PRACTICAL DATA* VISUALIZATION WITH JAVASCRIPT

WHEN, *WHAT, AND HOW

Ceily Boyd

ceily_boyd@harvard.edu

Program Management @ the Harvard Library
This talk introduces a methodology intended to help you decide:

- *When* to use JavaScript to visualize your data (and when to choose alternate means)
- *What* data, or aspects of your data to render visually, and
- *How* best to use the tools at your disposal to visualize your data
Demand for, and popularity of data analysis, data mining, data-driven visualization has led to ...
...THIS

Image source.
SO MANY OPTIONS

- Proliferation of technologies, languages, platforms, applications, and tools with overlapping functionality
  - Excel (PowerPivot, Pivot Charts) - Google Sheets/Charts
  - Jupyter - MATLAB
  - Octave - OpenRefine - Plot.ly - R - SAS - Stata - Tableau
  - JavaScript - Python - Ruby
  - WebGL - SVG

- Proliferation of graphics and visualization libraries & APIs
  - Bokeh - Chart.js - D3.js - Dygraphs.js - Google Charts API
  - jqPlot.js - Processing.org, Processing.js - VisPy - Vis.js
SO MUCH DATA

• Library-related data:
  - ARL data - Google Analytics - Harvard Library operational data - Harvard Library Open Metadata - Program/project assessment data & results

• Library-related APIs:
  - Dataverse API - DPLA API - HathiTrust - WorldCat API - Library Cloud

• Research data & APIs:
MANY QUESTIONS & CONSIDERATIONS

- When does visualization work best for libraries and library data?
- Data
  - How to model, structure, and manipulate data for visualization
  - How to work with data from different sources
  - How to decide when to use which visualization based upon data
- What makes a successful visualization?
- Concerns about knowledge, skills, and access to technology and training
WE HAVE DATA. WE WANT TO ANALYZE & VISUALIZE IT...WHAT NEXT?

Image source.
This talk will focus on a methodology intended to help you:

- **Choose** amongst potential tools (including JavaScript) to visualize your data

It will use three Library-related examples to:

- **Demonstrate** the methodology in action, and
- **Demonstrate** how to use JavaScript for visualization workflows
AUDIENCE

People who:

• Want to add interactive visualizations to their Web documents (to create Data-Driven Web Documents)
• Are new or relatively new to programming but who are familiar with HTML, CSS, and some JavaScript
• For whom programming is not a frequent part of their daily work
• Have experience with programming but who do not usually work with data, graphics, or visualization
• Need to decide how best to create sustainable visualizations
DATA VISUALIZATION WORKFLOW

METHODOLOGY & FRAMEWORK

Image source.
DATA VISUALIZATION WORKFLOW

Goal: Reporting, Storytelling

Assemble Data  Transform Data  Visualize Data
DATA VISUALIZATION WORKFLOW

COMPONENTS

Data sources
- Sources
  - File
  - Database
  - API
  - Streams,
  - URL
- Formats
  - JSON
  - CSV
  - TSV
  - Binary
  - Function

Data transformers
- Tools
  - Applications
  - Standalone/Embedded Tools
  - Batch/Interactive
  - Libraries
- Types
  - Format conversion
  - Type conversion
  - Subset
  - Anonymize
  - Cleaning

Data visualizer
- Types
  - 2-D
  - 3-D
  - Geospatial
  - Etc.
- Goals
  - Reporting
  - Storytelling
  - Data exploration
WORKFLOW COMPONENTS

DATA SOURCE

A component that provides data to the workflow. It could be a repository for any kind of data, such as a database, a file, an API, a program; or it could be a function. Multiple data sources might be used in a visualization workflow.
WORKFLOW COMPONENTS

DATA TRANSFORMER

A component that performs some kind of data processing, such as a filter or a file format converter. Multiple data transformers may be needed in a workflow to prepare data for visualizers.
WORKFLOW COMPONENTS

