April 2018

Sight.js: Data Analysis in Splunk

Alexander Dyer
Worcester Polytechnic Institute

Follow this and additional works at: https://digitalcommons.wpi.edu/mqp-all

Repository Citation

This Unrestricted is brought to you for free and open access by the Major Qualifying Projects at Digital WPI. It has been accepted for inclusion in Major Qualifying Projects (All Years) by an authorized administrator of Digital WPI. For more information, please contact digitalwpi@wpi.edu.
Sight.js: Data Analysis in Splunk

Major Qualifying Project

Advisor:
LANE HARRISON

Written By:
ALEXANDER DYER

A Major Qualifying Project
WORCESTER POLYTECHNIC INSTITUTE

Submitted to the Faculty of the Worcester Polytechnic Institute in partial fulfillment of the requirements for the Degree of Bachelor of Science in Computer Science.

OCTOBER 24TH, 2017 - APRIL 26TH, 2018
Abstract

A comparison of tools, which track user interaction with web visualizations, and an assertion of the necessities for gathering meaningful insights from user interactions with web visualizations.
# Table of Contents

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Tables</td>
</tr>
<tr>
<td>List of Figures</td>
</tr>
</tbody>
</table>

1 Introduction
   1.1 User Interaction in the Information Age | 1 |
   1.2 Examination of the Design Space | 1 |
   1.3 The Data Hierarchy of Visualizations on the Web | 2 |
   1.4 Closing the Feedback Loop in Web Visualization | 3 |

2 Background
   2.1 Understanding User Interactions through Interaction Logs | 5 |
   2.2 Remotely Monitoring User Interaction | 5 |
   2.3 Applications of the Union of Remote Monitoring and Usability Logs | 6 |

3 Methodology
   3.1 Study Objectives | 7 |
   3.2 Instrumentation Evaluation Methodology | 7 |
   3.3 Criteria for Evaluating Instrumentation Tools for Interactive Data Visualization | 9 |
   3.4 Visualizing User Interaction with Sight.js | 10 |
      3.4.1 Applying Sight.js to an Existing Web Visualization | 10 |
      3.4.2 Insights into User Interaction in Splunk | 10 |

4 Comparing Instrumentation Tools for Interactive Data Visualizations | 12 |
   4.1 Evaluation Strategy in Brief | 12 |
   4.2 Google Analytics | 13 |
      4.2.1 Effort | 13 |
      4.2.2 Versatility | 13 |
      4.2.3 Data | 14 |
      4.2.4 Performance | 15 |
      4.2.5 Visualization | 15 |
TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2.6 Price</td>
<td>16</td>
</tr>
<tr>
<td>4.3 Session Stack</td>
<td>17</td>
</tr>
<tr>
<td>4.3.1 Effort</td>
<td>17</td>
</tr>
<tr>
<td>4.3.2 Versatility</td>
<td>18</td>
</tr>
<tr>
<td>4.3.3 Data</td>
<td>18</td>
</tr>
<tr>
<td>4.3.4 Performance</td>
<td>18</td>
</tr>
<tr>
<td>4.3.5 Visualization</td>
<td>18</td>
</tr>
<tr>
<td>4.3.6 Interface</td>
<td>18</td>
</tr>
<tr>
<td>4.3.7 Price</td>
<td>18</td>
</tr>
<tr>
<td>4.4 Sight.js</td>
<td>19</td>
</tr>
<tr>
<td>4.4.1 Effort</td>
<td>19</td>
</tr>
<tr>
<td>4.4.2 Versatility</td>
<td>20</td>
</tr>
<tr>
<td>4.4.3 Data</td>
<td>20</td>
</tr>
<tr>
<td>4.4.4 Performance</td>
<td>20</td>
</tr>
<tr>
<td>4.4.5 Visualization</td>
<td>20</td>
</tr>
<tr>
<td>4.4.6 Interface</td>
<td>21</td>
</tr>
<tr>
<td>4.4.7 Price</td>
<td>21</td>
</tr>
<tr>
<td>4.5 Mixpanel</td>
<td>21</td>
</tr>
<tr>
<td>4.5.1 Effort</td>
<td>21</td>
</tr>
<tr>
<td>4.5.2 Versatility</td>
<td>22</td>
</tr>
<tr>
<td>4.5.3 Data</td>
<td>22</td>
</tr>
<tr>
<td>4.5.4 Performance</td>
<td>22</td>
</tr>
<tr>
<td>4.5.5 Visualization</td>
<td>22</td>
</tr>
<tr>
<td>4.5.6 Interface</td>
<td>22</td>
</tr>
<tr>
<td>4.5.7 Price</td>
<td>22</td>
</tr>
<tr>
<td>4.6 Numerical Evaluation of All Tools</td>
<td>23</td>
</tr>
<tr>
<td>5 Applying Sight.js to Gain Insight into User Interaction</td>
<td>26</td>
</tr>
<tr>
<td>5.1 Visualizing User Interaction Logs with Splunk</td>
<td>26</td>
</tr>
<tr>
<td>5.2 Applications of Visualized User Interaction Logs</td>
<td>29</td>
</tr>
<tr>
<td>6 Discussion</td>
<td>31</td>
</tr>
<tr>
<td>6.1 Patterns Across Solutions</td>
<td>31</td>
</tr>
<tr>
<td>6.2 Insights from Exploring the Design Space</td>
<td>32</td>
</tr>
<tr>
<td>6.3 Benefits of a Visualization Specific Tool</td>
<td>32</td>
</tr>
<tr>
<td>7 Conclusions and Recommendations</td>
<td>34</td>
</tr>
<tr>
<td>7.1 Impact</td>
<td>34</td>
</tr>
<tr>
<td>7.2 Conclusion</td>
<td>34</td>
</tr>
</tbody>
</table>

iii
| Appendix A: Splunk Visualizations from Sight.js Data | i |
| Bibliography | iii |
LIST OF TABLES

