CIS 467/602-01: Data Visualization

JavaScript and Design

Dr. David Koop
Assignment 1

• Any questions?

John Adam Smith

#00000000

Data Visualization (CIS 467/602-01)

Assignment #1

This assignment is all my own work. I did not copy the code from any other source.
Assignment 1

• Due to snow days and missed office hours, I’ll push due date to Monday @5pm.
• For those of you who want to enjoy a long weekend, finish the assignment by Friday :)
JavaScript in one slide

• Interpreted and Dynamically-typed Programming Language
• Statements end with semi-colons, normal blocking with brackets
• Variables: var a = 0;
• Operators: +, -, *, /, [ ]
• Control Statements: if, else
• Loops: for, while, do-while
• Arrays: var a = [1,2,3]; a[99] = 100; console.log(a.length);
• Functions: function myFunction(a,b) { return a + b; } 
• Objects: var obj; obj.x = 3; obj.y = 5;
  - Protoypes for instance functions
• Comments are /* Comment */ or // Single-line Comment
Manipulating the DOM with JavaScript

- Key global variables:
  - `window`: Global namespace
  - `document`: Current document
  - `document.getElementById(...)`: Get an element via its id

- HTML is parsed into an in-memory document (DOM)
- Can access and **modify** information stored in the DOM
- Can add information to the DOM
Manipulating SVG via JavaScript

• SVG can be navigated just like the DOM

• Example:

```javascript
function addEltToSVG(svg, name, attrs) {
  var element = document.createElementNS(
    "http://www.w3.org/2000/svg", name);
  if (attrs === undefined) attrs = {};
  for (var key in attrs) {
    element.setAttributeNS(null, key, attrs[key]);
  }
  svg.appendChild(element);
}

mysvg = document.getElementById("mysvg");
addEltToSVG(mysvg, "rect", {
  "x": 50, "y": 50,
  "width": 40,"height": 40,
  "fill": "blue"});
```
Recap (Data)

- Basic Data Types:
  - Items, Attributes, Links, Positions, Grids
- Dataset Types:
  - Tabular, Networks, Fields, Geometry, Sets
- Attribute Types:
  - Categorical and Ordered (Ordinal and Quantitative)
  - Ordered: Sequential, Diverging, and Cyclic
- Tables: Keys and Values
- Fields: Scalar, Vector, and Tensor
- Time-Varying Data
Tasks

• Why? Understand data, but what do I want to do with it?
• Levels: High (Produce/Consume), Mid (Search), Low (Queries)
• Another key concern: Who?
  - Designer <-> User (A spectrum)
  - Complex <-> Easy to Use
  - General <-> Context-Specific
  - Flexible <-> Constrained
  - Varied Data <-> Specific Data
Tasks

<table>
<thead>
<tr>
<th>Why?</th>
<th>Actions</th>
<th>Why?</th>
<th>Targets</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Analyze</td>
<td></td>
<td>All Data</td>
</tr>
<tr>
<td></td>
<td>→ Consume</td>
<td>→ Trends</td>
<td>Outliers</td>
</tr>
<tr>
<td></td>
<td>→ Discover</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>→ Present</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>→ Enjoy</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>→ Produce</td>
<td>→ Attributes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>→ Annotate</td>
<td>→ One</td>
<td></td>
</tr>
<tr>
<td></td>
<td>→ Record</td>
<td>→ Many</td>
<td></td>
</tr>
<tr>
<td></td>
<td>→ Derive</td>
<td>→ Distribution</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>→ Dependency</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>→ Correlation</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>→ Similarity</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Search</td>
<td></td>
<td>Network Data</td>
</tr>
<tr>
<td></td>
<td></td>
<td>→ Topology</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>→ Paths</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Query</td>
<td></td>
<td>Spatial Data</td>
</tr>
<tr>
<td></td>
<td></td>
<td>→ Shape</td>
<td></td>
</tr>
</tbody>
</table>

[Munzner (ill. Maguire), 2014]
Actions: Analyze

→ Analyze

→ Consume
  → Discover

→ Present

→ Enjoy

→ Produce
  → Annotate

→ Record

→ Derive

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

- Discover new knowledge
  - Generate new hypothesis or verify existing one
  - Designer doesn’t know what users need to see
- Present known information
  - Presenter already knows what the data says
  - Wants to communicate this to an audience
- Enjoy
  - Similar to discover, but without concrete goals
Name Voyager

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 babies

[Graph showing name trends over time for names starting with 'AN'.]

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)
MTA Fare Data Exploration
## Actions: Search

**Search**

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

- **ALL DATA**
  - Trends
  - Outliers
  - Features

- **ATTRIBUTES**
  - One: Distribution, Extremes
  - Many: Dependency, Correlation, Similarity

- **NETWORK DATA**
  - Topology, Paths

- **SPATIAL DATA**
  - Shape

[Munzner (ill. Maguire), 2014]
Analysis Example: Different “Idioms”

[SpaceTree, Grosjean et al.]  [TreeJuxtaposer, Munzner et al.]
“Idiom” Comparison

SpaceTree

TreeJuxtaposer


Munzner (ill. Maguire), 2014
“Idiom” Comparison

SpaceTree

TreeJuxtaposer


What?

Actions

Present → Locate → Identify

Targets

Path between two nodes

Why?

How?

Encode → Navigate → Select → Filter → Aggregate

Encode → Navigate → Select → Arrange

[Munzner (ill. Maguire), 2014]
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
  - In Quantitative attribute on nodes
  - Out Filtered Tree

- **Why**?
  - Summarize
  - Topology

- **How**?
  - Reduce
  - Filter

[Munzner (ill. Maguire), 2014]
Design Iteration

http://chartsnthings.tumblr.com/post/62679766588/19-sketches-of-quarterback-timelines
### 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</td>
</tr>
<tr>
<td>New Orleans Saints</td>
<td>Aaron Brooks, Drew Brees</td>
</tr>
<tr>
<td>Atlanta Falcons</td>
<td>Michael Vick, 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</td>
</tr>
<tr>
<td>Carolina Panthers</td>
<td>Jake Delhomme, Cam Newton</td>
</tr>
<tr>
<td>Denver Broncos</td>
<td>Jake Plummer, Kyle Orton, Tim Tebow, Peyton Manning</td>
</tr>
<tr>
<td>Arizona Cardinals</td>
<td>Matt Leinart, Kurt Warner</td>
</tr>
<tr>
<td>Jacksonville Jaguars</td>
<td>Byron Leftwich, David Garrard, David Garrard, Blaine Gabbert</td>
</tr>
<tr>
<td>Detroit Lions</td>
<td>Joey Harrington, Jon Kitna, Matthew Stafford</td>
</tr>
<tr>
<td>Tampa Bay Buccaneers</td>
<td>Chris Simms, Bruce Grac, Jeff Garcia, Josh Freeman, Josh Freeman</td>
</tr>
<tr>
<td>Dallas Cowboys</td>
<td>Drew Bledsoe, Tony Romo, Tony Romo, Tony Romo</td>
</tr>
</tbody>
</table>

Design Iteration

Each streak shows consecutive starts by a quarterback for a single team. Streaks include playoffs.

Only two players have longer streaks: Brett Favre (275) and Eli's brother, Peyton (227).

Among active players, Philip Rivers (122) and Joe Flacco (96) are closest behind Eli.

Find a quarterback

Eli Manning (149)

http://chartsnthings.tumblr.com/post/62679766588/19-sketches-of-quarterback-timelines
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
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
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
Examples

- Read about the validation examples in the book