Data Visualization (DSC 530/CIS 602-02)

JavaScript and Data

Dr. David Koop
Scalable Vector Graphics (SVG)

- Vector graphics vs. Raster graphics
- Drawing commands versus a grid of pixels
- Why vector graphics?
SVG Example

- [Codepen](http://codepen.io/dakoop/pen/yexVXb)
- `<svg id="mysvg" width="300" height="600">
   <circle cx="50" cy="50" r="50"/>
   <rect class="lego" x="150" y="150" width="50" height="20"/>
   <path id="triangle" d="M 20 200 L 120 200 L 120 250 Z"/>
 </svg>
- `circle { fill: green; stroke: black; stroke-width: 4px; }`
- `.lego { fill: red; stroke:`
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 (<expr>) {...} else {...}`, `switch`
- 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;`  
  - Prototypes for instance functions
- Comments are `/* Comment */` or `// Single-line Comment`
Important JavaScript Concepts

• Functional Programming: you can pass functions to functions
  - Array map/filter/reduce/forEach
  - cTemps=fTemps.map(function(d){return (d-32)*5.0/9.0;});

• Closures: functions keep track of their environments
  - function makeAdder(x) {
    return function(y) { return x + y; }; }

• Objects and Properties:
  - var student = {name: "John Smith", id: "000012345", class: "Senior", hometown: "Fall River, MA, USA"};
  - Properties can be accessed via dot or bracket notation
  - Objects can also function as associative arrays

• Function chaining: succinct, avoids intermediate variables in code
Quiz

• Given this data:
  - `var a = [6, 2, 6, 10, 7, 18, 0, 17, 20, 6];`

• Questions:
  - How would I subtract one from each item?
  - How would I find only the values >= 10?
  - How would I sum the array?
  - How would I create a reversed version of the array?
Quiz Answers

• Data: \( \text{var } a = [6, 2, 6, 10, 7, 18, 0, 17, 20, 6]; \)

• How would I subtract one from each item?
  - \( a.\text{map}(\text{function}(d) \{ \text{return } d-1; \}) \)

• How would I find only the values >= 10?
  - \( a.\text{filter}(\text{function}(d) \{ \text{return } d >= 10; \}) \)

• How would I sum the array?
  - \( a.\text{reduce}(\text{function}(s,d) \{ \text{return } s + d; \}) \)

• How would I create a reversed version of the array?
  - \( b = []; \)
    - \( a.\text{forEach}(\text{function}(d) \{ b.\text{unshift}(d); \}); \)
    - \( \ldots \text{or } a.\text{reverse}() \) // modifies in place
Assignment 1

- Use HTML, CSS, SVG, and JavaScript
- Part 3 will take longer
- Due next Friday (Feb. 10)
- Start soon
Manipulating the DOM with JavaScript

• Key global variables:
  • `window`: Global namespace
  • `document`: Current document

• Methods to get specific elements:
  • `document.querySelector(...)` : Get an element via a selector
  • `document.getElementById(...)` : Get an element via its id
  • `document.querySelectorAll(...)` : Get a list of all matching elts

• HTML is parsed into an in-memory document (DOM)
• Can access and **modify** information stored in the DOM
• Can add information to the DOM
Example: JavaScript and the DOM

• Start with no real content, just divs:
  `<div id="firstSection"></div>`
  `<div id="secondSection"></div>`
  `<div id="finalSection"></div>`

• Get existing elements:
  - `document.querySelector`
  - `document.getElementById`

• Programmatically add elements:
  - `document.createElement`
  - `document.createTextNode`
  - `Element.appendChild`
  - `Element.setAttribute`
Creating SVG figures via JavaScript

• SVG elements can be accessed and modified just like HTML elements

• Create a new SVG programmatically and add it into a page:

  - var divElt = document.getElementById("chart");
    var svg = document.createElementNS("http://www.w3.org/2000/svg", "svg");
    divElt.appendChild(svg);

