Chapter

Visual Data Science

Johanna Schmidt

Abstract

Organizations are collecting an increasing amount of data every day. To make use of this rich source of information, more and more employees have to deal with data analysis and data science. Exploring data, understanding its structure, and finding new insights, can be greatly supported by data visualization. Therefore, the increasing interest in data science and data analytics also leads to a growing interest in data visualization and exploratory data analysis. We will outline how existing data visualization techniques are already successfully employed in different data science workflow stages. In some cases, visualization is beneficial, while still future research will be needed for other categories. The vast amount of libraries and applications available for data visualization has fostered its usage in data science. We will highlight the differences among the libraries and applications currently available. Unfortunately, there is still a clear gap between visualization research developments over the past decades and the features provided by commonly used tools and data science applications. Although basic charting options are commonly available, more advanced visualization techniques have hardly been integrated as new features yet.

Keywords: visual data science, data visualization, visual analysis, data visualization libraries, data visualization systems

1. Introduction

Within the last years, data science has been established as its own important emergent scientific field. Data science is defined as a "concept to unify statistics, data analysis, machine learning, and their related methods" to "understand and analyze actual phenomena with data" [1]. As such, data science comprises more than pure statistical data analytics, but the interdisciplinary integration of techniques from mathematics, statistics, computer science, and information science [2]. Data science also involves the consideration of domain knowledge for the analysis and the interpretation of the data and the results [3].

Data visualization research is largely driven by current use cases that users have to face when working with data. The problems and tasks that need to be solved by data scientists are, naturally, a precious source for further developments in data visualization research. On the other hand, data scientists already use data visualizations to visualize data on a daily basis. It is, therefore, worthwhile to think about how the well-established methods for visual analysis fit into the existing workflows of data scientists [4]. According to recent findings from interviews with people working with data [5], data scientists' tasks usually follow a very similar workflow path, and along this path, different stages can be identified. Every stage poses different challenges for data handling. For example, at the beginning of the

workflow, data wrangling is considered to be an essential and tedious part of the workflow. Data wrangling comprises, among others, data parsing, cleaning, and merging. Data visualization techniques can help to quickly identify data flaws like missing data, anomalies like duplicates or outliers, and other inconsistencies in this stage. As a next step, data scientists have to understand the data at hand and evaluate its usefulness for modeling. Here, data visualization can help understand the structure of the data, detect correlations and clusters, and select data parts suitable for modeling.

The rise in data science currently very strongly fuels data visualization techniques by users from very diverse domains. This has now led to many new data visualization tools and libraries being developed. Many of these libraries are open-source and are embedded into programming environments like Python, R, and JavaScript. Prominent examples for such libraries are, for example, *Matplotlib* (Python), *ggplot2* (R), and *D3* (JavaScript). Open source technologies are a great advantage since data scientists can rely on a large community that can provide them with advice and support, and access to a wide range of libraries and plugins. Especially for Python, there are libraries for high-performance computing, numerical calculations, regression modeling, and visualization, which are regularly extended and maintained. On the other hand, feature-rich, standalone visual analysis applications have been increasingly established within the last years. These applications, such as *Tableau*, *Microsoft Power BI* and *Qlik*, provide easy access to data visualization and visual data exploration for users unfamiliar with programming scripting, data wrangling, and/or data visualization design. Standalone applications are usually commercial, since a lot of maintenance and continous development has to happen in the background. As many of these applications are available, data visualization and visual analysis are more widely known and used today in many different domains and are used and applied by many users and domain experts.

This chapter aims to provide a concise overview of existing data visualization techniques for data science and how they fit into the different stages of the data science workflow. Several studies that focused on categorizing and evaluating the different libraries and applications for data visualization currently used in data science will be outlined to create a better picture of which libraries should be used for which type of tasks. Unfortunately, there is still a gap between current research in data visualization and the features and techniques actually provided by libraries and applications. We would, therefore, like to foster the usage of data visualization in data science to bring both communities closer together.

2. Visualization supporting data science

Data science is an interdisciplinary approach that combines input from other domains like mathematics, statistics, computer science, or graphics. Given this vast amount of tasks and skills, several studies have been conducted to understand better and start to categorize the tasks and requirements of data scientists. Kim et al. [6] highlighted the diversity of skills, tasks, and toolsets used by data scientists in software development teams. As an important conclusion, they highlighted that the heterogeneity and diversity make it hard to reuse work. Kandel et al. [5] conducted an interview study with several data scientists and categorized them into the three archetypes of *Hacker*, *Scripter*, and *Application User*. Based on the archetype, data scientists use very diverse tools to solve their tasks. The survey by Harris et al. [7] among different data workers, as they call people working with data, from different disciplines, provided a very comprehensive overview of the different tasks data

scientists need to solve. As a result of their study, they were then able to categorize data scientists into one of four major categories based on their skills (e.g., business orientation vs. programming skills). In general, both studies concluded that data scientists either prefer to use hands-on scripts and program their own algorithms over using fully-featured applications.

As a basis for better explaining how data visualization fits into the data science workflow, we would like to use the categorization introduced by Kandel et al. [5]. They proposed to divide the data science workflow into five Stages:

- **Discover**: As a first step, data scientists usually search for suitable datasets, either by locating them in databases or online or by asking colleagues. Especially within large organizations, finding and understanding relevant data is often considered a significant bottleneck in the work process, also due to access restrictions.

