CSE 332
Introduction to Visualization

Introduction to D3

Klaus Mueller

Computer Science Department
Stony Brook University
<table>
<thead>
<tr>
<th>Lecture</th>
<th>Topic</th>
<th>Projects</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Intro, schedule, and logistics</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Applications of visual analytics, data types</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Data sources and preparation</td>
<td>Project 1 out</td>
</tr>
<tr>
<td>4</td>
<td>Data reduction, similarity &amp; distance, data augmentation</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Dimension reduction</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Introduction to D3</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Visual perception and cognition</td>
<td>Project 2 out</td>
</tr>
<tr>
<td>8</td>
<td>Visual design and aesthetic</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Visual analytics tasks</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Cluster analysis</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>High-dimensional data, dimensionality reduction</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Principal component analysis (PCA)</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Visualization of spatial data: volume visualization intro</td>
<td>Project 3 out</td>
</tr>
<tr>
<td>14</td>
<td>Introduction to GPU programming</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Visualization of spatial data: raycasting, transfer functions</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Illumination and isosurface rendering</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Midterm</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Scientific visualization</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Non-photorealistic and illustrative rendering</td>
<td>Project 4 out</td>
</tr>
<tr>
<td>20</td>
<td>Midterm discussion</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Principles of interaction</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Visual analytics and the visual sense making process</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Visualization of graphs and hierarchies</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Visualization of time-varying and streaming data</td>
<td>Project 5 out</td>
</tr>
<tr>
<td>25</td>
<td>Maps</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>Memorable visualizations, visual embellishments</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>Evaluation and user studies</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>Narrative visualization, storytelling, data journalism, XAI</td>
<td></td>
</tr>
</tbody>
</table>
The material presented in these slides is derived from this book:

Also available online
What is D3.js?

D3 = Data Driven Documents

JavaScript library for manipulating documents based on data
  ▪ frequent tool to support data journalism (New York Times)

D3 helps you bring data to life using HTML, SVG, and CSS
  ▪ great library to construct animated visualizations (D3 website)

Runs in any modern web browser (Chrome, Firefox, IE)
  ▪ no need to download any software
  ▪ independent of OS (Linux, Windows Mac)
Makes Use Of

- HTML  Hypertext Markup Language
- CSS   Cascading Style Sheets
- JS    JavaScript
- DOM   The Document Object Model
  - tree structured organization of HTML objects
- SVG   Scalable Vector Graphics

![Raster Images (.jpeg, .gif, .png)](image1)
![Vector Image (.svg)](image2)
What You Need

A text editor
- textMate, eclipse/aptana, sublime text 2...
- you can also use the native editor in chrome or firefox
- need an editor with syntax highlighting. else it’s easy to get lost

The d3 library
- from http://d3js.org

Data files for your code

A web server (recommended)
- if your visualization is reading data from files or a database (XMLHttpRequest)
- many options: EasyPHP (windows), Mac OS X Server, MAMP, Python
- else need to specify the data in the code

A browser
- to run the code
Your folder structure should look like this:

```
project-folder/
  d3/
    d3.v3.js
    d3.v3.min.js (optional)
index.html
```
Your initial webpage should look like this:

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>D3 Page Template</title>
    <script type="text/javascript" src="d3/d3.v3.js"></script>
  </head>
  <body>
    <script type="text/javascript">
      // Your beautiful D3 code will go here
    </script>
  </body>
</html>
```
MAMP = My Apache, MySQL, PHP
  - really only need Apache for now
  - MS Windows = WampServer and XAMPP for Windows
  - Mac = MAMP or XAMPP for Mac

Procedure
  - install package (Linux has it already installed)
  - find webserver folder (only files residing there will be served)
  - put project files there
  - open browser and point to http://localhost/ or http://localhost:8888/ or http://localhost:8888/project-folder/
var dataset = [ 5, 10, 15, 20, 25 ];
Consider the following js code ... all methods are chained:

d3.select("body").selectAll("p")
  .data(dataset)
  .enter()
  .append("p")
  .text("New paragraph!");

which gives this output
  - how did this happen?
Consider the following js code ... all methods are chained:

d3.select("body").selectAll("p") // selects all paragraphs in the DOM (none so far, but soon)
  .data(dataset)  // counts and parses the data values
  .enter()  // creates new, data-bound elements (placeholders) for the data values
  .append("p")  // takes the empty placeholder and adds a p-element
  .text("New paragraph!"); // takes the p-element and inserts a text value

which gives this output
- how did this happen?
Change the last line to:

```javascript
d3.select("body").selectAll("p")
  .data(dataset)
  .enter()
  .append("p")
  .text(function(d) { return d; });
```

which gives this output

- how did this happen?
Change the last line to:

d3.select("body").selectAll("p")
  .data(dataset)
  .enter()
  .append("p")
  .text(function(d) { return d; }); // used the data to populate the contents of each paragraph of the data-driven document

which gives this output
  - how did this happen?
Change the last line to:

```javascript
d3.select("body").selectAll("p")
  .data(dataset)
  .enter()
  .append("p")
  .text(function(d) { return "I can count up to " + d; });
```

which gives this output

- how did this happen?
Change the last line to:

d3.select("body").selectAll("p")
  .data(dataset)
  .enter()
  .append("p")
  .text(function(d) { return "I can count up to " + d; })
  .style("color", "red");

which gives this output
  - how did this happen?
More Complex Functions

Replace the last line with:

d3.select("body").selectAll("p")
  .data(dataset)
  .enter()
  .append("p")
  .text(function(d) { return "I can count up to " + d; })
  .style("color", function(d) { if (d > 15) {return "red"; } else { return "black"; } });

which gives this output

- how did this happen?
Let's draw some bar charts

For this, put this embedded style in the document head

div.bar {
    display: inline-block;
    width: 20px;
    height: 75px; /* We'll override height later */
    background-color: teal;
}
Run this code:

```javascript
var dataset = [ 5, 10, 15, 20, 25 ];

d3.select("body").selectAll("div")
 .data(dataset)
 .enter()
 .append("div")
 .attr("class", "bar");
```

which gives this output

- how did this happen?
- five bars with no space between them
Run this code:

```javascript
var dataset = [ 5, 10, 15, 20, 25 ];

d3.select("body").selectAll("div")
  .data(dataset)
  .enter()
  .append("div")
  .attr("class", "bar")
  .style("height", function(d) { return d + "px"; })
```

which gives this output

- how did this happen?
Run this code:

```javascript
var dataset = [ 5, 10, 15, 20, 25 ];

d3.select("body").selectAll("div")
  .data(dataset)
  .enter()
  .append("div")
  .attr("class", "bar")
  .style("height", function(d) { return d + "px"; }) // adds text “px” to specify that the units are pixels → heights are 5px, 10px, 15px, 20px, and 25px
```

which gives this output

- how did this happen?
Run this code: (also add margin-right: 2px; to the css style)

```javascript
var dataset = [ 5, 10, 15, 20, 25 ];

d3.select("body").selectAll("div")
 .data(dataset)
 .enter()
 .append("div")
 .attr("class", "bar")
 .style("height", function(d) { var barHeight = d * 5; return barHeight + "px"; });
```

which gives this output

- how did this happen?
Optionally define some variable beforehand, e.g.:

```javascript
// width and height
var w = 500;
var h = 50;

Define the svg object:

```javascript
var svg = d3.select("body")
 .append("svg")
 .attr("width", w)
 .attr("height", h);
```
Define the circles as variables for ease of reference:

```javascript
var circles = svg.selectAll("circle")
    .data(dataset)
    .enter()
    .append("circle");
```