• You can assign attributes:

  - svg.setAttribute("height", 400);
    svg.setAttribute("width", 600);
    svgCircle.setAttribute("r", 50);
SVG Manipulation Example

• Draw a horizontal bar chart
  - var a = [6, 2, 6, 10, 7, 18, 0, 17, 20, 6];

• Steps?
SVG Manipulation Example

• Draw a horizontal bar chart
  - \texttt{var a = \[6, 2, 6, 10, 7, 18, 0, 17, 20, 6\];}

• Steps:
  - Programmatically create SVG
  - Create individual rectangle for each item
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.setAttribute(key, attrs[key]);
    }
    svg.appendChild(element);
  }

  mysvg = document.getElementById("mysvg");
  addEltToSVG(mysvg, "rect", {
    "x": 50, "y": 50,
    "width": 40,"height": 40,
    "fill": "blue"});
  ```
“Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively.”
Data

- What is this data?

<table>
<thead>
<tr>
<th></th>
<th>42ND STREET &amp; 8TH AVENUE</th>
<th>00228985</th>
<th>00008471</th>
<th>00000441</th>
<th>00001455</th>
<th>00000134</th>
<th>00033341</th>
<th>00071255</th>
</tr>
</thead>
<tbody>
<tr>
<td>R011</td>
<td>14TH STREET-UNION SQUARE</td>
<td>00224603</td>
<td>00011051</td>
<td>0000827</td>
<td>00003026</td>
<td>00000660</td>
<td>00089367</td>
<td>00199841</td>
</tr>
<tr>
<td>R046</td>
<td>42ND STREET &amp; GRAND CENTRAL</td>
<td>00207758</td>
<td>00007908</td>
<td>0000323</td>
<td>00001183</td>
<td>00003001</td>
<td>00040759</td>
<td>00096613</td>
</tr>
</tbody>
</table>

- **Semantics**: real-world meaning of the data
- **Type**: structural or mathematical interpretation
- Both often require **metadata**
  - Sometimes we can infer some of this information
  - Line between data and metadata isn’t always clear
### Data

<table>
<thead>
<tr>
<th>REMOTE</th>
<th>STATION</th>
<th>FF</th>
<th>SEN/DIS</th>
<th>7-D AFAS UNL</th>
<th>D AFAS/RMF</th>
<th>JOINT RR TKT</th>
<th>7-D UNL</th>
<th>30-D UNL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>R011 42ND STREET &amp; 8TH AVENUE</td>
<td>00228985</td>
<td>00008471</td>
<td>00000441</td>
<td>00001455</td>
<td>00000134</td>
<td>00033341</td>
<td>00071255</td>
</tr>
<tr>
<td>2</td>
<td>R170 14TH STREET-UNION SQUARE</td>
<td>00224603</td>
<td>0011051</td>
<td>00000827</td>
<td>00003026</td>
<td>00000660</td>
<td>00089367</td>
<td>00199841</td>
</tr>
<tr>
<td>3</td>
<td>R046 42ND STREET &amp; GRAND CENTRAL</td>
<td>00207758</td>
<td>0007908</td>
<td>00000323</td>
<td>00001183</td>
<td>00003001</td>
<td>00040759</td>
<td>00096613</td>
</tr>
<tr>
<td>4</td>
<td>R012 34TH STREET &amp; 8TH AVENUE</td>
<td>00188311</td>
<td>0006490</td>
<td>00000498</td>
<td>00001279</td>
<td>00003622</td>
<td>00035527</td>
<td>00067483</td>
</tr>
<tr>
<td>5</td>
<td>R293 34TH STREET - PENN STATION</td>
<td>00168768</td>
<td>0006155</td>
<td>00000523</td>
<td>00001065</td>
<td>00005031</td>
<td>00030645</td>
<td>00054376</td>
</tr>
<tr>
<td>6</td>
<td>R033 42ND STREET/TIMES SQUARE</td>
<td>00159382</td>
<td>0005945</td>
<td>00000378</td>
<td>00001205</td>
<td>00000690</td>
<td>00058931</td>
<td>00078644</td>
</tr>
<tr>
<td>7</td>
<td>R022 34TH STREET &amp; 6TH AVENUE</td>
