dec 2018.
36. https://plot.ly/ggplot2/. https://plot.ly/ggplot2/. Accessed 12 Dec 2018.
37. Introduction to R Graphics with <code>ggplot2</code>. http://tutorials.iq.harvard.edu/R/Rgraphics/Rgraphics.html#orgheadline19. Accessed 12 Dec 2018.
38. Shiny. http://shiny.rstudio.com/. Accessed 12 Dec 2018.
39. Paradis, A. Illustrator: Graphs. Retrieved from https://www.youtube.com/watch?v=K_vENVmUluM. Accessed 12 Dec 2018.
40. Joseph Rickert (2015) R-bloggers: Some basics for base graphics. In: R-Bloggers. http://www.r-bloggers.com. Accessed 12 Dec 2018.
41. What is Adobe Illustrator? | Adobe Illustrator CC tutorials. https://helpx.adobe.com/illustrator/how-to/what-is-illustrator.html. Accessed 12 Dec 2018.
42. How to create graphs in Illustrator. https://helpx.adobe.com/illustrator/using/graphs.html. Accessed 12 Dec 2018.
43. Grolemund G, Wickham H (2017) R for data science: Data Visualization. https://r4ds.had.co.nz/data-visualisation.html. Accessed 12 Dec 2018.
44. Snyder J, Shilton K, Anderson S (2016) Observing the Materiality of Values in Information Systems Research. Proc HICSS
45. Fleischmann KR (2013) Information and Human Values. Synth Lect Inf Concepts Retr Serv 5:1–99
46. Miller JK, Friedman B, Jancke G, Gill B (2007) Value Tensions in Design: The Value Sensitive Design, Development, and Appropriation of a Corporation's Groupware System. In: Proc GROUP. New York, NY, USA, pp 281–290
47. Siegrist M, Cvetkovich G, Roth C (2000) Salient value similarity, social trust, and risk/benefit perception. Risk Anal 20:353–362 . doi: 10.1111/0272-4332.203034
48. Kosara R (2018) Visualization for Communication. In: Vis. Commun. VisComm IEEE VIS 2018 Workshop. https://viscomm.io/. Accessed 24 Aug 2018.