DATA VISUALIZER

A component that presents a graphical representation of information or data that is suitable for display on a device (e.g. via a Web browser or app), in a document, or in print.
DATA EXPLORATION WORKFLOW

1. Assemble Data
2. Transform Data
3. Visualize Data
4. Explore Data Interactively
5. Select Data Subset
APPLICABLE TO USE-CASES, SYSTEMS, TOOLS, & LIBRARIES
## Compare Data Analysis & Visualization Options

### Table: Data Analysis & Visualization Tools

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Data sources</th>
<th>Data transformers</th>
<th>Data visualizers</th>
<th>End-to-end Workflow Support</th>
<th>Integration</th>
<th>Scripting Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google Maps API</td>
<td>API</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Google Maps</td>
<td>API</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>GeoCoding API</td>
<td>API</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Library Cloud</td>
<td>API</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Excel</td>
<td>Application</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Google Sheets</td>
<td>Application</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Jupyter</td>
<td>Application</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>OpenRefine</td>
<td>Application</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Plot.ly</td>
<td>Application</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>R</td>
<td>Application</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Tableau</td>
<td>Application</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Google Charts</td>
<td>Application, Library, API</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MySQL</td>
<td>Application/Database</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Manual</td>
</tr>
<tr>
<td>OCLC Datasets</td>
<td>Data repository</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Manual</td>
</tr>
<tr>
<td>Dataverse</td>
<td>Data repository, API</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Manual</td>
</tr>
<tr>
<td>Hathitrust Data &amp; API</td>
<td>Data repository</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Manual</td>
</tr>
<tr>
<td>Ms-Access</td>
<td>Database Development</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Manual</td>
</tr>
<tr>
<td>Processing.org</td>
<td>Environment</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Manual</td>
</tr>
<tr>
<td>Chart.js</td>
<td>JavaScript Library</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Manual</td>
</tr>
<tr>
<td>D3.js</td>
<td>JavaScript Library</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Manual</td>
</tr>
<tr>
<td>DataTables.js</td>
<td>JavaScript Library</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Manual</td>
</tr>
<tr>
<td>D3Graphs.js</td>
<td>JavaScript Library</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Manual</td>
</tr>
<tr>
<td>jqPlot.js</td>
<td>JavaScript Library</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Manual</td>
</tr>
<tr>
<td>Processing.js</td>
<td>JavaScript Library</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Manual</td>
</tr>
<tr>
<td>Vital.js</td>
<td>JavaScript Library</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Manual</td>
</tr>
</tbody>
</table>

*(See handout)*
WORKFLOW CONSIDERATIONS

WHAT TO CONSIDER WHEN BUILDING A VISUALIZATION
IMPLEMENTATION CONSIDERATIONS ARE...

- Related to an individual workflow component
- Related to implementation of a workflow segment
- Related to the entire workflow
IMPLEMENTATION CONSIDERATION CATEGORIES

INCLUDE:

- Legal & Ethical
- Technical
- Usability
- Sustainability
- Aesthetic
COMPONENT CONSIDERATIONS

Component considerations involve:

- Data
- Component inputs & outputs
- Representation of the data within the component
- Presentation of the data
SEGMENT & FULL WORKFLOW CONSIDERATIONS

Segment/Workflow considerations involve:

- Resources: e.g. Skills & knowledge
- Licenses & agreements affecting use, sharing, presentation of results
- Technical: e.g. compute capacity, bandwidth, security, scalability
- Sustainability, reuse, automation, preservation/reproduction of results
# SUMMARY OF COMPONENT & WORKFLOW CONSIDERATIONS