- **Wrangle**: When available, the datasets need to be brought into the desired format. Data wrangling involves parsing files, manipulating data layouts, and also integrating multiple heterogeneous data sources. Being considered to be a very tedious and highly manual task, data wrangling eats up a majority of the time spent on data analysis.

- **Profile**: After being available in the desired format, the quality of the data has to be verified, and the suitability for the analysis has to be estimated. Datasets often contain severe flaws, including missing data, outlier, erroneous values, and other problems. Understanding the structure of the data is therefore considered an important task in data science.

- **Model**: Finally, an essential and interesting part of the data science workflow is to use the datasets as training sets to train prediction models. In this stage, the models have to be created and evaluated against existing real-world data to test their performance.

- **Report**: All analysis results usually need to be reported to external people, colleagues, or customers. In such a presentation, it is important to cover the essential findings discovered during the data science process. In many cases dashboards or reports are used to present the findings.

In the *Discover* stage, data scientists need to identify the data relevant to their current project. This involves searching for internal but also external data sources. The main challenges in this stage are restricted data access and missing documentation of data attributes. This stage is, in general, not supported by data visualization applications. There are approaches in data visualization to, for example, improve the visualization of search engine results [8], but the general problem of data being difficult to find/access is not treated. We therefore do not concentrate on this stage here in this chapter.

2.1 Data wrangling

Data wrangling in the *Wrangle* stage requires to, on the one hand, focus on data flaws like duplicates and inconsistencies (e.g., in naming), and, on the other hand, the process of profiling and transforming datasets. Data wrangling's central goal is to make the data usable in the subsequent steps.

Initially, data wrangling was not considered by data visualization itself, which started to operate once the data was available in the desired format. Since data wrangling nowadays became an essential and tedious task in data science, which eats

up a lot of the time in the whole workflow (up to 50–80% [9]), data visualization researchers started to think about techniques how to support this task. *Wrangler* [10] is the most prominent application to mention here, an interactive system for creating data transformations. Changes in the data are visualized, and data scientists can explore the space of possible operations. The *Wrangler* system infers further actions from what has been done by the user so far manually, and in this way, greatly speeds up the wrangling process. The idea was picked up by the company Trifacta, which included the Wrangler idea into their product to build data pipelines.

In general, data wrangling itself constitutes a very interesting use case which, hopefully, in the future, will get more attention by data visualization research. At the moment, data scientists mostly have to rely on manual tasks and scripting tools to get the data into the right format.

2.2 Data profiling

The most demanding stage in terms of visualization design is the *Profile* stage, where data scientists need to explore the data to understand its structure. This process is very circular and undirected, without a specific goal in mind. Basic information about the related problem domain is required. The goal is to understand the patterns found in the data. This includes, but is not limited to, the distribution of values, correlations, outliers, and clusters.

Datasets usually contain several quality issues, such as missing values, outliers or extreme values, and inconsistencies. Missing data might be due to observations completely missing in a dataset, which can be in many cases identified by empty cells of *null* values. There might also be cases where numbers encode missing data (e.g., 0 or −1) or characters (e.g., "N/A"), something that needs to be considered during the analysis. Inconsistencies and heterogeneous information are often erroneously created by humans, especially in names and terms, and because certain cells have been overloaded with information. Data scientists need to be aware of these flaws when working with a particular dataset. Checking the quality of a dataset has already been addressed by several approaches in visualization. *Profiler* [11] was intended to support the quality assessment of datasets visually; *Visplause* [12] provided the same for time series data. More generally, Bertini et al. [13] developed quality metrics for multi-dimensional datasets. Quality checks are nowadays also provided as features in standalone visualization applications. In Python, the package *pydqc* provides automatic quality checks.

In data visualization, the process of looking at data from different directions and studying different aspects to understand the data structure [14] is called *exploratory data analysis (EDA)*. EDA requires a high degree of interactivity and interconnectivity between different visualizations from a data visualization perspective. EDA has been studied quite extensively within the last years in visualization research. Several paradigms about interaction design [15] and system design [16] have been established. Exploration of data usually happens by using different views and different visualizations. EDA contrasts with the more traditional statistical approach to data analysis that starts with hypothesis testing. In EDA, data scientists usually do not have a clear goal and should support the hypothesis-building process. The typical EDA tasks [17] are:

- Plotting the raw data,
- Plotting simple statistics (e.g., mean, standard deviation, box plots), and
- Positioning such plots for comparison (e.g., in a multiple-view setting).

These tasks are, in general, supported by all visualization applications. In case scripting languages are used, data scientists tend to create several data representations to check various aspects. Programming environments like *Jupyter* Notebooks [19] enable scientists to combine both data analysis scripts and visualization. An example for showing Python *Plotly* visualizations in *Jupyter* can be seen in **Figure 1**. Such narrative or literate programming tools [20] as notebooks help data scientists to record their steps and decisions in a data analysis workflow. They allow scientists to save whole workflows and, in this way, make decisions and results reproducible. This has also been recognized by data visualization research [21], where researchers increasingly think about new solutions for more advanced data visualization in notebook environments and literate programming.

2.3 Modeling

Data scientists make assumptions to find out which types of transformations they need to use for modeling. This also includes understanding which of the data fields are most relevant to a given analysis task. In the *Model* stage, the data is used as input data for building models of the underlying phenomenon. When models have been built, it is important to evaluate them against suitable real-world data.