<table>
<thead>
<tr>
<th>TABLE</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Sample Google Analytics Export.</td>
</tr>
<tr>
<td>4.2</td>
<td>Numerical Evaluation of Metrics</td>
</tr>
<tr>
<td>Figure</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>1.1</td>
<td>2</td>
</tr>
<tr>
<td>3.1</td>
<td>8</td>
</tr>
<tr>
<td>3.2</td>
<td>11</td>
</tr>
<tr>
<td>4.1</td>
<td>12</td>
</tr>
<tr>
<td>4.2</td>
<td>13</td>
</tr>
<tr>
<td>4.3</td>
<td>13</td>
</tr>
<tr>
<td>4.4</td>
<td>15</td>
</tr>
<tr>
<td>4.5</td>
<td>15</td>
</tr>
<tr>
<td>4.6</td>
<td>16</td>
</tr>
<tr>
<td>4.7</td>
<td>16</td>
</tr>
<tr>
<td>4.8</td>
<td>17</td>
</tr>
<tr>
<td>4.9</td>
<td>17</td>
</tr>
<tr>
<td>4.10</td>
<td>17</td>
</tr>
<tr>
<td>4.11</td>
<td>19</td>
</tr>
<tr>
<td>4.12</td>
<td>19</td>
</tr>
<tr>
<td>4.13</td>
<td>20</td>
</tr>
<tr>
<td>4.14</td>
<td>21</td>
</tr>
<tr>
<td>4.15</td>
<td>21</td>
</tr>
<tr>
<td>4.16</td>
<td>22</td>
</tr>
<tr>
<td>4.17</td>
<td>23</td>
</tr>
<tr>
<td>4.18</td>
<td>23</td>
</tr>
<tr>
<td>4.19</td>
<td>24</td>
</tr>
<tr>
<td>4.20</td>
<td>24</td>
</tr>
<tr>
<td>5.1</td>
<td>26</td>
</tr>
<tr>
<td>5.2</td>
<td>27</td>
</tr>
<tr>
<td>5.3</td>
<td>27</td>
</tr>
<tr>
<td>5.4</td>
<td>28</td>
</tr>
</tbody>
</table>
5.5 Numerically represented counts of mouse events and the average absolute velocity of a user’s session. .................................................. 28
5.6 A breakdown of mouse actions and page events broken down by user sessions. .... 28
5.7 The top ten most visited planets. ................................................. 29
5.8 Exoplanet radius and count of DOM Events. .................................... 29

1 Manhattan distance dissimilarity function plotted against count of DOM events. . . i
2 A multiseries timechart of user interactions with the exoplanets visualization. Note that the use of multiple series aids readability when compared to figure 5.3. .......... i
3 Tracking of mouse movements and categorization by event type. This became the more robust implementation of figure 5.4. ........................................ ii
4 Pie charts of DOM events occurring on individual planets. .......................... ii
1.1 User Interaction in the Information Age

In the advent of the information age, the amount of trackable user information is increasing past previously known limits. Large scale data aggregation and analysis is becoming a very real, and sometimes essential, field of endeavour. Online services such as search engines, online retailers, and social media sites use the interactions of their users to tailor the experience on the platform. This application of the user feedback inherent in the user’s interaction is invaluable. Despite the widespread knowledge of this, many news websites and other companies which make visualizations on the web are not monitoring users’ interaction with web visualizations.

1.2 Examination of the Design Space

There are a few tools in the current design space which can potentially fill the role of combining remote user interaction logs and systemic evaluation. The tools Google Analytics, Session Stack, and Mixpanel are primarily centered around tracking websites in general, not visualizations in particular, but have the ability to track user interaction [4–6].

Google Analytics and Mixpanel seek to understand user engagement with a webpage and determining user trends. Many applications of these tools center around which demographics engage with specific portions of the webpage. For example, a company might employ Google Analytics to determine how many users are interacting with a video on the page. For monetization purposes or to determine what a user interacts with, these tools are very helpful. The difficulty arises when lower level, more descriptive data is desired.

Session Stack was designed with a different philosophy, yet can accomplish similar goals. By recording user sessions on a webpage, allowing playback, and displaying a debug log, Session
Stack allows for viewing of the user’s actual session, complete with generated errors or info level data. Rather than relying on an amalgamation of user logs, Session Stack shows exactly what the user is doing. This application is well suited to the task of debugging webpages, seeing what went wrong and where.

These tools will be compared to the visualization specific Javascript library, Sight.js, and the efficacy of each tool to track user interaction will be evaluated. As a part of understanding the usefulness of considering user interaction logs to close the feedback loop, visualization of user interactions will be examined.

### 1.3 The Data Hierarchy of Visualizations on the Web

When looking at user interaction with a web visualization there are a number of data facets to consider. Functionally, the tracked data can be broken down into three distinct sections. This hierarchical structure of web visualization data is illustrated in figure 1.1.

![Data Hierarchy of Visualizations on the Web](image)

**Figure 1.1:** The data hierarchy of web visualization.

The top layer is the data associated with the Document Object Model (DOM). This layer encompasses events like mouse movement and clicking. Basic interaction with a web visualization is captured here. It is possible to see what users are doing in this layer, but motives are largely indiscernable.
1.4 CLOSING THE FEEDBACK LOOP IN WEB VISUALIZATION

The middle layer holds the data associated with the Scalable Vector Graphic (SVG) elements. These elements are the shapes which make up web visualizations. The type of data points captured in this layer are the exact coordinates of an element on the page, the radius or length of the element, and the actual shape of the element. At this level of data, a more holistic picture of the user’s interaction is drawn. Limited understanding of user motives can be achieved at this point, correlating user interaction and shape.

The deepest layer contains the data bound to the SVG elements. In this layer, the SVG elements are no longer just shapes, but a representation of the underlying data. The addition of this layer fills out the flaws in the other two to form a gestalt. The user’s interaction with the web visualization reaches its clearest point. In the deepest layer, user interactivity is placed in complete context. This allows for educated inferences on user motives, tying together the visual cues of the SVG elements and the intellectual cues of the underlying data.