<td>00156008</td>
<td>0006276</td>
<td>00000487</td>
<td>00001543</td>
<td>00000712</td>
<td>00058910</td>
<td>00110466</td>
</tr>
<tr>
<td>8</td>
<td>R084 59TH STREET/COLUMBUS CIRCLE</td>
<td>00155262</td>
<td>0009484</td>
<td>00000589</td>
<td>00002071</td>
<td>00000542</td>
<td>00053397</td>
<td>00113966</td>
</tr>
<tr>
<td>9</td>
<td>R020 47-50 STREETS/ROCKEFELLER</td>
<td>00143500</td>
<td>0006402</td>
<td>00000384</td>
<td>00001159</td>
<td>00000723</td>
<td>00037978</td>
<td>00090745</td>
</tr>
<tr>
<td>10</td>
<td>R179 86TH STREET-LEXINGTON AVE</td>
<td>00142169</td>
<td>0010367</td>
<td>00000470</td>
<td>00001839</td>
<td>00000271</td>
<td>00050328</td>
<td>00125250</td>
</tr>
<tr>
<td>11</td>
<td>R023 34TH STREET &amp; 6TH AVENUE</td>
<td>00134052</td>
<td>0005005</td>
<td>00000348</td>
<td>00001112</td>
<td>00000649</td>
<td>00031531</td>
<td>00075040</td>
</tr>
<tr>
<td>12</td>
<td>R029 PARK PLACE</td>
<td>00121614</td>
<td>0004311</td>
<td>00000287</td>
<td>00000931</td>
<td>00000792</td>
<td>00025404</td>
<td>00065362</td>
</tr>
<tr>
<td>13</td>
<td>R047 42ND STREET &amp; GRAND CENTRAL</td>
<td>00100742</td>
<td>0004273</td>
<td>00000185</td>
<td>00000704</td>
<td>00001241</td>
<td>00022808</td>
<td>00068216</td>
</tr>
<tr>
<td>14</td>
<td>R031 34TH STREET &amp; 7TH AVENUE</td>
<td>00095076</td>
<td>0003990</td>
<td>00000232</td>
<td>00000727</td>
<td>00001459</td>
<td>00024284</td>
<td>00038671</td>
</tr>
<tr>
<td>15</td>
<td>R017 LEXINGTON AVENUE</td>
<td>00094655</td>
<td>0004688</td>
<td>00000190</td>
<td>00000833</td>
<td>00000754</td>
<td>00020018</td>
<td>00055066</td>
</tr>
<tr>
<td>16</td>
<td>R175 8TH AVENUE-14TH STREET</td>
<td>00094313</td>
<td>0003907</td>
<td>00000286</td>
<td>00001144</td>
<td>00000256</td>
<td>00038272</td>
<td>00074661</td>
</tr>
<tr>
<td>17</td>
<td>R057 BARCLAYS CENTER</td>
<td>00093804</td>
<td>0004204</td>
<td>00000454</td>
<td>00001386</td>
<td>00001491</td>
<td>00039113</td>
<td>00068119</td>
</tr>
<tr>
<td>18</td>
<td>R138 WEST 4TH ST-WASHINGTON SO</td>
<td>00093562</td>
<td>0004677</td>
<td>00000251</td>
<td>00000965</td>
<td>00000127</td>
<td>00031628</td>
<td>00074458</td>
</tr>
</tbody>
</table>
Data Types

• Items
  - An item is an individual discrete entity
  - e.g. row in a table, node in a network

• Attributes
  - An attribute is some specific property that can be measured, observed, or logged
  - a.k.a. variable, (data) dimension
## Items & Attributes

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>S</th>
<th>T</th>
<th>U</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Order ID</td>
<td>Order Date</td>
<td>Order Priority</td>
<td>Product Container</td>
<td>Product Base Margin</td>
</tr>
<tr>
<td>3</td>
<td>10/14/06</td>
<td>5-Low</td>
<td>Large Box</td>
<td>0.8</td>
<td>10/21/06</td>
</tr>
<tr>
<td>6</td>
<td>2/21/08</td>
<td>4-Not Specified</td>
<td>Small Pack</td>
<td>0.55</td>
<td>2/22/08</td>
</tr>
<tr>
<td>32</td>
<td>7/16/07</td>
<td>2-High</td>
<td>Small Pack</td>
<td>0.79</td>
<td>7/17/07</td>
</tr>
<tr>
<td>32</td>
<td>7/16/07</td>
<td>2-High</td>