Building and evaluating simple models like regression models is already supported by some visualization applications [22]. More advanced techniques, often summarized under the term "explainable AI" [23], try to find new, often visual, ways for humans to explain the decision structure of AI (artificial intelligence)

```
import plotly.io as pio
import plotly.express as px
import plotly.offline as py

df = px.data.iris()
fig = px.scatter(df, x="sepal_width", y="sepal_length", color="species", size="sepal_length")
fig
```

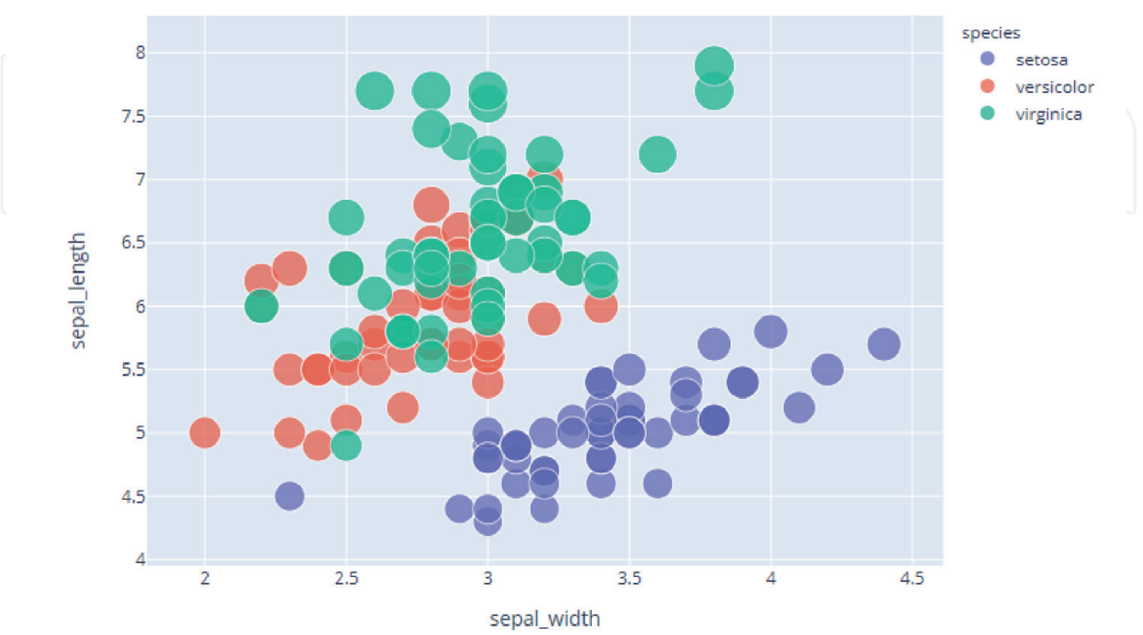

Figure 1.
Jupyter Notebook and Plotly. Literate programming tools as notebooks allow scientists to combine both data analysis scripts and visualization. Figure by [18].

models and verify their decision according to their own ground-truth knowledge. One problem that is mentioned often by data scientists is the scalability of model testing to large data. Currently, models are often evaluated using EDA techniques very similar to the ones described in the *Profile* stage. In the future, data visualization research will concentrate on advanced methods for the verification of model outputs, especially in relation to the input and output data.

2.4 Reporting

In the *Report* stage, mostly simple and easy-to-understand visualizations are needed since here the results of the data analysis stage have to be presented to a broader audience. The use cases in this stage can be mostly covered by employing basic charts, which are already well supported by current data science tools.

In many cases, dashboards with more or less interactivity are used to present the results. Many data science tools already support building dashboards. This was also recognized by the data visualization community recently. Sarikaya et al. [24] pointed out that dashboards are actually much more than just a collection of different graphs and that they need to be treated as separate research objects in data visualization. In their work, they categorized existing dashboards into seven categories, mostly based on the intended task (e.g., information and education vs. decision-making). Three examples are shown in **Figure 2**. Such approaches point