1.4 Closing the Feedback Loop in Web Visualization

In order to evaluate the existing market solutions for tracking user interaction on the web, a set of criteria had to be developed. Each tool was ranked based upon the difficulty of its implementation, the flexibility of the data gathered, the level of data collected from the data hierarchy of web visualizations, the celerity with which the tool was able to complete its task, the robustness of any built-in visualization or analysis tools, the complexity of the interface, and the cost of the tool. These criteria were assessed through a standardized trial with the same visualization.

The two most important criteria were found to be the versatility of the tool and level of data gathered. Of all the metrics discussed, these two contribute the most to the task of understanding user interactions with web visualizations. Without the entire picture of DOM, SVG, and library level data, it is more likely that experts would misinterpret user motives, as they interact with visualizations. Furthermore, if the entirety of the collected data is locked within the tool used to capture it, the data can become effectively useless. Full analysis of user interaction falls outside the capabilities of any of the data collection tools examined. In order to achieve a full picture from the data, it was found that outside tools are required to form more robust visualizations and analyses.

In addition to evaluating existing tools, this project explored the impact of understanding user interactions with visualizations using the Sight.js library. On the same visualization used to evaluate existing solutions, a user was asked to interact for a brief period. The resulting interaction logs were explored, analysed, and visualized to show the potential information made possible by closing the feedback loop with online web visualizations.

Closing the feedback loop between visualization creator and consumer has broad reaching implications. Web visualizations can be designed to maximize user engagement. If there are portions of a visualization that are never interacted with or interaction features that are never
used, creators can take this into account resulting in interfaces explaining features better or the trimming of unnecessary features. Functionally, every visualization on the web could have a usability study performed with limited effort on the creator’s part. The potential improvement in visualizations as a whole is almost immeasurable.
2.1 Understanding User Interactions through Interaction Logs

Soliciting feedback from users and conducting studies to understand more fully how they interact with interfaces has been a staple of human-computer interaction studies since the advent of the field. A more recent development is trying to understand the user’s interactions from logs of their activity. Insights that can be gained have the potential to explain user behavior [7] and possibly some user characteristics [8, 9]. Analysis done in prior studies show user interests and methods of exploration can be gleaned, though not without some challenges [10, 11]. Even users themselves have found usefulness in viewing their interaction past [12]. While gaining understanding of single users is interesting, the significance of interpreting user interaction logs comes from the generalizations of large studies. The ability to understand, and possibly predict, user interaction patterns adds an additional layer of insight into how humans operate on the web [13–15]. Knowledge and understanding of cognitive patterns on the scale of the number of internet users is indispensable.

2.2 Remotely Monitoring User Interaction

Remote usability testing has been examined along with its potential effectiveness compared to conventional methods [16]. The most effective usability studies have traditionally been ones where researchers remain with the participants. The trade off in significantly lowering the barrier to entry in usability studies with users is a slight reduction in accuracy [16]. It is a natural progression to combine the economies of scale afforded by remote usability testing and user interaction logs, as millions of people use websites and interact with the robust visualizations found there.
Many current methods of monitoring user interactions over the web have been evaluated as inconsistent or incomplete [17]. There is much to understand about how users interact with website content, and standard web server logging is not sufficient to capture the nuances necessary for usability testing [18]. Even simple monitoring of a user's mouse has been shown to uncover actionable areas of improvement [19]. Remote, cohesive monitoring of user interactions has the potential to usher in large improvements in the field of data visualization.

### 2.3 Applications of the Union of Remote Monitoring and Usability Logs

The addition of user interaction logs to novel visualization techniques can improve the process of academic studies. Being able to essentially automate feedback and usability studies goes a long way in evaluating the utility of new tools, in contrast to sending out hundreds of surveys [20]. However, the potential benefits are not limited to the academic sphere. The potential utility of the application of user interaction logs and remote usability testing can be a great boon to enterprise ventures [21, 22] and online news organizations alike [23, 24]. The basic supposition behind all of this research is that information about user interactions are not only useful, but valuable, and its application should be both widespread and utilized.
3.1 Study Objectives

The principle purpose of this research is to determine the efficacy of existing tools at monitoring users interaction with web visualizations. In some respects, this project serves as a proof of concept that such interactions are both trackable and meaningful. The exploration of these tools aims to illuminate the design space, showing where improvements can be made and what features are necessities.

The analysis of user interaction logs performed in this project seek to highlight the importance of a holistic view of user interaction. Honing a visualization based around user feedback is not a new concept, however, the mere fact that feedback can be gleaned from user interaction logs cannot be understated. This research aims to demonstrate that expensive studies and time consuming in-person interviews are not necessary to close the feedback loop. Rather, insights that can be gained from user interaction logs can be effective in providing actionable feedback.

3.2 Instrumentation Evaluation Methodology

The study needed to approach the problem from the base level. The visualization being used as a testbed needed nontrivial data associated with it, but also needed to be simple enough so as to not confuse the user with added features. The data of interest for this study was the captured data from users interacting with a visualization, not exploring dropdown menus, filters, or other common features in visualizations. Additionally, to implement each of the tools evaluated in this study, access to the source code to add user interaction tracking was necessary, which contributed to the choice in visualization.
Each tracking tool was implemented on Lane Harrison’s exoplanets visualization shown in figure 3.1. The visualization was created using D3.js, and is hosted on an HTML webpage. The webpage contained exclusively this visualization to reduce excess visual clutter. This environment mimics the key components of an online web visualization, and the visualization itself holds non-trivial data.

The circles shown in the visualization represent exoplanets. The circles as SVG elements are described by their radius, horizontal position in the SVG, and vertical position in the SVG. Bound to these circles is exoplanet data. Each exoplanet is defined by the exoplanet’s radius, name, atmosphere, distance to the nearest star, and the year it was discovered. When a circle is
3.3. CRITERIA FOR EVALUATING INSTRUMENTATION TOOLS FOR INTERACTIVE DATA VISUALIZATION

hovered over by the mouse, the corresponding exoplanet data appears at the top left corner.