<table>
<thead>
<tr>
<th>Category</th>
<th>Data sources</th>
<th>Data transformers</th>
<th>Data visualizers</th>
<th>Workflow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Legal &amp; Ethical</td>
<td>Restrictions on access/use</td>
<td>Restrictions on use</td>
<td>Restrictions on use</td>
<td>Licensing of application, plugins, and/or components</td>
</tr>
<tr>
<td></td>
<td>Sensitive content</td>
<td>Ethical concerns about transformation of data</td>
<td>Ethical concerns about presentation of data</td>
<td>Content-based restrictions on sharing presentation and/or results</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Restrictions on transformation of data</td>
<td>Restrictions on presentation of data</td>
<td>Access/use restrictions on sharing presentation and/or results</td>
</tr>
<tr>
<td>Technical</td>
<td>API limits</td>
<td>Relevant APIs</td>
<td>Compatible with dataset size</td>
<td>Application may require plugins</td>
</tr>
<tr>
<td></td>
<td>Data quantity</td>
<td>Impact upon data quality</td>
<td>Compatible with data type</td>
<td>Constraints on sharing results of workflow with others</td>
</tr>
<tr>
<td></td>
<td>Data complexity</td>
<td>Tied to use within system</td>
<td>Tied to use within system</td>
<td>Format of data outputs must align with expected input format for</td>
</tr>
<tr>
<td></td>
<td>Data format</td>
<td>Standalone option available</td>
<td>Standalone option available</td>
<td>transformers or visualizers</td>
</tr>
<tr>
<td></td>
<td>Tied to use within system</td>
<td>Compatible with dataset size</td>
<td>Support for viewing on multiple devices, in multiple environments</td>
<td>Workflow is scalable for large quantities of data (e.g. bandwidth,</td>
</tr>
<tr>
<td></td>
<td>Has standalone option</td>
<td>Compatible with data type</td>
<td></td>
<td>compute resources)</td>
</tr>
<tr>
<td>Usability</td>
<td>Useful as-is</td>
<td>Requires special skills/knowledge</td>
<td>Accessibility standards</td>
<td>Requires special skills/knowledge to implement</td>
</tr>
<tr>
<td></td>
<td>Ability to select subsets</td>
<td>Supports desired functionality (e.g. selecting subsets; exporting data)</td>
<td>Visuals support user engagement to the degree required</td>
<td>entire workflow</td>
</tr>
<tr>
<td>Sustainability</td>
<td>Time limit on access/use</td>
<td>Time limit on access/use</td>
<td>Time limit on access/use</td>
<td>Workflow save/export/reuse support</td>
</tr>
<tr>
<td></td>
<td>Data timeliness</td>
<td>Requires special skills</td>
<td>Requires special skills</td>
<td>Results save/export/reuse support</td>
</tr>
<tr>
<td></td>
<td>Vendor support commitment</td>
<td>Uses recognized standards to implement</td>
<td>Uses recognized standards for implementation</td>
<td>Restrictions on save/export/reuse of workflow and/or results</td>
</tr>
<tr>
<td></td>
<td>Active user community</td>
<td>Vendor support commitment</td>
<td>Vendor support commitment</td>
<td>Vendor support commitment</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Active user community</td>
<td>Active user community</td>
<td>Active user community</td>
</tr>
<tr>
<td>Aesthetic</td>
<td>N/A</td>
<td>N/A</td>
<td>Configuration and styling support</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Visuals support the story/report</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Visuals are engaging/compelling</td>
<td></td>
</tr>
</tbody>
</table>
EVALUATING VISUALIZATIONS
TOWARDS A STRATEGY

Network Graph 'Hair Ball', Source.
VISUALIZATION EVALUATION CRITERIA

1. Data overview
   - The capacity of a visualizer to provide a global view of all significant data elements simultaneously
2. Data navigation
   - The ease with which a user can use the visualizer to locate/engage with a specific data element
3. Expressive power
   - The capacity of a visualizer to communicate the full range of relationships within the dataset
4. Aesthetics
   - The visual utility and intuitive value associated with the visualizer