<td>Jumbo Box</td>
<td>0.62</td>
<td>7/17/07</td>
</tr>
<tr>
<td>32</td>
<td>7/16/07</td>
<td>2-High</td>
<td>Medium Box</td>
<td>0.6</td>
<td>7/18/07</td>
</tr>
<tr>
<td>32</td>
<td>7/16/07</td>
<td>2-High</td>
<td>Medium Box</td>
<td>0.63</td>
<td>7/18/07</td>
</tr>
<tr>
<td>35</td>
<td>10/23/07</td>
<td>4-Not Specified</td>
<td>Wrap Bag</td>
<td>0.52</td>
<td>10/24/07</td>
</tr>
<tr>
<td>35</td>
<td>10/23/07</td>
<td>4-Not Specified</td>
<td>Small Box</td>
<td>0.58</td>
<td>10/25/07</td>
</tr>
<tr>
<td>36</td>
<td>11/3/07</td>
<td>1-Urgent</td>
<td>Small Box</td>
<td>0.55</td>
<td>11/3/07</td>
</tr>
<tr>
<td>65</td>
<td>3/18/07</td>
<td>1-Urgent</td>
<td>Small Pack</td>
<td>0.49</td>
<td>3/19/07</td>
</tr>
<tr>
<td>66</td>
<td>1/20/05</td>
<td>5-Low</td>
<td>Wrap Bag</td>
<td>0.56</td>
<td>1/20/05</td>
</tr>
<tr>
<td>69</td>
<td>12/18/06</td>
<td>5-Low</td>
<td>Small Pack</td>
<td>0.44</td>
<td>6/6/05</td>
</tr>
<tr>
<td>69</td>
<td>12/18/06</td>
<td>5-Low</td>
<td>Wrap Bag</td>
<td>0.6</td>
<td>6/6/05</td>
</tr>
<tr>
<td>70</td>
<td>12/18/06</td>
<td>5-Low</td>
<td>Small Box</td>
<td>0.59</td>
<td>12/23/06</td>
</tr>
<tr>
<td>70</td>
<td>12/18/06</td>
<td>5-Low</td>
<td>Wrap Bag</td>
<td>0.82</td>
<td>12/23/06</td>
</tr>
<tr>
<td>96</td>
<td>4/17/05</td>
<td>2-High</td>
<td>Small Box</td>
<td>0.55</td>
<td>4/19/05</td>
</tr>
<tr>
<td>97</td>
<td>1/29/06</td>
<td>3-Medium</td>
<td>Small Box</td>
<td>0.38</td>
<td>1/30/06</td>
</tr>
<tr>
<td>129</td>
<td>11/19/08</td>
<td>5-Low</td>
<td>Small Box</td>
<td>0.37</td>
<td>11/28/08</td>
</tr>
<tr>
<td>130</td>
<td>5/8/08</td>
<td>2-High</td>
<td>Small Box</td>
<td>0.37</td>
<td>5/9/08</td>
</tr>
<tr>
<td>130</td>
<td>5/8/08</td>
<td>2-High</td>
<td>Medium Box</td>
<td>0.38</td>
<td>5/10/08</td>
</tr>
<tr>
<td>130</td>
<td>5/8/08</td>
<td>2-High</td>
<td>Small Box</td>
<td>0.6</td>
<td>5/11/08</td>
</tr>
<tr>
<td>132</td>
<td>6/11/06</td>
<td>3-Medium</td>
<td>Medium Box</td>
<td>0.6</td>
<td>6/12/06</td>
</tr>
<tr>
<td>132</td>
<td>6/11/06</td>
<td>3-Medium</td>
<td>Jumbo Box</td>
<td>0.69</td>
<td>6/14/06</td>
</tr>
<tr>
<td>134</td>
<td>5/1/08</td>
<td>4-Not Specified</td>
<td>Large Box</td>
<td>0.82</td>
<td>5/3/08</td>
</tr>
<tr>
<td>135</td>
<td>10/21/07</td>
<td>4-Not Specified</td>
<td>Small Pack</td>
<td>0.64</td>
<td>10/23/07</td>
</tr>
<tr>
<td>166</td>
<td>9/12/07</td>
<td>2-High</td>
<td>Small Box</td>
<td>0.55</td>
<td>9/14/07</td>
</tr>
<tr>
<td>193</td>
<td>8/8/06</td>
<td>1-Urgent</td>
<td>Medium Box</td>
<td>0.57</td>
<td>8/10/06</td>
</tr>
<tr>
<td>194</td>
<td>4/5/08</td>
<td>3-Medium</td>
<td>Wrap Bag</td>
<td>0.42</td>
<td>4/7/08</td>
</tr>
</tbody>
</table>

**Note:** The table shows the order details including items and attributes.
Data Types

• **Nodes**
  - Synonym for item but in the context of networks (graphs)

• **Links**
  - A **link** is a relation between two items
  - e.g. social network friends, computer network links
Items & Links

[Source: Bostock, 2011]
Data Types

• Positions:
  - A **position** is a location in space (usually 2D or 3D)