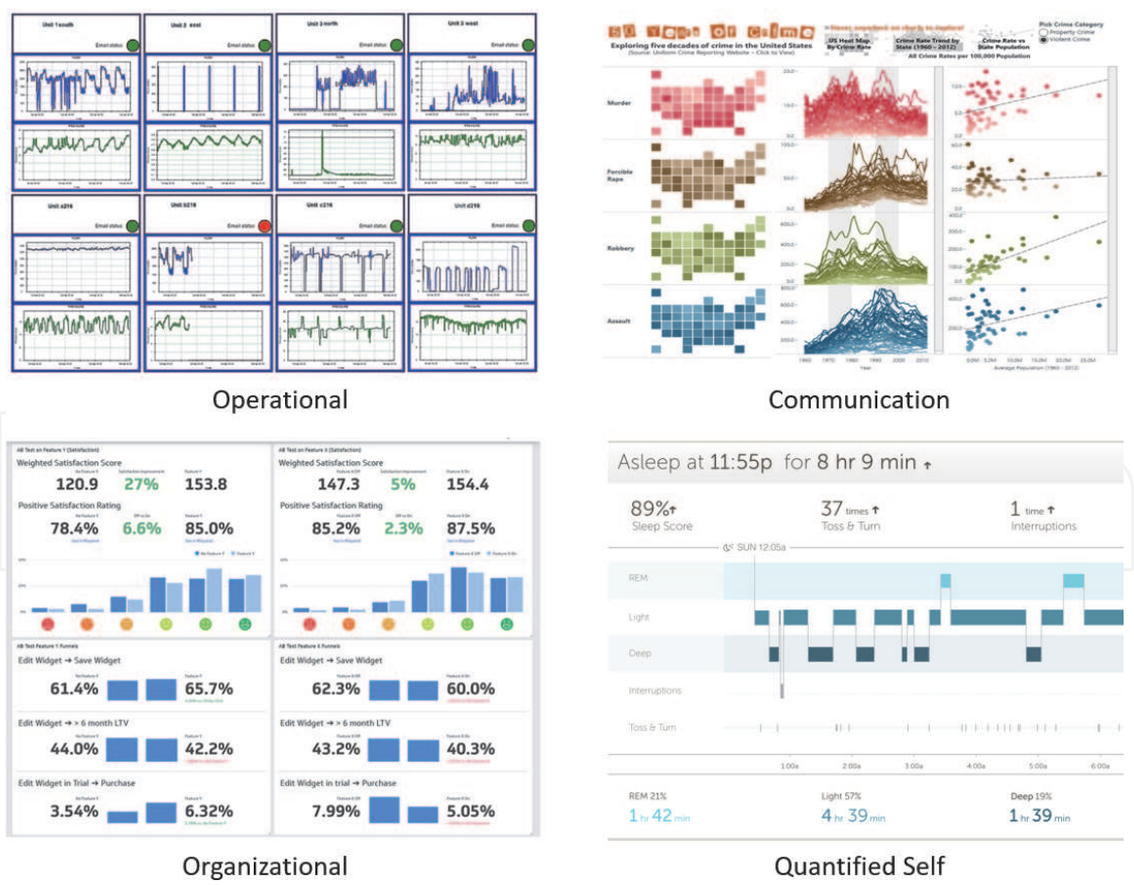

Figure 2.
Dashboards types by [24]. Dashboards for reporting data findings may differ according to the intended user group and task. In this figure, dashboards for operational decision-making, strategic decision-making, communication, and studying your own data (quantified self) are shown. The dashboards in the first column (operational and organizational) target a narrow group of users with particular tasks in mind. The second column's dashboards (communication and quantified self) are intended to be viewed by a larger audience. Images by [24].

out the necessity for dashboard designers to be clear about the intended user group and always have a clear story when presenting data to external people.

One important aspect to consider is to choose the right visualizations for the right type of data. This is especially important if the exact structures in the data are unknown to the viewers and if the viewers' experience with data visualization is unclear. Based on research on human perception and possibilities for data visualization, researchers started to create guidelines for data- or task-driven suggestions for data visualizations. The *Draco* system by Moritz et al. [25] uses predefined rules to suggest several visualizations based on the data and attempt what should be shown in the data. On their website *From Data to Viz*, Holtz and Healy [26] outline several paths how, starting from a specific data type, certain patterns in the datatype can be visualized. The *Data Visualisation Catalogue* [27] summarizes different visualization techniques and explains how they can be employed to encode information. All-in-all, these approaches show the need for further research on guidelines in data visualization research.

3. Data visualization toolboxes

As more and more people started working in data science, more and more software applications for data analysis, many of which are open source, have evolved within the last years [28]. All steps in the data science workflow contain circular processes where data scientists have to rethink actions they made and restart analysis processes from scratch. For this reason of a very interactive and undirected workflow [29], there are no applications, yet, that can cover the entire data science workflow. Data scientists must, therefore, always use a list of combinations of different tools, scripts, and applications to achieve their goals [30]. These tools are often focused on specific tasks, such as efficient data storage and access (e.g., for Big Data applications), data wrangling (i.e., mapping data to another format), or automated analysis (e.g., machine learning). They are based on different programming languages (e.g., Python, R, JavaScript) or are built as fully-featured, standalone applications. In this chapter we specifically concentrate on libraries and tools for data visualization.

When talking about libraries and application for data visualization we use the definition by Rost [31]. They conducted a study about features of visualization libraries and applications by creating one specific chart with different tools. In the study the authors differentiate between *charting libraries* (i.e., programming toolkits) and *apps* (i.e., fully-featured applications). As also noted by Kandel et al. [5], different types of data scientists tend to use different types of tools. A data scientist being identified as an archetype *hacker* would not be happy by having to use a standalone application, because he/she would not be able to access the latest library in a scripting environment, and would therefore not be able to customize his/her individual workflow. We, therefore, stick to this differentiation in this chapter.

3.1 Charting libraries

Charting libraries are considered to be all kinds of visualization libraries that need some programming environment to work. In many cases, this is a scripting environment, so many libraries nowadays are based on Python or R. The popularity of data visualization libraries changes from year to year since many of these libraries are open source and therefore undergo continuous adaptations and improvements. Open-source technologies are a great advantage since data scientists can rely on a

large community that can provide them with advice and support, and access to a wide range of libraries and plugins. There are some libraries which are repeatedly mentioned in high score lists [32], which are, among others: *ggplot2* (R), *Matplotlib* (Python), *Seaborn* (Python), *Bokeh* (Python), *D3* (JavaScript), *Chart.js* (JavaScript) *Lattice* (R), *Vegas* (Scala), *Breeze-viz* (Scala), *Rgl* (R). The differences between these libraries are, on the one hand, given by the different programming environments they live in. On the other hand, the libraries also offer different features and assets for data visualization. Especially for Python, there are libraries for high-performance computing, numerical calculations, regression modeling, and visualization, which are regularly extended and maintained. This is very similar in the case of R.