IMPLEMENTING VISUALIZATION WORKFLOWS WITH JAVASCRIPT
BENEFITS OF JAVASCRIPT

- Established Web technology
  - Ubiquitous browser support
  - Easy integration with UI elements for building applications (e.g. jQuery)
  - Variety of frameworks for building applications
  - Many visualization and data manipulation libraries
- Standalone console & server-side support via Node.js
  - Many data manipulation libraries
  - Many APIs/libraries for connecting to data sources (e.g. MySQL, MS-Access, Library Cloud, Dataverse)
BENEFITS OF JAVASCRIPT (CON'T)

- One language for data acquisition, analysis, visualization, application development, and document delivery
- End-to-end visualization workflow support
- Natural fit for Data-Driven Documents
- Useful for prototyping and production
- No licensing fees
# SUMMARY OF JAVASCRIPT CONSIDERATIONS

<table>
<thead>
<tr>
<th>Category</th>
<th>Data sources</th>
<th>Data transformers</th>
<th>Data visualizers</th>
<th>Workflow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Legal &amp; Ethical</td>
<td>N/A</td>
<td>Libraries are often Open Source</td>
<td>Libraries are often Open Source</td>
<td>JavaScript and Node.js are free to use</td>
</tr>
<tr>
<td>Technical</td>
<td>Many data sources have JavaScript APIs or are accessible via Web services</td>
<td>JavaScript can be used to transform data in the browser or standalone (Node.js)</td>
<td>Many JavaScript visualizers and graphics library options</td>
<td>JavaScript skills are a common extension of Web skills</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JavaScript libraries support many types of data transformations</td>
<td>JavaScript libraries exist for WebGL and SVG</td>
<td>JavaScript can be run in the browser, server, or as a console app</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Many JavaScript data import libraries</td>
<td></td>
<td>One language can span the entire workflow</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Node.js can be to process large quantities of data outside the browser</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Usability</td>
<td>Many data sources support XML or JSON output which JavaScript handles well</td>
<td>JavaScript data structures are easy to manipulate and are human-readable</td>
<td>JavaScript can support user interaction with UI elements and visual elements</td>
<td>Requires JavaScript skills to implement entire workflow</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tools exist for easy-viewing of JSON files</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sustainability</td>
<td>JavaScript ubiquity/popularity drives vendor commitment and user community</td>
<td>Many Open Source options</td>
<td>Many Open Source options</td>
<td>Popular scripting language</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Libraries often use recognized standards</td>
<td>Vendor support commitment</td>
<td>Downside: Limited workflow save/export/reuse support</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Vendor support commitment</td>
<td>Active user community</td>
<td>Vendor support commitment</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Active user community</td>
<td></td>
<td>Active user communities</td>
</tr>
<tr>
<td>Aesthetic</td>
<td>N/A</td>
<td>N/A</td>
<td>Depends upon the library</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>JavaScript support easy configuration through objects</td>
<td></td>
</tr>
</tbody>
</table>

(See handout)
EXAMPLES

JAVASCRIPT FOR DATA-DRIVEN DOCUMENTS
THREE HARVARD LIBRARY-RELATED EXAMPLES

1. Selected Datasets for the Towards a Collections & Content Development Strategic Plan Project (TCCDS) (HTML & CSS, with JavaScript & Node.js)
2. Selected Datasets for the Colonial North American Project (HTML & CSS, with JavaScript & Node.js)
3. OASIS Timeline Viewer Application (Data exploration: Browser only)
TOWARDS A COLLECTIONS & CONTENT DEVELOPMENT STRATEGIC PLAN DATASETS

DEMO
TCCDS WORKFLOW & CONSIDERATIONS

Workflow

Data sources
- Email
- Word
- Excel (TSV)

Data transformers
- TSV
- Node.js

Visualizers
- jqPlot.js
- Bubble chart
- Bar chart
- DataTables

Considerations

<table>
<thead>
<tr>
<th>Data sources</th>
<th>Data transformers</th>
<th>Visualizers</th>
<th>Workflow</th>
</tr>
</thead>
</table>
| - Interviews contain sensitive data  
- Tabular data is easy to work with | - Data is primarily textual, not numeric  
- Requires custom transformers | - Many options  
- Size of datasets has impact on visual design | - Manual component integration  
- Management of multiple visualizers |
COLONIAL NORTH AMERICAN DATASETS