Sessions for evaluation ranged between twenty and thirty seconds in length. Throughout these sessions, multiple exoplanets were interacted with by mouse movement and mouse clicking. The tracking tools were then evaluated based on seven metrics.

3.3 Criteria for Evaluating Instrumentation Tools for Interactive Data Visualization

The following criteria give insight into the usefulness of the tool, how and when it should be applied, as well as its strengths and weaknesses. The task of each tool was to convey as much user information as possible from a twenty to thirty second session of interaction with the exoplanets visualization shown in figure 3.1. The garnered data was evaluated as well. Criteria are rated on a scale from one to ten, with higher numbers corresponding to a better performance from the tool.

**Effort.** The effort associated with a particular tool corresponds to the difficulty in setting up a working implementation. Every tool requires access to the source code of the webpage, and as a result, some level of domain specific knowledge. This measure is therefore more comparative than some of the other criteria.

**Versatility.** This measure takes into account how accessible and usable the data becomes once gathered by the tool. Methods of storing data, whether the data is serializable, and the difficulty incorporated in transferring the data for use in other applications contribute to this score.

**Data.** The data criteria is evaluated based on the hierarchy of data in a web visualization. The scope of this project is only concerned with how users interact with domain elements, SVG elements, and the underlying data. Other data points such as user demographics, locations, or referrals fall outside the aim of this project and are excluded from this evaluation.

**Performance.** Performance in a tool is an integral part of its daily use. Here, performance is associated with the amount of time taken to process large data, the amount of overhead innate in the tool, and the delay from data generation by the user to being able to access the data. It is worth clarifying that a higher value for performance means better performance, lower delays, and less overhead.

**Visualization.** Many of the tools have built-in visualization creation tools, the robustness of which influences this metric. While useful, the visualization capabilities of these tools pale in comparison to dedicated visualization libraries and software. As a result, this metric becomes more important in tools with lower versatility and less essential in tools that integrate well with libraries and applications.

**Interface.** An interface has the capability to make a tool significantly more user friendly. Considerations in this category are number of options available to the user, the complexity and accompanying visual noise, and how well the interface aids users in their task. The goal of this
metric is to identify if the interface needlessly adds complexity and is detrimental to the tool's use.

**Price.** As many of the solutions being compared are created by companies, price becomes a fairly important factor. The level of features provided by the free version of the application, as well as how costly the tool can actually become, should be kept in mind.

### 3.4 Visualizing User Interaction with Sight.js

While the different interaction tracking tools were evaluated, a deep dive was made into the information that user interaction logs on a web visualization could give. The data was gathered from simply recording a brief session of a user interacting with the exoplanets visualization. To give the greatest amount of control over the data, Sight.js was used to track the user's interactions.

#### 3.4.1 Applying Sight.js to an Existing Web Visualization

A handful of user sessions were recorded, and the implementation of Sight.js became iterative. Each time user interaction logs were generated, new ideas to improve the gathered data occurred. Implementation of session identification numbers, timestamps, mouse position, and mouse velocity were added after each successive session. Each data field added a new layer of analysis that could be performed. Each session exported a .json file containing up to one thousand individual logs for sessions less than a minute long. Some of these logs were events triggered on the webpage's body or the moment the webpage was loaded. These logs were largely ignored, not due to lack of usefulness, but rather to make room for the focus of the analysis, the exoplanets.

#### 3.4.2 Insights into User Interaction in Splunk

Splunk is a log aggregation and visualization tool which uses a query language to search, transform, and visualize data. For this study, the .json files from Sight.js were uploaded into Splunk and subsequently visualized. A sample of the Splunk interface is shown in figure 3.2.

Some of the fields added into Sight.js proved less valuable than initially thought. While session identification numbers are useful, this study did not have a number of user sessions that necessitated identification in that manner and as such they added minimal value in this specific instance. Additionally, the implementation of mouse velocity did not end up aiding significantly in meaningful analysis.

A number of the fields did prove to provide interesting insights into the user's session. The timestamp allowed for a timeline of the user's session to be created, showing actions in the order they occurred. Through some deduction, the amount of time a user spent on a single element could be found, a potential indicator of an interesting data point. The addition of tracking mouse position when DOM events fired allowed for a trace of the user's path through the visualization.
The visualizations created in Splunk evolved as more interactions were logged. Some visualizations leaned toward numerical analyses such as averages, counts of occurrences, and maximum values. Others looked for a way to mesh what the user may have been thinking with the reality of the interaction logs.
4.1 Evaluation Strategy in Brief

Each of the following tools for tracking user interaction with web visualizations were implemented using the methodology laid out in the previous chapter. The tools were evaluated based on their performance with a twenty to thirty second interaction session on the exoplanets visualization. The principle criteria judged were the difficulty in setting up the tool, the flexibility of the data from the tool, the level of data reporting, the celerity of the tool in performing the task, the built-in visualization tools, and the price.

Figure 4.1: Google Analytics’ Logo [1].
4.2 Google Analytics

Google Analytics is a web analytics tool provided by Google as a means for customers to track user events and demographics. The primary aim of this tool is to track how users interact with the webpage as a whole and how that interaction relates to other pages.