But could so this just as well:

```javascript
svg.selectAll("circle")
    .data(dataset)
    .enter()
    .append("circle"); // now circles are appended to the end of the SVG element
```
Run this code (still using var dataset = [ 5, 10, 15, 20, 25 ];)

circles.attr("cx", function(d, i) {return (i * 50) + 25;})
  .attr("cy", h/2)
  .attr("r", function(d) {return d;});

or append it to the .append("circle") method

This gives this output
  • how did this happen?
Run this code (still using var dataset = [ 5, 10, 15, 20, 25 ];)

circles.attr("cx", function(d, i) {return (i * 50) + 25;})  // i increments by 1 each time, starting at 0
  .attr("cy", h/2)
  .attr("r", function(d) {return d;});

or append it to the .append("circle") method

This gives this output
  - how did this happen?
Run this code (still using var dataset = [ 5, 10, 15, 20, 25 ];)

circles.attr("cx", function(d, i) {return (i * 50) + 25;})
  .attr("cy", h/2)
  .attr("r", function(d) {return d;})
  .attr("fill", "yellow")
  .attr("stroke", "orange")
  .attr("stroke-width", function(d) {return d/2;});

This gives this output
  - how did this happen?
Code
This will update the bar chart on a **mouse click**:

```javascript
d3.select("p")
  .on("click", function() {

    //New values for dataset
    dataset = [ 11, 12, 15, 20, 18, 17, 16, 18, 23, 25, 5, 10, 13, 19, 21, 25, 22, 18, 15, 13 ];

    //Update all rects
    svg.selectAll("rect")
      .data(dataset)
      .attr("y", function(d) {
        return h - yScale(d);
      })
      .attr("height", function(d) {
        return yScale(d);
      });

  });
```

Smooth animations are desirable:

```javascript
svg.selectAll("rect")
  .data(dataset)
  .transition()
  .attr("y", function(d) {
    return h - yScale(d);
  })
  .attr("height", function(d) {
    return yScale(d);
  })
  .attr("fill", function(d) {
    return "rgb(0, 0, " + (d * 10) + ")";
  });
```
Now run this code:

```javascript
svg.selectAll("rect")
  .data(dataset)
  .transition()
  .duration(1000)  // <-- Now this is new!
  .attr("y", function(d) {
    return h - yScale(d);
  })
  .attr("height", function(d) {
    return yScale(d);
  })
  .attr("fill", function(d) {
    return "rgb(0, 0, " + (d * 10) + ")";
  });
```
Facilitated by event handlers (listeners), e.g.:

d3.select("p")
  .on("click", function() {
    //Do something on click
  });

others react on
  - mouse hovering
  - mouse over
  - mouse out
  - and others

Example
Assume you selected a certain item by mouseover

```javascript
.on("mouseover", function() {
    //Do something on mouseover of any bar

});
```

Keyword “this” maps the action to the selected item

```javascript
.on("mouseover", function() {
    d3.select(this)
        .attr("fill", "orange");

});
```
D3 layouts take data that you provide

- remap or otherwise transform it
- and so generating new data that is more convenient for a specific visual task

The supported layouts are:

- Bundle and Chord
- Cluster
- Force
- Histogram
- Pack, Partition, and Pie
- Stack
- Tree and Treemap
var dataset = {
    nodes: [
        {name: "Adam"},
        {name: "Bob"},
        {name: "Carrie"},
        {name: "Donovan"},
        {name: "Edward"},
        {name: "Felicity"},
        {name: "George"},
        {name: "Hannah"},
        {name: "Iris"},
        {name: "Jerry"}
    ],
    edges: [
        {source: 0, target: 1},
        {source: 0, target: 2},
        {source: 0, target: 3},
        {source: 0, target: 4},
        {source: 1, target: 5},
        {source: 2, target: 5},
        {source: 2, target: 5},
        {source: 3, target: 4},
        {source: 5, target: 8},
        {source: 5, target: 9},
        {source: 6, target: 7},
        {source: 7, target: 8},
        {source: 8, target: 9}
    ];
}
var force = d3.layout.force()
    .nodes(dataset.nodes)
    .links(dataset.edges)
    .size([w, h])
    .linkDistance([50]) // <-- New!
    .charge([100]) // <-- New!
    .start();

