Data Visualization (CSCI 627/490)

D3

Dr. David Koop
Tasks

What?

Why?

How?

<table>
<thead>
<tr>
<th>Actions</th>
<th>Why?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analyze</td>
<td>All Data</td>
</tr>
<tr>
<td>→ Consume</td>
<td>→ Trends</td>
</tr>
<tr>
<td>→ Discover</td>
<td></td>
</tr>
<tr>
<td>→ Present</td>
<td>→ Outliers</td>
</tr>
<tr>
<td>→ Enjoy</td>
<td></td>
</tr>
<tr>
<td>→ Produce</td>
<td>→ Features</td>
</tr>
<tr>
<td>→ Record</td>
<td></td>
</tr>
<tr>
<td>→ Derive</td>
<td></td>
</tr>
<tr>
<td>Search</td>
<td>Attributes</td>
</tr>
<tr>
<td>Target known</td>
<td>One</td>
</tr>
<tr>
<td>Target unknown</td>
<td>Many</td>
</tr>
<tr>
<td>Location known</td>
<td>→ Distribution</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Location unknown</td>
<td>→ Dependency</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Explore</td>
<td>→ Correlation</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Query</td>
<td>Network Data</td>
</tr>
<tr>
<td>→ Identify</td>
<td>Topology</td>
</tr>
<tr>
<td></td>
<td>→ Extremes</td>
</tr>
<tr>
<td>→ Compare</td>
<td></td>
</tr>
<tr>
<td></td>
<td>→ Paths</td>
</tr>
<tr>
<td>→ Summarize</td>
<td></td>
</tr>
</tbody>
</table>

[Munzner (ill. Maguire), 2014]
Visualization for Production

- Generate new material
- Annotate:
  - Add more to a visualization
  - Usually associated with text, but can be graphical
- Record:
  - Persist visualizations for historical record
  - Provenance (graphical histories): how did I get here?
- Derive (Transform):
  - Create new data
  - Create derived attributes (e.g. mathematical operations, aggregation)
Actions: Search

- What does a user know?
  - Lookup: check bearings
  - Locate: find on a map
  - Browse: what’s nearby
  - Explore: where to go
  - Patterns

<table>
<thead>
<tr>
<th>Location known</th>
<th>Target known</th>
<th>Target unknown</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lookup</td>
<td>Browse</td>
</tr>
<tr>
<td>Location unknown</td>
<td>Locate</td>
<td>Explore</td>
</tr>
</tbody>
</table>
Query

- Number of targets: One, Some (Often 2), or All
- Identify: characteristics or references
- Compare: similarities and differences
- Summarize: overview of everything

[Munzner (ill. Maguire), 2014]
Roadmap

- What? → Data
  - Types
  - Semantics
- Why? → Tasks
  - Actions
  - Targets
- How → Vis Idioms/Techniques
  - Data Representation
  - Visual Encoding
  - Interaction Encoding
“Idiom” Comparison

SpaceTree

TreeJuxtaposer

<table>
<thead>
<tr>
<th>What?</th>
<th>Why?</th>
<th>How?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tree</td>
<td>Actions</td>
<td>SpaceTree</td>
</tr>
<tr>
<td></td>
<td>Present, Locate, Identify</td>
<td>Encode, Navigate, Select, Filter, Aggregate</td>
</tr>
<tr>
<td></td>
<td>Targets</td>
<td>TreeJuxtaposer</td>
</tr>
<tr>
<td></td>
<td>Path between two nodes</td>
<td>Encode, Navigate, Select, Arrange</td>
</tr>
</tbody>
</table>


D. Koop, CSCI 627/490, Fall 2022

Northern Illinois University
Assignment 2

- Due Monday
- Process Data
- Create Bar Charts using SVGs and JavaScript
- Interaction: Select by Decade
- Wording Updates
- Skipping 2010?
d3.js

Data-Driven Documents
Data-Driven Documents (D3)

- Open-Source JavaScript Library
- http://d3js.org/
- Original Authors: Mike Bostock, Vadim Ogievestky, and Jeff Heer
- Focus on Web standards, customization, and usability
- Grew from work on Protovis: more standard, more interactive
- By nature, a **low-level** library; you have control over all elements and styles
- A top project on GitHub (over 93,000 stars as of Sept. 2020)
- Lots of impressive examples
  - Bostock was a New York Times Graphics Editor
D3 Key Features

• Supports data as a core piece of Web elements
  - Loading data
  - Dealing with changing data (joins, enter/update/exit)
  - **Correspondence** between data and DOM elements
• Selections (similar to CSS) that allow greater manipulation
• Method Chaining
• Integrated layout algorithms, axes calculations, etc.
• Focus on interaction support
  - Straightforward support for transitions
  - Event handling support for user-initiated changes
D3 Introduction

- Ogievetsky has put together a nice set of interactive examples that show off the major features of D3
  - https://observablehq.com/@dakoop/d3-intro
  - Standalone version: http://dakoop.github.io/IntroD3/
    - (Updated from original)
- Other references:
  - Murray’s book on Interactive Data Visualization for the Web
  - The D3 website: d3js.org
D3 Data Joins

- Two groups: data and visual elements
- Three parts of the join between them: enter, update, and exit
- enter: `s.enter()`, update: `s`, exit: `s.exit()`
Merge vs. Join

• Merge creates a new selection that includes the items from both selections
  - If you want to update all elements (including those just added via enter), use merge!
  - Useful when enter+update have similar transitions
• Join allows you to modify different parts of the selection in a single statement
  - Also will create the final selection
  - Does enter+append and exit+remove automatically
  - Pass functions to modify the enter, update, and exit parts of the selection
  - Examples: https://observablehq.com/@d3/selection-join
Transitions

- Nested transitions (those that "hang off" of a parent transition) follow immediately after the parent transition