4.2.1 Effort

Google Analytics requires the insertion of two Javascript snippets into the head of the webpage to be tracked. One snippet provides identification for Google Analytics to find the webpage. The other provides event sending functionality. Google Analytics can be instantiaed using only the identification code, as general information, such as number of concurrent users and their location, can still be tracked. Both of these snippets are displayed in figure 4.2.

```javascript
<script src="https://www.googletagmanager.com/gtag/js?id=UA-123456789-1"></script>
<script>
  window.dataLayer = window.dataLayer || [];
  function gtag(){dataLayer.push(arguments)};
  gtag("js", new Date());
  gtag("config", "ACCOUNT NUMBER");
</script>
```

**Figure 4.2:** The Google Analytics code present in the webpage head.

In order to track events in a visualization, each SVG element requires an event listener. In the event listener, a function must be called which sends a Google Analytics object to the Google Analytics instance as shown in figure 4.3. This functionality is not available unless both Javascript snippets are present in the head of the webpage.

```javascript
(function (a, b, c, d, e) {
  ('google-analytics-object' in b || !b) ? function () {
    e.getElementsByTagName('script')[0].async = !a;
    e.src = 'https://www.google-analytics.com/analytics.js';
  }() : !e.createElement('a') ? function (a, b, c, d, e) {
    b.createElement('script').src = 'https://www.google-analytics.com/analytics.js';
    b.getElementsByTagName('script')[0].async = !a;
  }()
}(true, false, false, false, false));
```

**Figure 4.3:** The Google Analytics code to send a Google Analytic object.

4.2.2 Versatility

Google Analytics can export data into .pdf, .csv, .xlsx, and Google Sheets format. When exported, the data is serializable, which enables the bulk viewing of many data points. However, the data
is aggregated based on the data fields or time, which can make some avenues of analysis either difficult or impossible to pursue.

Output from the exoplanet visualization events are shown in table 4.1. Notice that the export does not denote types of events or labels, though that data is available for analysis using Google Analytics' built-in visualizations.

Table 4.1: Sample Google Analytics Export.

<table>
<thead>
<tr>
<th>Hour Index</th>
<th>Total Events</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>131</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>21</td>
<td>0</td>
</tr>
<tr>
<td>22</td>
<td>0</td>
</tr>
<tr>
<td>23</td>
<td>0</td>
</tr>
</tbody>
</table>

4.2.3 Data

With respect to the data hierarchy of web visualization, Google Analytics captures the top layer, but merely brushes the other two. DOM events can be captured, but only a small portion of SVG data or the underlying data can be seen. This is the result of the Google Analytics object used to send the data. With a limited number of fields and labels, not all of the desired data can be sent in one event.

Samples of data gathered from the exoplanets visualization are shown in figures 4.4 and 4.5. In this instance, the event label is the name of the planet. Due to the nature of the Google Analytics object, only one field of either SVG data or underlying data can effectively be sent.
4.2. GOOGLE ANALYTICS

4.2.4 Performance

As Google Analytics was built with measuring large amounts of traffic in mind, it handles large influxes of events well and loading times are generally short. There is typically a delay between events firing on the webpage and the events appearing in Google Analytics, but other tracking features do appear in real-time.

4.2.5 Visualization

Google Analytics has a built-in visualization tool, though its capabilities are limited. Event data can be visualized with line charts, bar charts, pie charts, and pivot tables. Sample visualizations from the Google Analytics tool are shown in figure 4.6.

4.2.5.1 Interface

The Google Analytics interface is shown in figure 4.7. Many options are provided to the user, making the view visually busy at times.

<table>
<thead>
<tr>
<th>Event Label</th>
<th>Total Events</th>
<th>% Total Events</th>
</tr>
</thead>
<tbody>
<tr>
<td>WASP-19 b</td>
<td>23</td>
<td>17.56%</td>
</tr>
<tr>
<td>WASP-66 b</td>
<td>14</td>
<td>10.69%</td>
</tr>
<tr>
<td>Kepler-13 b</td>
<td>9</td>
<td>6.87%</td>
</tr>
<tr>
<td>WASP-78 b</td>
<td>7</td>
<td>5.34%</td>
</tr>
<tr>
<td>Kepler-7 b</td>
<td>6</td>
<td>4.58%</td>
</tr>
<tr>
<td>POT9-1 b</td>
<td>5</td>
<td>3.82%</td>
</tr>
<tr>
<td>XO-5 b</td>
<td>5</td>
<td>3.82%</td>
</tr>
<tr>
<td>GJ 3021 b</td>
<td>4</td>
<td>3.05%</td>
</tr>
<tr>
<td>HD 60532 c</td>
<td>4</td>
<td>3.05%</td>
</tr>
<tr>
<td>WASP-82 b</td>
<td>4</td>
<td>3.05%</td>
</tr>
</tbody>
</table>

Figure 4.4: The Google Analytics event label table.

<table>
<thead>
<tr>
<th>Event Action</th>
<th>Total Events</th>
<th>% Total Events</th>
</tr>
</thead>
<tbody>
<tr>
<td>mouseMove</td>
<td>76</td>
<td>58.02%</td>
</tr>
<tr>
<td>mouseOut</td>
<td>25</td>
<td>19.08%</td>
</tr>
<tr>
<td>mouseOver</td>
<td>25</td>
<td>19.08%</td>
</tr>
<tr>
<td>mouseClick</td>
<td>5</td>
<td>3.92%</td>
</tr>
</tbody>
</table>

Figure 4.5: The Google Analytics action table.
4.2.6 Price

Google Analytics has a free version which was used for this project. The paid version primarily provides increased integration with other applications to monitor monetization of the webpage and increased data storage. The price for Google Analytics 360 is $150,000 USD a year [25].
4.3 Session Stack

Session Stack is a web debugging tool that records user sessions to provide insight into how users interact with a webpage by seeing the session played out in full.

4.3.1 Effort

Session Stack requires a Javascript snippet in the head of the webpage in order to track and record the user’s session, seen in figure 4.9. In addition to this code snippet, to gain detailed data on the events occurring on the visualization, each SVG element must have an event listener which sends a log containing data, seen in figure 4.10. These are the only two additions to the code required to implement Session Stack.

```javascript
// Start Session Stack code

// Function to append Session Stack script
function(sessionStack) {
  (function(){
    var c=document.createElement('script');
    c.setAttribute('data-stack', sessionStack);
    document.head.appendChild(c);
    c.onload = function() {
      // Code for processing data
      console.log(data);
    }
  }());
}

// Start Session Stack code

// Function to append Session Stack script
function(sessionStack) {
  (function(){
    var c=document.createElement('script');
    c.setAttribute('data-stack', sessionStack);
    document.head.appendChild(c);
    c.onload = function() {
      // Code for processing data
      console.log(data);
    }
  }());
}

// End Session Stack code

// Function to append Session Stack script
function(sessionStack) {
  (function(){
    var c=document.createElement('script');
    c.setAttribute('data-stack', sessionStack);
    document.head.appendChild(c);
    c.onload = function() {
      // Code for processing data
      console.log(data);
    }
  }());
}

// End Session Stack code
```

Figure 4.9: The Session Stack code present in the webpage head.