  - May be subject to projections
  - e.g. cities on a map, a sampled region in an CT scan

• Grids:
  - A **grid** specifies how data is sampled both geometrically and topologically
  - e.g. how CT scan data is stored
Positions and Grids
Dataset Types

- **Tables**
  - Attributes (columns)
  - Items (rows)
  - Cell containing value

- **Networks**
  - Link
  - Node (item)

- **Fields (Continuous)**
  - Grid of positions
  - Cell
  - Attributes (columns)
  - Value in cell

- **Geometry (Spatial)**
  - Position

- **Multidimensional Table**
  - Key 1
  - Key 2
  - Attributes
  - Value in cell

- **Trees**

[Munzner (ill. Maguire), 2014]
# Tables

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Order ID</td>
<td>Order Date</td>
<td>Order Priority</td>
<td></td>
<td></td>
<td>Product Container</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>10/14/06</td>
<td>5-Low</td>
<td>Large Box</td>
<td>0.8</td>
<td>10/21/06</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>2/21/08</td>
<td>4-Not Specified</td>
<td>Small Pack</td>
<td>0.55</td>
<td>2/22/08</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>7/16/07</td>
<td>2-High</td>
<td>Small Pack</td>
<td>0.79</td>
<td>7/17/07</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>7/16/07</td>
<td>2-High</td>
<td>Jumbo Box</td>
<td>7/17/07</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>7/16/07</td>
<td>2-High</td>
<td>Medium Box</td>
<td>7/18/07</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>7/16/07</td>
<td>2-High</td>
<td>Medium Box</td>
<td>7/18/07</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>10/23/07</td>
<td>4-Not Specified</td>
<td>Wrap Bag</td>
<td>0.52</td>
<td>10/24/07</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>10/23/07</td>
<td>4-Not Specified</td>
<td>Small Box</td>
<td>0.58</td>
<td>10/25/07</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>11/3/07</td>
<td>1-Urgent</td>
<td>Small Box</td>
<td>0.55</td>
<td>11/3/07</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>65</td>
<td>3/18/07</td>
<td>1-Urgent</td>
<td>Small Pack</td>
<td>0.49</td>
<td>3/19/07</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>66</td>
<td>1/20/05</td>
<td>5-Low</td>
<td>Wrap Bag</td>
<td>0.56</td>
<td>1/20/05</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>69</td>
<td>5-Not Specified</td>
<td>Small Pack</td>
<td>0.44</td>
<td>6/6/05</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>69</td>
<td>5-Not Specified</td>
<td>Wrap Bag</td>
<td>0.6</td>
<td>6/6/05</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>12/18/06</td>
<td>5-Low</td>
<td>Small Box</td>
<td>0.59</td>
<td>12/23/06</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>12/18/06</td>
<td>5-Low</td>
<td>Wrap Bag</td>
<td>0.82</td>
<td>12/23/06</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>96</td>
<td>4/17/05</td>
<td>2-High</td>
<td>Small Box</td>
<td>0.55</td>
<td>4/19/05</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>97</td>
<td>1/29/06</td>
<td>3-Medium</td>
<td>Small Box</td>
<td>0.38</td>
<td>1/30/06</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>129</td>