The study by Rost [31] reveals fascinating differences between some of the charting libraries. The libraries which have been tested in this study have been classified according to whether they are more suited for analysis tasks or presentation tasks. The results can be seen in **Figure 3**. The analysis very nicely shows that charting libraries for both analysis (*Wrangle*, *Profile*, *Model*) and presentation (*Report*) purposes can be found. Interestingly, the charting libraries rather suited for presentation are based on JavaScript (highlighted by underline). This also shows that web-based visualization methods are currently rather placed in the presentation or reporting phase of a data science workflow. This also makes sense when thinking about the client–server environment of web-based visualizations, and that visualization designers have to carefully think about which type of data to show in this setting–since large datasets could probably not be transferred over the network and could potentially lead to processing or rendering problems on the client-side (e.g., smartphones). Such a careful design can usually only be done after the analysis (*Profile*, *Model*) is already finished.

In the study by Schmidt [33], different charting libraries were compared according to how many different visualization techniques they support. This study revealed big differences between the libraries and identified two leaders in the field, which currently offer the largest range of different visualization techniques. The first leader is *D3* (short for Data-Driven Documents), which is based on JavaScript and uses SVG (Scalable Vector Graphics) elements to display data in the web browser. It was released in 2011 [34] as a successor to the earlier *Protovis* framework to provide a more expressive framework that, at the same time, focuses on web standards and provides improved performance. The second leader is *Plotly*, a collaborative browser-based plotting and analytics platform based on Python [35]. *Plotly* developers especially take care to allow data scientists to share visualizations and information within a large community.

All-in-all, the field of charting libraries is constantly changing, and many more advances are expected to be seen in the future. When deciding for a charting

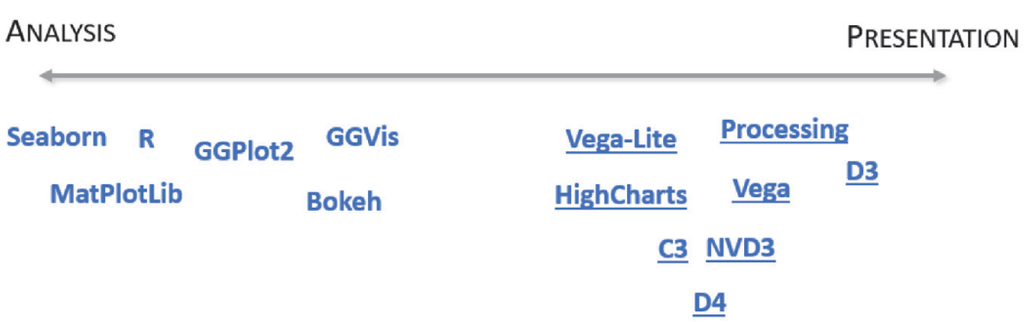

Figure 3.
Comparison of charting libraries. The chart shows the charting libraries used in the study by Rost [31] ranked by whether they are rather suited for analysis or presentation. Charting libraries based on JavaScript (which are, therefore, web-based) are marked by underline. Figure adapted from [31].

library to be used, other factors like the task to be solved and programming skills have to be considered.

3.2 Apps

Apps are considered to be fully-featured, standalone applications. They do not require any programming environment to be installed on a system to run them. Data visualizations can be created by using the user interface tools provided by the application. Apps are more targeted towards users without programming skills who are not familiar with manual data processing, analytics, and visualization. In almost all cases, apps are commercial products. This is because a lot of maintenance and continuous development is needed in the background to keep the apps up-to-date. According to Gartner's Magic Quadrants, a study that is done every year in different areas, the leaders in the field of business intelligence platforms [36] are considered to be *Tableau*, *Microsoft Power BI*, and *Qlik*.

In its yearly study, Gartner compares business intelligence applications that are considered most significant in the marketplace. The applications are evaluated and placed in one of four quadrants, rating the applications as either challengers, leaders, visionaries, or niche players. Many apps are currently available on the market. These applications differ in terms of targeted user groups and also visualization features that they offer. Since Gartner's Magic Quadrants are published every year, interesting patterns can be detected by looking at the yearly changes, as shown in **Figure 4**. The three leaders that have been identified previously already show an excellent performance throughout the last six years. Interestingly, the field of leaders is left to the three main players over the last years. It can also be seen that

Figure 4.
Gartner's Magic Quadrants of the last six years. The quadrants are divided into the four fields of Leaders, Visionaries, Niche Players, and Challengers. It can be seen that the group of leaders only slightly changed over the last four years. Especially Qlik stayed quite constant. The group of other apps (indicated by a gray circle) shows the field's dynamic movements. New apps have been developed (e.g., Infor, 2021) and others disappeared (e.g., GoodData, 2019). Figure adapted from [37].

other tools appeared or vanished over the years, which shows the dynamic of the market of business intelligence tools.

The differences between commercial tools have also been highlighted in other studies. Zhang et al. [38] concentrated on specific visualization techniques and evaluated their usage in commercial business analytics tools. They ranked tools and applications based on classifications according to feature richness, flexibility, learning curve, and tasks (e.g., for analysis or presentation). Behrisch et al. [39] conducted an exhaustive survey on commercial visual analytics tools, evaluating them according to which degree they feature data handling, visualization, and automated analysis. Their findings classified the applications according to whether they are more suited for presentation or exploratory analysis. The results show that basically all applications feature data presentation, which is mainly supported by creating dashboards. Some of the applications like *Tableau* or *Qlik* also provide the ability to publish web-based dashboards. Interestingly, only about 50% of the applications were identified to be suited for exploratory analysis (like *Tableau*, *TIBCO Spotfire*, or *Microsoft Power BI*). The authors also identified the applications as useful for different types of users, mainly upper management, reporting managers, or data analysts.