Next, we create an SVG line for each edge:

var edges = svg.selectAll("line")
    .data(dataset.edges)
    .enter()
    .append("line")
    .style("stroke", ".ccc")
    .style("stroke-width", 1);

Note that I set all the lines to have the same stroke color and weight, but of course you could set this dynamically based on data (say, thicker or darker lines for "stronger" connections, or some other value).

Then, we create an SVG circle for each node:
Too many data items?
- k-means clustering followed by stratified sampling

Too many attributes?
- PCA followed by dimension projection and scree plot culling

Not enough data items or attributes?
- data synthesis using data augmentation

Use Python library scikit-learn
- lots of simple and efficient tools for data mining and data analysis
- is available on the department’s Linux server via ssh login into allv28.all.cs.stonybrook.edu (port 130)
- use putty to login and upload

You will earn 10 pts for whatever option you pursue
- 10 pts are regular credit, additional options earn 10 pts extra credit each

Submission via conference website
Due Tuesday, Oct. 11, 2016
Replace console name by an ID, like Nintendo DS • 1, Sony PSP • 2,...
Replace the title by an ID, just label 1, 2, 3, ....
Likewise replace all text strings by ID
You can consolidate related attributes like HistoricalBattle, Horror, InteractiveFiction (which are genres in this case) into a single attribute and give them an ID instead, such as Horror • 1, HistoricalBattle • 2, ...,
Visualize the data you secured in project #1 using D3 bar charts

Your D3-based visual interface should be able to (all 10 pts):
1. pick a variable and bin it into a fixed range (equi-width) of your choice
2. create a bar chart of the variable you picked in 1.
3. respond to mouse clicks to cycle through all variables and update chart
4. only on mouse-over, display the value of the bar on top of the bar
5. on mouse-over, make the bar wider and higher to focus on it
6. create a drop-down menu to allow users to select the variable shown

An additional 10 pts for elegant implementation/function

Extra credit (10 pts):
- mouse moves left (right) should decrease (increase) bin width/size

Submission via conference website
Due Tuesday, Oct. 11, 2016
You need to upload the following by the due date

- 2-3 page report with illustrated description of your program’s capabilities and implementation detail
- data processing (2.1) and visualization (2.1)
- your can add code snippets as an appendix
- mpeg or avi video file that shows your software in action
- as before, do not reveal your identity on your submission

Grading

- TA will pick students at random for thorough code review sessions
- you better know your code !!!
- so, please do not just copy code beyond the D3 templates
- or even worse, videotape someone else’s program
Aka, cheating

Discussion with your class mates (but not others) is OK

Cut and paste from any source is **not** OK
- any suspected activity of this kind will result in zero points
- also for the person providing the original
- two-strikes and out rule is in effect (including an academic misconduct report)
- this includes any feeble attempt to cover the tracks somehow

Stay honest and resist the temptation!
Evaluating Project #1

You will all be assigned 3 random peer project reports
  ▪  you will serve as a reviewer
  ▪  we will send email on specific login procedures and deadlines

Two review criteria:
  ▪  Does the report describe the chosen data and their domain well?
  ▪  Does the report describe the prospects of the data for interesting insight well?

Review will be along these two dimensions:
  ▪  scoring with the Likert scale
  ▪  verbal comments

<table>
<thead>
<tr>
<th>Strongly Disagree</th>
<th>Disagree</th>
<th>Undecided</th>
<th>Agree</th>
<th>Strongly Agree</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>(2)</td>
<td>(3)</td>
<td>(4)</td>
<td>(5)</td>
</tr>
</tbody>
</table>