Data Visualization (CIS/DSC 468)

Web Programming

Dr. David Koop
Definition of Visualization

“Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively.”
— T. Munzner
Why Visual?

[F. J. Anscombe]
What are the limitations of visualization?
Design Iteration

Design Iteration

| New York Giants | Eli Manning |
| Indianapolis Colts | Peyton Manning, Andrew Luck |
| San Diego Chargers | Drew Brees, Philip Rivers, Tony Brady |
| Baltimore Ravens | Kyle Boller, Joe Flacco |
| New England Patriots | Tom Brady, Matt Cassel, Aaron Rodgers |
| Green Bay Packers | Brett Favre, Aaron Rodgers |
| New Orleans Saints | Drew Brees, Drew Brees |
| Atlanta Falcons | Michael Vick, Matt Ryan |
| New York Jets | Chad Pennington, Brett Favre |
| Cincinnati Bengals | Carson Palmer, Ryan Fitz, Carson Palmer |
| Houston Texans | David Carr, Matt Schaub |
| Carolina Panthers | Jake Delhomme, Cam Newton |
| Denver Broncos | Jake Plummer, Jon Kitna |
| Arizona Cardinals | Matt Leinart, Kurt Warner |
| Jacksonville Jaguars | Byron Leftwich, David Garrard, Blaine Gabbert |
| Detroit Lions | Joey Harrington, Matthew Stafford |
| Tampa Bay Buccaneers | Chris Simms, Bruce Gradkowski, Jeff Garcia, Josh Freeman |
| Dallas Cowboys | Tony Romo |

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)

Effectiveness

Average Annual Global Temperature in Fahrenheit
1880-2015

[S. Hayward, 2015]
Effectiveness

Average Annual Global Temperature in Fahrenheit
1880-2015

[S. Hayward, 2015]
Administrivia

• **Course Web Site**
  - Please review!
• **Textbook**
  - Munzner (required)
  - Murray (available online)
• **Assignments**
• **Exams (2 + Final, 3 Total)**
• **Registration**
  - Add/Drop Deadline **Today**
How do we create modern visualizations?
Tools

• Desktop Applications:
  - Excel (see excelcharts.com)
  - Tableau
  - ...

• Programming Frameworks
  - Processing
  - OpenFrameworks
  - d3.js
  - ...

• Advantages and disadvantages
  - Speed, customization, control, data, audience
**Data-Driven Documents**

**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.
Web Programming Tools

- Basic: Text editor and Modern Browser
- Developer Tools: Built in to browsers
- Web Environments: JSFiddle, Liveweave, CodePen, etc.
- IDEs: WebStorm, etc.
What languages do we use on the Web?
Languages of the Web

- HTML
- CSS
- SVG
- JavaScript
  - Versions of Javascript: ES6, ES2015, ES2017…
  - Specific frameworks: react, jQuery, bootstrap,
What is HTML?
Hyper Text Markup Language (HTML)

• Markup languages allow users to encode the **semantics** of text
• Tags define the boundaries of the structures of the content
  - Tags are enclosed in angle brackets (e.g. `<html>`)  
  - Most of the time, you have a start and end tag  
  - End tags are just like start tags except that they have forward slash after the open bracket (e.g. `</html>`)  
  - Tags may be nested but not mismatched  
    • `<p>A <strong><em>very</em></strong> cool example</p>`  
    • `<p>A <strong>very <em>cool</em></strong> example</p>`  
  - What about `<img src="mypicture.png" alt="My Image">`?
HTML Elements and Attributes

• Tags denote **elements** of the content (e.g. sections, paragraphs, images)

• Each element may have **attributes** which define other information about the element
  
  - An attribute has a **key** and **value** (*key=“value”*)
  
  - e.g. `<img src=“mypicture.png” alt=“My Image”>`

• Many different elements available
  
  - Common: headers (h1, ..., h6), paragraph (p), lists (ul, ol, li), emphasis (em, strong), link (a), spans & divisions (span, div)
  
  - Lots more (e.g. abbr): see [MDN Documentation](#)

• Many different attributes available
  
  - See [MDN Documentation](#): note that some are legacy due to CSS
HTML Element Structure & Naming

- Elements structure a document
  - Document Object Model (DOM)
  - We can visualize this information
  - More importantly, we can **navigate** this tree

- Identifying and Classifying elements: `id` and `class` attributes
  - `id` identifies a **single** element—use for a unique case
  - `class` may identify **multiple** elements—use for common cases
  - Each element may have multiple classes, separate by spaces
  - Use normal identifiers: don’t start the name with a number
Other HTML Trivia

• `<`, `>`, `,`, and `"` are special characters, escape with `&lt;`, `&gt;`, `&`, and `&quot;` (note the semi-colon)

• Comments are enclosed by `<!--` and `-->`
  - `<!-- This is a comment -->`

• HTML Documents begin with a DOCTYPE declaration
  - For HTML5, this is easier `<!DOCTYPE html>`

• meta tag: `<meta charset="UTF-8"/>`

• HTML has audio and video tags, math equation support, and more
Basic HTML File

<!DOCTYPE html>
<html>
    <head>
        <title>A Basic Web Page</title>
    </head>
    <body>
        <h1>My Wicked Awesome Web Page</h1>
        <p><em>This is <strong>cool</strong>. What about <u><strong>this?</strong></u></em></p>
        <img src="patriots.jpg"/>
    </body>
</html>
What is CSS?
Cascading Style Sheets (CSS)

• Separate from content, just specifies how to style the content
• Style information can appear in three places:
  - External file
  - In a style element at the beginning of the HTML file
  - In a specific element in the body of a document (least preferable)
• Why Cascading?
  - Don’t want to have to specify everything over and over
  - Often want to use the same characteristics in a region of the DOM
  - Use inheritance: properties that apply to children cascade down
CSS Selectors

• How do we specify what part(s) of the page we want to style?
• The element types themselves (the HTML tag)
  - strong { color: red; }
• Classes of elements (ties to HTML class attribute)
  - .cool { color: blue; }
• A specific element (ties to HTML id attribute)
  - #main-section { color: green; }
• Relationships
  - Descendant: p em { color: yellow; }
  - Child: p > em { color: orange; }
• Pseudo-classes: a:hover { color: purple; }

D. Koop, CIS 468, Spring 2017
Other CSS Bits

• Comments: /* This is a comment in CSS */

• Grouping Selectors: p, li { font-size: 12pt; }

• Multiple Classes: .cool.temp { color: blue; }

• Colors:
  - Names (Level 1, 2, & 3): red, orange, antiquewhite
  - Dash notation (3- & 6-character): #fff, #00ff00
  - Integer or % RGB and HSL Functions: rgb( 255, 0, 0), rgb(50%, 50%, 0%), hsl(120, 100% 50%)
    - Also background-color

• Watch out for multiple rules (look at how a web browser parses)

• Again, much more documentation at MDN
Sample CSS

body {
  font-face: sans-serif;
  font-size: 12pt;
}

em { color: green; }

em u { color: red; }

em > strong { color: blue; }

img { border: 4px solid red; }

• What colors are displayed for this HTML (with the above stylesheet)?
  - <em>This is <strong>cool</strong>. What about <u><strong>this?</strong></u></em>
How to add CSS to HTML

• External: a separate file via a link element (in the <head> section):
  - `<link rel="stylesheet" href="styles.css">`

• Embedded: in the header:
  - `<style type="text/css"> ... </style>`

• Inline: for a specific element: (Discouraged!)
  - `<p style="font-weight: bold;">Some text</p>`