The vast amount of libraries and tools being available has inspired researchers to conduct studies for quantifying, evaluating, and ranking tools and applications that data scientists use. Gartner's Magic Quadrant and several studies about apps for data visualization in data science show no tools that cover all tasks and needs. The selection of an app to be used mainly depends on the tasks that need to be solved (e.g., analysis vs. presentation) and on the scope where the app should be used in.

4. Integration of visualization

Visualization researchers were very successful within the last decades, generating many different novel techniques for the visual representation of data. These techniques range from approaches for the efficient representation of data (e.g., parallel coordinates) to proposed interaction and user guidance workflows (e.g., overview-first, details-on-demand). Current surveys show a large variety of visualization techniques. A survey of survey papers in information visualization by McNabb and Laramee [40] classified already over 80 survey papers describing relevant state-of-the-art techniques, and a more recent survey of books in information visualization revealed a similar quantity and variety [41]. Unfortunately, there is only minimal overlap between the recent developments in visualization research and the data visualization features offered by charting libraries and apps. Most of the tools and applications feature basic charts and plots (e.g., scatter plots, bar charts, bubble charts, radar charts), but more advanced visualization techniques (e.g., chord diagrams, horizon graphs) can hardly be found.

This was confirmed by several studies on the integration of visualization techniques in common libraries and applications. Harger and Crossno [42] evaluated the feature richness of open source toolkits for visual analytics. They evaluated the toolkits used for the study based on which basic chart types (e.g., bar charts, line charts), which types of graph visualization (e.g., circular or force-directed layouts), and which types of geo-spatial visualization techniques (e.g., choropleth maps, cartograms) they feature. They concluded that some toolkits are more targeted towards analytics, and some are more targeted towards visualization. Like this study, Schmidt [33] surveyed commonly used tools and applications and evaluated the visualization techniques they feature. They focused on visualization techniques rather than on derived attributes (e.g., feature richness) and included more recent

advances in visualization research, considered open source tools and commercial applications to produce a complete picture of visualization techniques usage. They also concentrated on 2D information visualization techniques, as these techniques are more relevant for data science and data analytics, and disregarded spatial techniques like 3D volume rendering.

In all studies that have been conducted so far, not surprisingly, basic chart types like scatter plots and bar charts are highly supported by all evaluated tools and applications. From the more advanced visualization techniques, multi-dimensional techniques like parallel coordinates and radar charts are already widely used and known and therefore included in many of the tools. The same applies to scatter plot matrices and heatmaps. Techniques for hierarchical data are also well supported, especially by open source tools. Visualization techniques for temporal data are not available in the majority of the tools and applications. This is probably because temporal data (e.g., time-series data) is a particular data type used only for specific tasks. Users usually use their own tools for these purposes. Therefore, temporal data techniques have not been included yet in common tools and applications, as these tools usually try to address a broader range of data scientists and data analysts. Some visualization techniques have not been integrated into any tool or application yet, like time nets, data vases, or people garden.

From a tools and applications point of view, *Plotly* and *D3* notably provide the most features among all the tested open source tools. Other tools are targeted towards exceptional functionalities, like *dygraphs* for scientific plots, which only feature a minimal range of visualization techniques. Other libraries which are intended to be used in web-based applications (e.g., *Chart.js* or *Google Charts*) feature only visualization techniques that will most likely be needed in a web-based context. Open source tools, especially *ggplot2*, benefit a lot from the community's input since many advanced visualization techniques are only featured via extensions. In the group of commercial tools, it can be depicted that *Tableau*, *Microsoft Power BI*, and *Highcharts* feature most of the hereby evaluated visualization techniques.

Data scientists could be supported in all stages of their workflow by using visual tools. Interestingly, visualization techniques are currently mostly applied in the *Report* stage, at the end of the data science workflow. This stands in contrast to the fact that interactive data exploration workflows are strongly promoted by visualization research. Even worse, the support for more advanced visualization techniques, especially for interactive data exploration, is still minimal. This has been identified as the "Interactive Visualization Gap" by Batch and Elmquist [43]. Further exchange with data science is considered a valuable and important goal for the visualization community. Previous research efforts in data science revealed that the gap between new developments in visualization research and their application "in the wild" still exists and will hopefully be further mitigated in the future.

5. Conclusions

Data visualization can provide substantial support for users working with data. Data visualization techniques have proven to be useful for different steps in the data science workflow. The techniques differ in the interactivity and complexity of the representations. Many of the visualization techniques have been successfully integrated into libraries and applications for data visualization. Especially in the open-source sector, many new directions have opened up within the last years. Due to the programming languages' success, like Python, R, and Scala, libraries targeted towards these programming environments are becoming especially popular.