Figure 4.10: The Session Stack code present in the event listeners.
4.3.2 Versatility

Session Stack stores its sessions as HTML files. The session consists of a video and a log of events. The logs cannot be separated from the video natively. As a result, the data is not serializable, and to gain insight into user interaction each user session must be watched one at a time. The session can be downloaded as an HTML file, but the data is not easily parsable by other conventional analysis tools.

4.3.3 Data

The data captured by Session Stack reaches the entirety of the data hierarchy of web visualization. It is worth noting that in the event listener found in figure 4.10 D3 objects can be sent. DOM events, SVG data, and the underlying data can all be captured by the tool, however the limited versatility greatly impacts the use of this data.

4.3.4 Performance

Session Stack’s uniqueness attributes to much of the overhead associated with the tool. No other tool allows for the actual viewing of the user’s session, however, the sending of a screen capture requires more data to be sent to achieve a similar end when compared with other tools. Additionally, when viewing a user’s session, a large influx of events can cause stuttering. With this in mind, the delay between the user’s session and it being viewable from Session Stack was merely seconds.

4.3.5 Visualization

Session Stack does not have a built-in visualization tool.

4.3.6 Interface

The interface of Session Stack is very unique. A log of events appears on the left side of the screen that is searchable. On the right side of the screen, the user’s session video plays. Session Stack also allows for the viewing of all events that have occurred across all sessions. Both of these interfaces are shown in figures 4.11 and 4.12 below.

4.3.7 Price

Session Stack charges based on the number of sessions per month. This project used the free version, which allows for less than one thousand sessions a month. Prices increase to $99, $199, $399, and $599 for ten thousand, twenty-five thousand, one hundred thousand, and two hundred fifty thousand sessions per month, respectively [26].
4.4 Sight.js

Sight.js is a Javascript library being created by Lane Harrison for the purpose of extracting user interaction data for web visualizations.

4.4.1 Effort

As a library, Sight.js requires an import into the webpage in order to function. Like the other tracking tools, Sight.js utilizes event listeners on SVG elements to gather data, but rather than calling functions to send the data to a server, Sight.js exports locally.
4.4.2 Versatility

Sight.js exports all of the collected data to a .json file. This format allows for easily serializable data that can be read by most third party libraries or applications. The size of the file has the potential to become unwieldy as a direct result of the sheer number of events collected, which is worth bearing in mind when deciding on events to track.

4.4.3 Data

Sight.js reaches each layer of the data hierarchy of web visualizations. DOM events, SVG data, and the underlying data are all captured. A sample of the data output by Sight.js is shown in figure 4.13.

```json
{"sessionID":"jBzpn6808VweCrksAAAA",
"time":1512689700977,
"x-position":544,
"y-position":441,
"x-velocity":544,
"y-velocity":441,
"abs-velocity":708.2977937991808,
"type":"mousemove",
"packageName":"planets",
"className":"HD 4203 b",
"radius":"13.53",
"distance":"0.99",
"year":"2001",
"atmosphere":"hydrogen-rich",
"depth":1,
"value":183.06089999999998,
"r":16.06321449721978,
"x":406.31390787125713,
"y":126.2663301677847,
"element":"circle"},
```

Figure 4.13: A sample interaction log generated by Sight.js.

4.4.4 Performance

Sight.js has little overhead being a Javascript library. The data file is generated with a very short delay. The amount of data has little effect on the performance of Sight.js, as it does not need to read or visualize the data.

4.4.5 Visualization

Sight.js does not have a built-in visualization tool.
4.5. MIXPANEL

4.4.6 Interface

Sight.js does not have an interface.

4.4.7 Price

Sight.js does not currently have a price.

4.5 Mixpanel

Mixpanel is a business analytics tool aimed at tracking user interaction with webpages, then visualizing the resulting patterns and behaviors.

4.5.1 Effort

Mixpanel requires only two Javascript additions to a webpage to function. The first is a Javascript snippet in the head of the webpage to provide tracking information demonstrated in figure 4.15.

![Figure 4.14: Mixpanel’s Logo [3].](image)

![Figure 4.15: The tracking code present in the head of the webpage for Mixpanel.](image)

The second addition required is an event listener on each SVG element to be tracked. This event listener calls a function which sends key-value pairs of labels and data to Mixpanel as seen in figure 4.16.
CHAPTER 4. COMPARING INSTRUMENTATION TOOLS FOR INTERACTIVE DATA VISUALIZATIONS

Figure 4.16: A sample code snippet present in the event listeners of the webpage for Mixpanel.

4.5.2 Versatility

Mixpanel can export data to a .csv file, which makes it easily incorporated to most third party libraries and applications for further analysis. The exported file provides a row for each event, making it serializable as well.

4.5.3 Data

Mixpanel is able to track each layer of the data hierarchy of web visualizations. DOM events, SVG data, and the underlying data can all be sent to the Mixpanel instance. However, each data field has to be individually recorded in a key-value pair. This requires each field of the underlying data to be manually labeled in the function that sends the data to Mixpanel. While not impossible, with large data, this has the potential to become an impractical task.

4.5.4 Performance

Mixpanel is able to handle a large number of events at one time, and does so with very little delay. With only one Javascript snippet in the head of the webpage, the overhead on the webpage itself is not very large.

4.5.5 Visualization

By far, Mixpanel has the most robust built-in visualization tool. It allows for grouping by multiple fields in bar charts, line charts, and tables. Examples of these visualizations are shown in figures 4.17, 4.18, 4.19.

4.5.6 Interface

Mixpanel’s interface has a moderately simple design, offering the user a lot of functionality with limited visual clutter. The interface is shown in figure 4.20.

