Data Visualization (DSC 530/CIS 602-02)

Design and D3

Dr. David Koop
## Tasks

### What?

- **Analyze**
  - Consume
    - Discover
  - Produce
    - Annotate
  - Search
  - Query

### Why?

- **Actions**
  - Consume
  - Present
  - Enjoy

- **Targets**
  - All Data
    - Trends
  - Outliers
  - Features
  - Attributes
    - One
    - Distribution
    - Extremes
  - Many
    - Dependency
    - Correlation
    - Similarity
  - Network Data
    - Topology
    - Paths
  - Spatial Data
    - Shape

### How?

- **Search**
  - Target known
    - Location known: Lookup, Browse
  - Location unknown: Locate, Explore

- **Query**
  - Identify
  - Compare
  - Summarize

---

[Munzner (ill. Maguire), 2014]
Consume
Explore MTA Fare Data
Present Known Information

Each solid circle represents a bee species active in Carlinville, Ill., in both the late 1800s and 2010.

Hatching represents a bee species active in the 1800s but now locally extinct.

The spot where each block rests on the circle indicates one of 26 plant species frequented by these bees.

In the 1880s, scientists observed the following about the bee-plant encounters:
- Present
- Frequent
- Abundant

Studies in 2009 and 2010 showed many bee-plant interactions had changed:
- Lost
- Persisted
- New

[M. Stefaner, 2013]
Enjoy Visualizations of Names

NameVoyager: Explore baby names and name trends letter by letter
Looking for the perfect baby name? Sign up for free to receive access to our expert tools!

Baby Name > [An]

Names starting with 'AN' per million births

[Graph showing name popularity over time]

Produce
Record: Provenance of MTA Data Exploration

D. Koop, DSC 530, Spring 2017
Derived Data

Original Data

Derived Data

trade balance = exports – imports
### Search

#### Search

<table>
<thead>
<tr>
<th></th>
<th>Target known</th>
<th>Target unknown</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location known</td>
<td>![Lookup Icon] (Look up)</td>
<td>![Browse Icon] (Browse)</td>
</tr>
<tr>
<td>Location unknown</td>
<td>![Locate Icon] (Locate)</td>
<td>![Explore Icon] (Explore)</td>
</tr>
</tbody>
</table>

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

---

[Munzner (ill. Maguire), 2014]
Number of targets: One, Some (Often 2), or All
Identify: characteristics or references
Compare: similarities and differences
Summarize: overview of everything
Analysis Example: Derivation

- **Strahler number**
  - centrality metric for trees/networks
  - derived quantitative attribute
  - draw top 5K of 500K for good skeleton


 Task 1

- **What?**
  - **In** Tree
  - **Out** Quantitative attribute on nodes

- **Why?**
  - **Derive**

 Task 2

- **What?**
  - **In** Tree
  - **Out** Quantitative attribute on nodes

- **Why?**
  - **Summarize**
  - **Reduce**

- **How?**
  - **Filter**

[Munzner (ill. Maguire), 2014]
Assignment 1

• http://www.cis.umassd.edu/~dkoop/dsc530/assignment1.html
• Use HTML, CSS, SVG, and JavaScript
• Part 3 will take longer
• Due next Friday (Feb. 10)
• Start soon
“Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively.”
Design Iteration

## Design Iteration

<table>
<thead>
<tr>
<th>Team</th>
<th>Quarterbacks</th>
</tr>
</thead>
<tbody>
<tr>
<td>New York Giants</td>
<td>Eli Manning</td>
</tr>
<tr>
<td>Indianapolis Colts</td>
<td>Peyton Manning, Andrew Luck</td>
</tr>
<tr>
<td>San Diego Chargers</td>
<td>Drew Brees, Philip Rivers</td>
</tr>
<tr>
<td>Baltimore Ravens</td>
<td>Kyle Boller, Joe Flacco</td>
</tr>
<tr>
<td>New England Patriots</td>
<td>Tom Brady, Matt Cassel, Tom Brady</td>
</tr>
<tr>
<td>Green Bay Packers</td>
<td>Brett Favre, Aaron Rodgers, Aaron Rodgers</td>
</tr>
<tr>
<td>New Orleans Saints</td>
<td>Aaron Brooks, Drew Brees, Drew Brees</td>
</tr>
<tr>
<td>Atlanta Falcons</td>
<td>Michael Vick, Matt Ryan, Matt Ryan</td>
</tr>
<tr>
<td>New York Jets</td>
<td>Chad Pennington, Brett Favre, Mark Sanchez</td>
</tr>
<tr>
<td>Cincinnati Bengals</td>
<td>Carson Palmer, Ryan Fitz, Carson Palmer, Andy Dalton</td>
</tr>
<tr>
<td>Houston Texans</td>
<td>David Carr, Matt Schaub, Matt Schaub</td>
</tr>
<tr>
<td>Carolina Panthers</td>
<td>Jake Delhomme, Kyle Orton, Tim Tebow, Peyton Manning</td>
</tr>
<tr>
<td>Denver Broncos</td>
<td>Jake Plummer, Jay Cutler, Tim Tebow</td>
</tr>
<tr>
<td>Arizona Cardinals</td>
<td>Matt Leinart, Kurt Warner</td>
</tr>
<tr>
<td>Jacksonville Jaguars</td>
<td>Byron Leftwich, David Garrard</td>
</tr>
<tr>
<td>Detroit Lions</td>
<td>Joey Harrington, Jon Kitna</td>
</tr>
<tr>
<td>Tampa Bay Buccaneers</td>
<td>Chris Simms, Bruce Gradkowski, Jeff Garcia</td>
</tr>
<tr>
<td>Dallas Cowboys</td>
<td>Drew Bledsoe, Tony Romo, Tony Romo</td>
</tr>
</tbody>
</table>