DEMO
CNA WORKFLOW & CONSIDERATIONS

Workflow

Data sources
- MS-Access
- Excel (TSV)

Data transformers
- MS-Access
- TSV -> JSON (Node.js)

Visualizers
- Chart.js
- Radar plot
- jqPlot.js
- Bar chart

Considerations

Data sources
- Multiple data sources
- Medium-sized dataset (5,000+ data points)
- Tabular data is easy to work with

Data transformers
- Mixed text, numerical, and categorical data
- Data is primarily textual, not numeric
- Requires custom transformers
- JavaScript + MS-Access

Visualizers
- Many options
- Size of datasets has impact on visual design

Workflow
- Manual component integration
- Management of multiple visualizers
- Project is ongoing so datasets will be updated frequently
OASIS TIMELINE VIEWER

DEMO
OASIS VIEWER WORKFLOW & CONSIDERATIONS

Workflow

Data sources
- CSV

Data transformers
- JavaScript in the browser
- Heavy use of JS libraries

Visualizers
- jqPlot.js
- Bar chart
- Vis.js
- Timeline
- DataTables

Considerations

Data sources
- Data source will be retired

Data transformers
- Data is primarily textual, not numeric
- Requires custom transformers
- Multiple visualizers required multiple data formats
- Complex data structures & parsing required

Visualizers
- Many options
- Visualizers supported easy configuration via JavaScript
- Size of dataset has impact on usability

Workflow
- Manual component integration
- Management of multiple visualizers
- Management of user interaction
- Workflow is entirely encapsulated in the application
/**
 * Create the drivers bubble chart of number of mentions
 *
 * @param {String} String containing JSON/TSV data
 * @return {Void}
 */

function createVisualizationFunction(dataString) {
  // parse the string and create JSON objects
  var data = JSON.parse(dataString);

  // process the data to get it into the proper format for the target visualizer
  var processedData = processData(data);

  // create the visualizer, initialized with processedData, adding
  var visualizer = createVisualizerUsingLibraryFunction('visualization');
QUESTIONS?
THE END

Thanks to Bobbi, Reinhart, and Gloria for the invitation, and...

Thanks to you all for participating!
RESOURCES
WEB AND JAVASCRIPT RESOURCES

• General
  ▪ Mozilla Web technology for developers. Includes online references for HTML, CSS, JavaScript (client-side), Web APIs, Graphics (SVG, WebGL), MathML

• JavaScript
  ▪ Mozilla JavaScript Language Reference
  ▪ JavaScript Design Patterns
  ▪ Secrets of the JavaScript Ninja. John Resig, Bear Bibeault
  ▪ JavaScript: the Definitive Guide. David Flanagan
  ▪ jQuery in Action. Bibeault, Katz
VISUALIZATION RESOURCES

- Visualize This: the FlowingData Guide to Design, Visualization, and Statistics. Nathan Yau
- Duke Library’s Introduction to Visualization LibGuide
- Sunlight Foundation’s Visualization Style Guide
- Data Visualization Catalogue
NODE.JS RESOURCES

- About Node.js
  - Open-source, cross-platform JavaScript runtime environment for developing server-side Web applications
  - Uses **Chrome V8 JavaScript engine**, implemented in C++
  - Runs on: Windows XP and later, Mac OS X 10.5+, Linux
  - Server-side, Event-driven, Asynchronous, Many modules available
- Node.js home page and docs
- Felix's Node Guide
- Node.js @ Lynda.com
- Node Beginner Book
- Node.js the Right Way. Jim Wilson
- Node.js in Action. Cantelon, Harter, Holowaychuk, Raillich