<td>11/19/08</td>
<td>5-Low</td>
<td>Small Box</td>
<td>0.37</td>
<td>11/28/08</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>130</td>
<td>5/8/08</td>
<td>2-High</td>
<td>Small Box</td>
<td>0.37</td>
<td>5/9/08</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>130</td>
<td>5/8/08</td>
<td>2-High</td>
<td>Medium Box</td>
<td>0.38</td>
<td>5/10/08</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>130</td>
<td>5/8/08</td>
<td>2-High</td>
<td>Small Box</td>
<td>0.6</td>
<td>5/11/08</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>132</td>
<td>6/11/06</td>
<td>3-Medium</td>
<td>Medium Box</td>
<td>0.6</td>
<td>6/12/06</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>132</td>
<td>6/11/06</td>
<td>3-Medium</td>
<td>Jumbo Box</td>
<td>0.69</td>
<td>6/14/06</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>134</td>
<td>5/1/08</td>
<td>4-Not Specified</td>
<td>Large Box</td>
<td>0.82</td>
<td>5/3/08</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>135</td>
<td>10/21/07</td>
<td>4-Not Specified</td>
<td>Small Pack</td>
<td>0.64</td>
<td>10/23/07</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>166</td>
<td>9/12/07</td>
<td>2-High</td>
<td>Small Box</td>
<td>0.55</td>
<td>9/14/07</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>193</td>
<td>8/8/06</td>
<td>1-Urgent</td>
<td>Medium Box</td>
<td>0.57</td>
<td>8/10/06</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>194</td>
<td>4/5/08</td>
<td>3-Medium</td>
<td>Wrap Bag</td>
<td>0.42</td>
<td>4/7/08</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table Visualizations

[D. Koop, DSC 530, Spring 2017] [M. Bostock, 2011]
Networks

• Why networks instead of graphs?
• Tables can represent networks
  - Many-many relationships
  - Also can be stored as specific graph databases or files
Networks

Figure 7: US airlines graph (235 nodes, 2101 edges) (a) not bundled and bundled using (b) FDEB with inverse-linear model, (c) GBEB, and (d) FDEB with inverse-quadratic model.

Figure 8: US migration graph (1715 nodes, 9780 edges) (a) not bundled and bundled using (b) FDEB with inverse-linear model, (c) GBEB, and (d) FDEB with inverse-quadratic model. The same migration flow is highlighted in each graph.

Figure 9: A low amount of straightening provides an indication of the number of edges comprising a bundle by widening the bundle. (a) $s=0$, (b) $s=10$, and (c) $s=40$. If $s$ is $0$, color more clearly indicates the number of edges comprising a bundle.

We generated use the rendering technique described in Section 4.1. To facilitate the comparison of migration flow in Figure 8, we use a similar rendering technique as the one that Cui et al. [CZQ08] used to generate Figure 8c.

The airlines graph is comprised of 235 nodes and 2101 edges. It took 19 seconds to calculate the bundled airlines graphs (Figures 7b and 7d) using the calculation scheme presented in Section 3.3. The migration graph is comprised of 1715 nodes and 9780 edges. It took 80 seconds to calculate the bundled migration graphs (Figures 8b and 8d) using the same calculation scheme. All measurements were performed on an Intel Core 2 Duo 2.66GHz PC running Windows XP with 2GB of RAM and a GeForce 8800GT graphics card.