Design Iteration

Design

• Unlike a math problem, there are many different approaches for the visualization of some data
• Need to have some way to discuss how to determine whether a visualization is doing what we want
• Validation: Understand why a design is effective
  - What problems can be effective
  - Do this at different levels
Four Nested Levels of Design

- Domain situation
- Data/task abstraction
- Visual encoding/interaction idiom
- Algorithm

[Munzner, 2014]
Potential problems at each level

- **Domain situation**
  - You misunderstood their needs

- **Data/task abstraction**
  - You’re showing them the wrong thing

- **Visual encoding/interaction idiom**
  - The way you show it doesn’t work

- **Algorithm**
  - Your code is too slow

[Munzner, 2014]
Validation at each level

- **Threat** Wrong problem
- **Validate** Observe and interview target users

- **Threat** Wrong task/data abstraction

  - **Threat** Ineffective encoding/interaction idiom
  - **Validate** Justify encoding/interaction design

  - **Threat** Slow algorithm
  - **Validate** Analyze computational complexity

  - **Validate** Implement system

  - **Validate** Measure system time/memory

- **Validate** Qualitative/quantitative result image analysis
  - Test on any users, informal usability study

- **Validate** Lab study, measure human time/errors for task

- **Validate** Test on target users, collect anecdotal evidence of utility

- **Validate** Field study, document human usage of deployed system

- **Validate** Observe adoption rates

[Munzner, 2014]
Design Activity Framework

**Understand**
- design requirements

**Ideate**
- ideas

**Make**
- prototypes

**Deploy**
- visualization system

[S. McKenna et al., 2014]
Understand

• Motivation: gather, observe, and research available information to find the needs of the user

• Design requirements can be broken into:
  - opportunities
  - constraints (limitations)
  - considerations (more flexible)

• Example (Cyber Security Dashboard):
  - read research papers to understand the field and different users
  - observed and interviewed many users
  - created personas to filter to a subset of users
  - identified high-level goal of communication of cyber information

[S. McKenna et al., 2014]
Ideate

- Motivation: to generate good ideas for supporting the understand artifacts
- Sketches often get externalized in various forms, up to mock-ups and wireframes
- Anyone can sketch! the goal is to capture an idea, not create a masterpiece or spend hours cleaning up the sketch
- Example (Cyber Security Dashboard):
  - sketched out various forms of the data
  - created data sketches (20 different ways to visualize the data)
  - evaluated these with an analyst
  - identified most clear encoding for all users

[S. McKenna et al., 2014]
Ideate

1. directed network of src-dest IP addresses
   internal = red, external = green
   1,000,000 flows - 100,000 IP’s - 400,000 edges
   zooming in on graph & selecting a node below

2. directed network of src-dest IP addresses
   same as before, but new layout
   zoomed-in, selecting a group of nodes

3. plotting IP addresses as points on a map
   sized by # packets, colored by # bytes

4. plotting IP address source & destination as lines on a map
   centered on Utah; over-plotted for entire dataset...

5.  

[S. McKenna et al., 2015]
Make

• Motivation: to concretize ideas into tangible prototypes
• Prototypes are "approximations of a product along some dimensions of interest" [B. Hartmann et al., 2006].
• Can be lower or high-fidelity prototypes, usually over time
• For visualization, often built using code and higher-fidelity
• Example (Cyber Security Dashboard):
  - built first prototype using a treemap of alerts
  - evaluated this treemap as an idea, leading to map view
  - constructed map-based dashboard
  - evaluated with users, anecdotally and in a usability study

[S. McKenna et al., 2014]
Make


[S. McKenna et al., 2015]
Make

System Usability Score by User

Score 100 80 80 85 90 80 80

Analysts A1 A2 A3 A4 A5 M1 M2 M3 M4

Managers

[S. McKenna et al., 2015]
Deploy

• Motivation: to bring a prototype into effective action in a real-world setting in order to support the target users’ work and goals
• More software engineering-related decisions
• Tool must be usable and fit into a user’s workflow
• May have to optimize algorithms to increase interactivity and speed
• Example (Cyber Security Dashboard):
  - showcased prototype to find its benefits
  - implemented some benefits in an existing toolkit
  - adoption of the simpler map-based view
  - utilization of multi-view, instant interactions

[S. McKenna et al., 2014]
Deploy


[D. Staheli et al., 2016]
Five Design-Sheet Methodology

Data Collection:  
- User Observation
- Interviews
- Workshops

Synthesis:  
- Sketching
- Storyboarding

Explore & Ideate:  
- Quick sketching
- User scenarios
- Improvisation

Prototype:  
- Refined Sketching
- Rendering
- Animated sequences
- Videos, model making

GoALS
- Document solutions
- Validate existing solutions
- Explore lots of solutions
- Communicate
- Demonstrate

GENEX
- COLLECT
- RELATE
- CREATE
- DONATE

[F. Roberts et al., 2016]
Five Stages

1. Meet with client and consider task; or contemplate task on own.
2. Ideate and sketch small ideas.
3. Sketch and plan three alternative designs.
4. Consider solutions with client; or deliberate on own.
5. Generate realization sheet, and implement prototype. Discuss with client and re-iterate if necessary.