4.5.7 Price

Mixpanel offers a free version, which was used for this project. The paid version allows for more data storage, additional features pertaining to the built-in visualizations, and the ability to export the data to .csv. The price for the StartUp package is $999 a year, and the Enterprise package price is customizable [27].

22
4.6. NUMERICAL EVALUATION OF ALL TOOLS

After an analysis of the evaluation criteria, numerical values were assigned for each tool's performance in the corresponding category. These numbers serve as a comparative measure to provide a more concrete reference point to distinguish the strengths, weaknesses, and capabilities of each of the tools. The rationale behind the numerical assignments of each category has been outlined in the previous section, and should be kept in mind when viewing the comparative measures found in table 4.2.
CHAPTER 4. COMPARING INSTRUMENTATION TOOLS FOR INTERACTIVE DATA VISUALIZATIONS

Figure 4.19: A line chart of events grouped by planet over time.

Figure 4.20: A portion of the Mixpanel interface.

This study does not posit that a particular tool is the definitively superior tool. Instead, this study seeks to inform the reader of the capabilities and shortfalls of the evaluated tools, to enable informed decisions. Each tool was constructed for a specific purpose, which influenced the design decisions of its creators, and ultimately the space in which the tool excels.
## 4.6. NUMERICAL EVALUATION OF ALL TOOLS

<table>
<thead>
<tr>
<th>Evaluation Metric</th>
<th>Google Analytics</th>
<th>Session Stack</th>
<th>Sight.js</th>
<th>Mixpanel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effort</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Versatility</td>
<td>4</td>
<td>3</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>Data</td>
<td>4</td>
<td>6</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>Performance</td>
<td>6</td>
<td>3</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>Visualization</td>
<td>6</td>
<td>N/A</td>
<td>N/A</td>
<td>7</td>
</tr>
<tr>
<td>Interface</td>
<td>6</td>
<td>7</td>
<td>N/A</td>
<td>7</td>
</tr>
<tr>
<td>Price</td>
<td>4</td>
<td>7</td>
<td>N/A</td>
<td>5</td>
</tr>
</tbody>
</table>

**Table 4.2: Numerical Evaluation of Metrics**
5.1 Visualizing User Interaction Logs with Splunk

The following samples of visualizations of user interaction with the exoplanets visualization were achieved from using the recorded interaction data from Sight.js with Splunk, a log aggregation and visualization tool. Of the tools explored here, only Sight.js and Mixpanel have the ability to produce these visualizations, as they track all three data layers and have the versatility to export a file which can be used with other software or libraries to increase their effectiveness.

The table shown in figure 5.1 displays the data from the exoplanets visualization. All three layers of data are represented here: mouse events from the DOM, the locations of circles in the SVG, and the exoplanet data.

![Figure 5.1: Interaction logs gathered by Sight.js, displayed in Splunk.](image)

Early visualizations of the user interaction data focused on the physical location of a user's mouse. Interaction was being tracked, but very few meaningful conclusions could be drawn from these visualizations. A center of focus was the frequency with which mouse coordinates were being visited as shown in figure 5.2, as a way to see if users interacted with the visualization in straight line patterns.
5.1. VISUALIZING USER INTERACTION LOGS WITH SPLUNK

One of the initial visualizations, mouse X and Y coordinate frequencies.

Figure 5.2: One of the initial visualizations, mouse X and Y coordinate frequencies.

One of the more enduring visualizations was figure 5.3. This visualization shows to some extent the amount of time the user spent either on or between elements. The disparity between the number of mouse moves and other mouse actions can make this visualization somewhat tricky to read, as all the data falls on the same axis.

Figure 5.3: A timechart of actions the user took.

The concept of visualizing the tracing the user’s path came fairly early on as well. Initial implementations of this visualization, like the one shown in figure 5.4, was a reflection on the y-axis of the user’s actual interaction. On an SVG element, the coordinate (0, 0) is at the top left corner of the element. As the user progresses from left to right, the x coordinate increments. As the user progresses from top to bottom, the y coordinate increments. Since the Splunk visualization’s origin is placed in the bottom left corner, the resulting visualization was one which traced a reflected image of the path of the mouse. Once this error was noticed, it was fixed in future iterations.

An exploration was made into the relationship between statistics and the user’s interaction. A few different methods of representing the overall count of mouse events were tried. To obtain a more clear idea of the exact value for each event, the numbers themselves were displayed to avoid the issues that plagued the visualization in figure 5.3. Sparklines also proved to be an interesting addition to the simple number representation. In a similar vein, the average absolute velocity of the mouse was captured and represented numerically, matched with their respective session number. Examples of these numerical findings are shown in figure 5.5.

User sessions can also be compared to one another as a result of session identification numbers. Users with high levels of interaction can then have their sessions examined, seeing where the points of interest are. Users with low levels of interaction can then be compared, to determine if the same elements are interacted with. Comparisons like these could be used to categorize user
interests and initiate the process to raise engagement of low interaction users. An overview of user interactions broken down by sessions is shown in figure 5.6.

The theme of visualizations ended up shifting toward exposing interesting occurrences of user interaction, hoping to explain why the user acted the way that they did. An essential piece of that understanding is the inclusion of the underlying data into the visualizations. Rather than focusing exactly on where the mouse cursor was, attention was paid to the element the user had interacted with. By using all three levels present in the web visualization data hierarchy, figure 5.7 is able to show the most visited exoplanets. This is achieved by establishing the sequence of
5.2 Applications of Visualized User Interaction Logs

A mouse over, any number of mouse moves, and then a mouse out as being a unique visit to a planet. The exoplanet names are sent in the interaction logs when a user visits one of the circle SVG elements. Without all three data layers, this deep analysis is not possible.

![Image of the top ten most visited planets.](image)

**Figure 5.7: The top ten most visited planets.**

A number of other interesting features arose from the data. Figure 5.8 compares the radius of an exoplanet to the number of events fired.