Among them are *Plotly* for Python and *ggplot2* for R. Also, web-based applications increasingly gain importance. That is why JavaScript-based libraries like *D3* and *Chart.js* can also be found among the most popular data visualization libraries. The market of business intelligence tools is also very dynamic, but it shows some three leaders in the field, namely *Tableau*, *Microsoft Power BI* and *Qlik*. Different types of data scientists require different libraries or applications. It, therefore, can be seen that applications are increasingly targeted towards a specific goal and are designed to solve specific types of tasks. However, when looking at the data visualization techniques offered by the most prominent libraries and applications, the "Interactive Visualization Gap" for exploratory data analysis still exists. Many recent developments and implementations in data visualization research do not find their way into existing libraries and applications. Therefore, the further exchange between data science and data visualization is highly recommended, as both parties can learn a lot from each other and, together, further foster the usage of data visualization in data analytics.

Acknowledgements

VRVis is funded by BMK, BMDW, Styria, SFG, Tyrol and Vienna Business Agency in the scope of COMET - Competence Centers for Excellent Technologies (879730) which is managed by FFG.

Author details

Johanna Schmidt
VRVis Zentrum für Virtual Reality und Visualisierung Forschungs-GmbH, Vienna, Austria

*Address all correspondence to: johanna.schmidt@vrvis.at

References

[1] Chikio Hayashi (1998) *What is Data Science? Fundamental Concepts and a Heuristic Example*. In Data Science, Classification, and Related Methods, pp. 40—51. Springer Japan.

[2] Mark A. Parsons, Øystein Godøy, Ellsworth LeDrew, Taco F. de Bruin, Bruno Danis, Scott Tomlinson, and David Carlson (2011) *A conceptual framework for managing very diverse data for complex, interdisciplinary science*. Journal of Information Science, 37(6): 555—569.

[3] David M. Blei and Padhraic Smyth (2017) *Science and Data Science*. Proceedings of the National Academy of Sciences, 114(33):8689—8692.

[4] Natalia Andrienko, Gennady Andrienko, Georg Fuchs, Aidan Slingsby, Cagatay Turkay, and Stefan Wrobel (2020) *Visual Analytics for Data Scientists*. Springer International Publishing.

[5] Sean Kandel, Andreas Paepcke, Joseph M. Hellerstein, and Jeffrey Heer (2012) *Enterprise Data Analysis and Visualization: An Interview Study*. IEEE Transactions on Visualization and Computer Graphics, 18(12):2917–2926.

[6] Miryung Kim, Thomas Zimmermann, Robert DeLine, and Andrew Bege (2018) *Data Scientists in Software Teams: State of the Art and Challenges*. IEEE Transactions on Software Engineering, 44(11):1024—1038.

[7] Harlan D. Harris, Sean P. Murphy and Marck Vaisman (2013) *Analyzing the Analyzers: An Introspective Survey of Data Scientists and Their Work*. O'Reilly Media, Inc.

[8] Edward Clarkson, Krishna Desai, and James Foley (2009) *ResultMaps: Visualization for Search Interfaces*. IEEE Transactions on Visualization and Computer Graphics 15(6):1057–1064.

[9] Tye Rattenbury, Joseph M. Hellerstein, Jeffrey Heer, Sean Kandel, and Connor Carreras (2017) *Principles of Data Wrangling: Practical Techniques for Data Preparation*. O'Reilly Media, Inc.

[10] Sean Kandel, Andreas Paepcke, Joseph Hellerstein, and Jeffrey Heer (2011) *Wrangler: Interactive visual specification of data transformation scripts*. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '11, May 7–12, Vancouver, Canada, pp. 3363–3372.

[11] Sean Kandel, Ravi Parikh, Andreas Paepcke, Joseph M. Hellerstein, and Jeffrey Heer (2012) *Profiler: Integrated Statistical Analysis and Visualization for Data Quality Assessment*. Proceedings of the International Working Conference on Advanced Visual Interfaces, AVI '12, May 22–25, Capri, Italy, pp. 547—554.

[12] Clemens Arbesser, Florian Spechtenhauser, Thomas Mühlbacher, and Harald Piringer (2016) *Visplause: Visual data quality assessment of many time series using plausibility checks*. IEEE Transactions on Visualization and Computer Graphics 23(1):641–650.

[13] Enrico Bertini, Andrada Tatu, and Daniel Keim (2011) *Quality Metrics in High-Dimensional Data Visualization: An Overview and Systematization*. IEEE Transactions on Visualization and Computer Graphics 17(12):2203–2212.

[14] Peter Filzmoser, Karel Hron, and Matthias Templ (2018) *Exploratory Data Analysis and Visualization*. Applied Compositional Data Analysis: With Worked Examples in R, pp. 69–83, Springer International Publishing.

[15] Arvind Satyanarayan, Kanit Wongsuphasawat, and Jeffrey Heer

(2014) *Declarative Interaction Design for Data Visualization*. Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology, UIST '14, Honolulu, Hawaii, Oct 5–8, USA, pp. 669—678.

[16] Tamara Munzner (2014) *Visualization Analysis and Design*. Taylor & Francis, Inc.

[17] NIST/SEMATECH e-Handbook of Statistical Methods (2021) http://www.itl.nist.gov/div898/handbook/ [Accessed 2021-03-01].

[18] Project Jupyter (2021) *Interactive data visualizations* https://jupyterbook.org/interactive/interactive.html [Accessed 2021-03-02].

[19] Project Jupyter (2021) https://jupyter.org/ [Accessed 2021-03-02].

[20] Mary Beth Kery, Marissa Radensky, Mahima Arya, Bonnie E. John, and Brad A. Myers (2018) *The Story in the Notebook: Exploratory Data Science Using a Literate Programming Tool*. In Proceedings of the CHI Conference on Human Factors in Computing Systems, CHI '18, Apr. 21–26, Montreal QC, Canada.