[J. Roberts et al., 2016]
Five Stages

---

[J. Roberts et al., 2016]
The Five Sheets

Sheet 1
Ideas
Filter
Categorize
Combine & Refine
Question

Sheet 2,3,4
Layout
Information
Focus / Parti
Operations
Discussion

Sheet 5
Layout
Information
Operations
Focus / Parti
Detail

Ideation
Alternative Designs
Realization

[J. Roberts et al., 2016]
Example: University Access for Disabled Students

[Image]

[J. Roberts et al., 2016]
Sheets 2-4

[D. Koop, DSC 530, Spring 2017]

[J. Roberts et al., 2016]
Sheets 2-4

[J. Roberts et al., 2016]
Sheet 5

[Fig. 6: An example of the FdS are shown on the left, with a picture of the final prototype on the right. Created for the Information Visualization module as part of the MSc course. The student chose to investigate data regarding University access for disabled students.

[J. Roberts et al., 2016]
discovery of penicillin and it is these that are actually good ideas: e.g., sticky-note glue or the well written examples where scientists make errors or have accidents. Either try to fix these mistakes or use the result to your favor. There are interpretation.

- Metaphors work both ways: they both inspire and are needed for in-
- Use analogies. Metaphors help users to instantly understand the ideas and learn from others' work.
- Discover every idea and every solution so far. Look at other ideas from other subjects for inspiration, e.g., biomimicry.
- Ent thoughts new ideas can be formed. De Bono's 'green-hat' suggests his six thinking hats.
- Few values, small size, evenly spread, sparse or dense?
- Range (all at one time or over several years)?
- Temporal (annual, quaterly, etc.)?
- Hierarchical?
- They stored? What is the access to the data (API, JSON & distribution. What is the distribution of the data?
- Variables?
- Are the variables categorical (nominal or ordinal)?
- Persist:
- Filter
- Dissimilar ideas:
- Transference:
- De Bono encourages us to take on different personalities for different situations, think of impossible solutions.
- Oversight:
- It is difficult to quantify the number of mini-ideas required; but the more ideas that are generated, the more likely it is that one of them will be a solution. However, the user does each action in turn, and decides whether they have completed it to a satisfactory level before moving onto the next task. But the very act of generating ideas can facilitate the exploration of ideas.
**D3.js** is a JavaScript library for manipulating documents based on data. **D3** helps you bring data to life using HTML, SVG, and CSS. D3's emphasis on web standards gives you the full capabilities of modern browsers without tying yourself to a proprietary framework, combining powerful visualization components and a data-driven approach to DOM manipulation.

See more examples.
JavaScript Libraries

- Building Blocks: HTML, CSS, SVG, and JavaScript
- More Ideas:
  - JavaScript Libraries
    - `<script src="http://d3js.org/d3.v4.js" charset="utf-8"></script>`
  - Minification: smaller code, no functional change
    - `<script src="http://d3js.org/d3.v4.min.js" charset="utf-8"></script>`
    - Can make debugging more difficult
  - Content Delivery Networks
    - Faster delivery of Web content, also works for js
    - `https://cdnjs.cloudflare.com/ajax/libs/d3/4.5.0/d3.min.js`
JavaScript Reminders

- Functions are first-class objects in JavaScript
- Closures are functions that remember their environment
- Method Chaining: methods can also return the objects passed in or derivative objects to allow you to call another function on the result
  - You often end up following specific patterns where an object being manipulated requires multiple calls:
    - `rect.attr("width", 200).attr("height", 100);`
  - Or it is clear that the method returns a specific object that you wish to make changes to:
    - `svg.select("#myrect").style("fill", "blue");`
  - Of course, you may store the returned object as a variable and make each call separately
  - Coding style: Indent, often put each call on a new line
Data-Driven Documents (D3)

• [http://d3js.org/](http://d3js.org/)
• Original Authors: Mike Bostock, Vadim Ogievestky, and Jeff Heer
• Open Source
• 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 if you wish
• A top project on GitHub (over 60,000 stars as of 2/8/2017)
• Lots of impressive examples
  - Bostock was a New York Times Graphics Editor
  - [http://bost.ocks.org/mike/](http://bost.ocks.org/mike/)
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

• http://dakoop.github.io/IntroD3/
  - (Updated from original for D3 v4)

• Other references:
  - Murrany’s book on Interactive Data Visualization for the Web
  - The D3 website: d3js.org