Our prototype was implemented in Borland Delphi 7.

[Holten & van Wijk, 2009]
Networks

Figure 7: US airlines graph (235 nodes, 2101 edges) (a) not bundled and bundled using (b) FDEB with inverse-linear model, (c) GBEB, and (d) FDEB with inverse-quadratic model.

Figure 8: US migration graph (1715 nodes, 9780 edges) (a) not bundled and bundled using (b) FDEB with inverse-linear model, (c) GBEB, and (d) FDEB with inverse-quadratic model. The same migration flow is highlighted in each graph.

Figure 9: A low amount of straightening provides an indication of the number of edges comprising a bundle by widening the bundle. (a) s = 0, (b) s = 10, and (c) s = 40. If s is 0, color more clearly indicates the number of edges comprising a bundle.

We generated the bundled graphs using the force-directed edge bundling algorithm described in Section 4.1. To facilitate the comparison of migration flow in Figure 8, we used a similar rendering technique as the one Cui et al. [CZQ-08] used to generate Figure 8c.

The airlines graph is comprised of 235 nodes and 2101 edges. It took 19 seconds to calculate the bundled airlines graphs (Figures 7b and 7d) using the calculation scheme presented in Section 3.3. The migration graph is comprised of 1715 nodes and 9780 edges. It took 80 seconds to calculate the bundled migration graphs (Figures 8b and 8d) using the same calculation scheme. All measurements were performed on an Intel Core 2 Duo 2.66GHz PC running Windows XP with 2GB of RAM and a GeForce 8800GT graphics card.

Our prototype was implemented in Borland Delphi 7.

[Holten & van Wijk, 2009]
Fields

Scalar Fields  Vector Fields  Tensor Fields

Each point in space has an associated...
Fields

Scalar Fields
(Order-0 Tensor Fields)

Vector Fields
(Order-1 Tensor Fields)

Tensor Fields
(Order-2+)

Each point in space has an associated...

Scalar

$\mathbf{s}_0$

Vector

$\begin{bmatrix}
  v_0 \\
  v_1 \\
  v_2
\end{bmatrix}$

Tensor

$\begin{bmatrix}
  \sigma_{00} & \sigma_{01} & \sigma_{02} \\
  \sigma_{10} & \sigma_{11} & \sigma_{12} \\
  \sigma_{20} & \sigma_{21} & \sigma_{22}
\end{bmatrix}$
**Fields**

- Difference between **continuous** and **discrete** values
- Examples: temperature, pressure, density
- **Grids** necessary to sample continuous data:

  ![Grids](image)

  - uniform
  - rectilinear
  - structured
  - unstructured

- **Interpolation**: “how to show values between the sampled points in ways that do not mislead”

[Weiskopf, Machiraju, Möller]
Spatial Data Example: MRI

[Image of MRI machine and grid]

[via Levine, 2014]
Scivis and Infovis

- Two subfields of visualization
- **Scivis** deals with data where the spatial position is given with data
  - Usually continuous data
  - Often displaying physical phenomena
  - Techniques like isosurfacing, volume rendering, vector field vis
- In **Infovis**, the data has no set spatial representation, designer chooses how to visually represent data