[21] Yifan Wu, Joseph M. Hellerstein, and Arvind Satyanarayan (2020) *B2: Bridging Code and Interactive Visualization in Computational Notebooks*. In Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology, UIST '20, Oct 20–23, Virtual Event, pp. 152–165.

[22] Thomas Mühlbacher and Harald Piringer (2013) *A partition-based framework for building and validating regression models*. IEEE Transactions on Visualization and Computer Graphics 19 (12):1962–1971.

[23] Wojciech Samek, Grégoire Montavon, Andrea Vedaldi, Lars Kai Hansen, and Klaus-Robert Müller (2019) *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Springer International Publishing.

[24] Alper Sarikaya, Michael Correll, Lyn Bartram, Melanie Tory, and Danyel Fisher (2019) *What Do We Talk About When We Talk About Dashboards?*. IEEE Transactions on Visualization and Computer Graphics 29(1):682—692.

[25] Dominik Moritz, Chenglong Wang, Gregory Nelson, Halden Lin, Adam M. Smith, Bill Howe, Jeffrey Heer (2019) *Formalizing Visualization Design Knowledge as Constraints: Actionable and Extensible Models in Draco*. IEEE Transactions on Visualization and Computer Graphics 25(1):438–448.

[26] Yan Holtz and Conor Healy (2017) Holtz, Y. and Healy, C. (2017) *From Data to Viz* https://www.data-to-viz.com/ [Accessed 2021-02-20].

[27] Severino Ribecca (2021) *The Data Visualisation Catalogue*. https://datavizcatalogue.com/ [Accessed 2021-03-02].

[28] Panagiotis Barlas, Ivor Lanning, and Cathal Heavey (2020) *A survey of open source data science tools*. International Journal of Intelligent Computing and Cybernetics 8(3):232–261.

[29] Jiali Liu, Nadia Boukhelifa, and James R. Eagan (2019) *Understanding the Role of Alternatives in Data Analysis Practices*. IEEE Transactions on Visualization and Computer Graphics 26(1):66—76.

[30] Sara Alspaugh, Nava Zokaei, Andrea Liu, Cindy Jin, and Marti A. Hearst (2019) *Futzing and Moseying: Interviews with Professional Data Analysts on Exploration Practices*. IEEE Transactions on Visualization and Computer Graphics 25(1):22—31.

[31] Lisa Charlotte Rost (2016). *What I Learned Recreating One Chart Using 24*

Tools. https://source.opennews.org/articles/what-i-learned-recreating-one-chart-using-24-tools/ [Accessed 2021-03-05].

[32] Bob Hayes (2019) *Business Broadway: Programming Languages Most Used and Recommended by Data Scientists*. https://businessoverbroadway.com/2019/01/13/programming-languages-most-used-and-recommended-by-data-scientists/ [Accessed 2021-02-21].

[33] Johanna Schmidt (2020) *Usage of Visualization Techniques in Data Science Workflows*. In Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISIGRAPP '20, Valletta, Malta, Feb 27–29, pp. 309–316.

[34] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer (2011) *D3: Data-Driven Documents*. IEEE Transactions on Visualization and Computer Graphics 17(12):2301—2309.

[35] Plotly Technologies Inc (2015) *Collaborative data science*. Montréal, QC.

[36] Gartner (2021) *Magic Quadrant for Analytics and Business Intelligence Platforms*. https://www.gartner.com/reviews/market/analytics-business-intelligence-platforms [Accessed 2021-03-01].

[37] Gartner (2021) *Gartner Magic Quadrant*. https://www.gartner.com/en/research/methodologies/magic-quadrants-research [Accessed 2021-02-05]

[38] Leishi Zhang, Andreas Stoffel, Michael Behrisch, Sebastian Mittelstadt, Tobias Schreck, René Pompl, Stefan Hagen Weber, Holger Last, and Daniel Keim (2012) *Visual analytics for the big data era – A comparative review of state-of-the-art commercial systems*. In Proceedings of the IEEE Conference on Visual Analytics Science and Technology, VAST '12, Oct. 14–19, Seattle, WA, USA, pp. 173—182.

[39] Michael Behrisch, Dirk Streeb, Florian Stoffel, Daniel Seebacher, Brian Matejek, Stefan Hagen Weber, Sebastian Mittelstaedt, Hanspeter Pfister, and Daniel Keim (2018) *Commercial Visual Analytics Systems-Advances in the Big Data Analytics Field*. IEEE Transactions on Visualization and Computer Graphics 25(1):3011–3031.

[40] Liam McNabb and Robert S. Laramee (2017) *Survey of Surveys (SoS) - Mapping The Landscape of Survey Papers in Information Visualization*. Computer Graphics Forum 36:589—617.

[41] Dylan Rees and Robert S. Laramee (2019) *A Survey of Information Visualization Books*. Computer Graphics Forum, 38:610—646.

[42] John R. Harger and Patricia J. Crossno (2012) *Comparison of Open Source Visual Analytics Toolkits*. Proceedings of SPIE - The International Society for Optical Engineering, 8294.

[43] Andrea Batch and Niklas Elmqvist (2018) *The Interactive Visualization Gap in Initial Exploratory Data Analysis*. IEEE Transactions on Visualization and Computer Graphics, 24(1):278—287.