![Image of exoplanet radius and count of DOM Events.](image)

**Figure 5.8: Exoplanet radius and count of DOM Events.**

In theory, if all exoplanets were of equal interest, exoplanets with larger radii would have more events occurring as a direct result of their surface area. In this user’s session, that was not the case. This begs the question, why was the user’s most interacted with planet not the largest? There is an entire range of potential answers. Perhaps the user was learning how the visualization worked by using that particular planet, or perhaps it was coincidence. From a single session, no concrete conclusions can be drawn. However, the large scale studies that can be enabled by user interaction logs have the potential to widen understanding of what users find interesting and engaging.

5.2 Applications of Visualized User Interaction Logs

The increase in visibility of user trends through the use of visualization allows for an easier understanding of data points that are of interest for users. The interpretations gained from
viewing user interaction can be used to tailor visualizations to specific audiences or demographics. Actionable goals to improve visualizations on the web can be made and implemented. Further examples of visualizations of user interactions can be found in Appendix A.
6.1 Patterns Across Solutions

When different people approach the same task, there are bound to be differences in the methods employed to accomplish it. This design space was no exception. Despite this, there were a large number of similarities in the approach taken by many of these tools. For both Google Analytics and Mixpanel, the goal of the tool is to track user interaction on the page in general and to find trends among users. Session Stack has a different focus, tracking the way users interact with a page to see what went wrong. Sight.js opts for specifically tracking user interaction within visualizations. These foci influence the implementation of the tool.

Across the different tools, similar themes occurred. Each tool, aside from Sight.js, had a web interface. This increases usability for those who are not domain experts, but increases the complexity of implementation, bouncing back and forth between the webpage code and the web interface. Another common thread across Google Analytics and Mixpanel was the implementation of visualization creation in the interface. Having a built-in visualization tool is helpful for simple and quick analysis of the data, however, they pale in comparison to dedicated visualization tools. As a result, to truly dig into the user’s interaction data, it becomes necessary to export the data from the tracking tool used. This flexibility in how the data is used became an essential piece in understanding the user’s motives, which not every tool provided.

Each of tools explored used event listeners to gain a clearer picture of the user’s interaction. For the three market solutions, functions that send the data to external servers are required. The three functions accepted different types of data: a predefined object, any number of key-value pairs, and any single object. At first glance, these three parameters do not seem to cause for much divergence. However, the ease of use to convey SVG and library level data vary greatly.
Google Analytic’s predefined object allowed for only the limited sending of some fields. In the study, the Event Label field of the Google Analytics object was used to send the exoplanet’s name, but conveying other data was either not possible or muddled the picture more by placing data where it did not necessarily belong. Mixpanel took a different approach, having data sent in key-value pairs in order for it to be filterable. This makes it very possible to collect DOM, SVG, and library level data, however, the person implementing Mixpanel needs prior knowledge of what data is desired. Each individual field can be sent with a label, but this adds either an additional layer of decision making or more work labelling every field. Session Stack was able to send the SVG object in question along with the bound data in a single object. This allows for all the data to be sent at once, rather than field by field. Sight.js uses the SVG object the same way, although it does not send it to an external server. This difference in data conveyance surprisingly goes a long way in easing implementation and improving the insights possible. Further tools developed in this space need to be able to send the SVG object with bound data without defining each field to be sent in order to achieve peak performance.

6.2 Insights from Exploring the Design Space

As a result of this study, an new awareness of the implications of the implementation of user interaction tracking was achieved. Among the forefront was the potential value of automatic instrumentation. Lowering the barrier of entry into obtaining feedback from users, while increasing the quantity of feedback increases the likelihood of its use. The implication is that every visualization on the web has the potential to practically have its own usability study, so that it can be incrementally improved over its lifespan.

Another interesting insight is the potential use as a mouse tracker to examine how users explore a webpage. A user’s path through a visualization was able to be traced and recreated, so in theory, a similar method could be employed to track webpage traversal. While mouse tracking is not a new concept, the binding of it with increased visibility into user interaction and data could provide an attractive pairing for web developers.

6.3 Benefits of a Visualization Specific Tool

Information about how a user interacts with a web visualization has a limited usefulness when viewed as just a list of events or fields. It is a natural progression to take the gathered interaction data and visualize it in some manner. As a result, the most important metrics of these interaction tracking tools are their versatility and quality of data collection. The entire purpose of this design space is to be able to analyse a user’s interaction with a visualization, a feat most easily achieved through a visualization specific tool.

It is beneficial for the tool to be aware of the type of data it is given, rather than agnostic. Not only does this awareness aid in parsing user interactions by being able to simply pass D3
objects, rather than the values for each field individually, but it also guarantees that all three layers of data will be captured. Only with a gestalten picture from the DOM, SVG elements, and underlying data can meaningful insights be garnered from user interaction.
7.1 Impact

Tracking of user interaction on web visualizations makes large scale asynchronous studies on the way people interact with visualizations feasible. Lowering the barrier to entry and difficulty to perform those types of studies would increase the rate of advancement in visualization. Web visualizations can more easily be tailored to users, through the use of this new feedback, and may result in higher visualization literacy through increased interest.

7.2 Conclusion

The design space of web visualization tracking is relatively barren. Not every existing tool is capable of gathering each level of the web visualization data hierarchy. All three layers are necessary to gain understanding of the user’s interaction and elicit feedback from the resulting data. In addition it needs to be serializable in order to perform large scale analysis. Without these key pieces, any user interaction data is functionally useless.
APPENDIX A: SPLUNK VISUALIZATIONS FROM SIGHT.JS DATA

Figure 1: Manhattan distance dissimilarity function plotted against count of DOM events.

Figure 2: A multiseries timechart of user interactions with the exoplanets visualization. Note that the use of multiple series aids readability when compared to figure 5.3.
Figure 3: Tracking of mouse movements and categorization by event type. This became the more robust implementation of figure 5.4.

Figure 4: Pie charts of DOM events occurring on individual planets.


[25] “Google analytics free and 360 comparison,” google.com/analytics/analytics/compare/.
