CSE 332
Introduction to Visualization

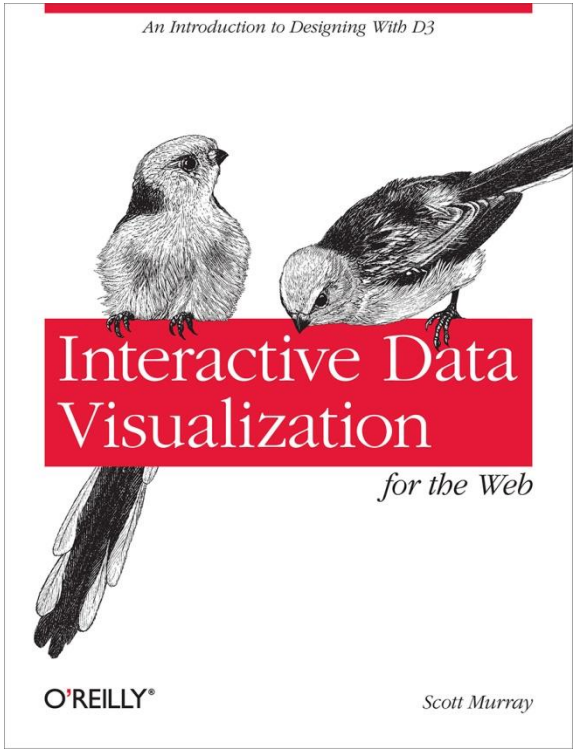
Introduction to D3

Klaus Mueller

Computer Science Department
Stony Brook University

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

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

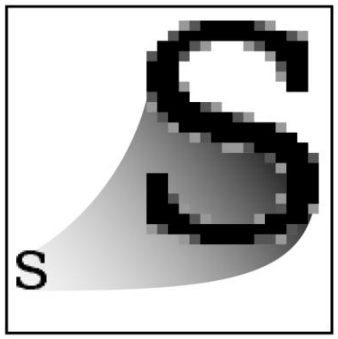
CSS Cascading Style Sheets

JS JavaScript

DOM The Document Object Model

- tree structured organization of HTML objects

SVG Scalable Vector Graphics



Raster

.jpeg .gif .png

Vector

.svg

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

Setup

Your folder structure should look like this:

```
project-folder/
    d3/
        d3.v3.js
        d3.v3.min.js (optional)
    index.html
```

Setup

Your initial webpage should look like this:

```
<!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>
```

Setting up a webserver

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/

Let's Use Some Simple Data

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

Chaining

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?

New paragraph!

New paragraph!

New paragraph!

New paragraph!

New paragraph!

Chaining

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?

New paragraph!

New paragraph!

New paragraph!

New paragraph!

New paragraph!

Using the Data

Change the last line to:

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

which gives this output

- how did this happen?

5

10

15

20

25

Using the Data

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?

5

10

15

20

25

Using Functions

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; });
```

which gives this output

- how did this happen?

I can count up to 5

I can count up to 10

I can count up to 15

I can count up to 20

I can count up to 25

Adding Aesthetics

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?

I can count up to 5

I can count up to 10

I can count up to 15

I can count up to 20

I can count up to 25

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?

I can count up to 5

I can count up to 10

I can count up to 15

I can count up to 20

I can count up to 25

Drawing With Data

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;
}
```

Simple Bar Chart

Run this code:

```
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

More Complex Bar Chart

Run this code:

```
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?

More Complex Bar Chart

Run this code:

```
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?

Make Bars Taller and Add Spaces

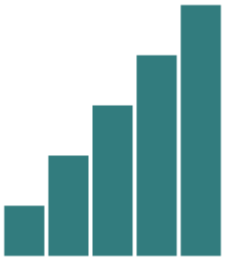
Run this code: (also add margin-right: 2px; to the css style)

```
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?



Get Ready to Draw Some SVGs

Optionally define some variable beforehand, e.g.:

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

Define the svg object:

```
var svg = d3.select("body")
            .append("svg")
            .attr("width", w)
            .attr("height", h);
```

get Ready to Draw Some SVG Circles

Define the circles as variables for ease of reference:

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

But could so this just as well:

```
svg.selectAll("circle")
   .data(dataset)
   .enter()
   .append("circle"); // now circles are appended to the end of the SVG element
```

Now Draw The Circles

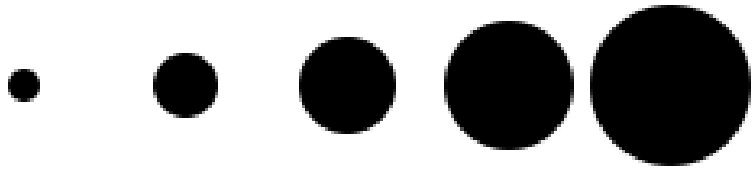
Run this code (still using var dataset = [5, 10, 15, 20, 25];)

```
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?



Now Draw The Circles

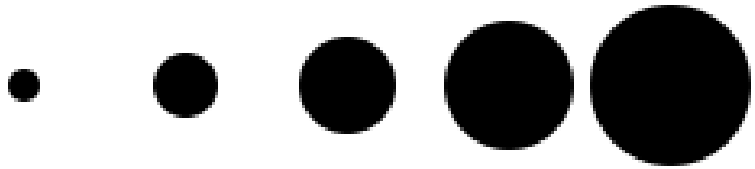
Run this code (still using var dataset = [5, 10, 15, 20, 25];)

```
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?



Adding Colors

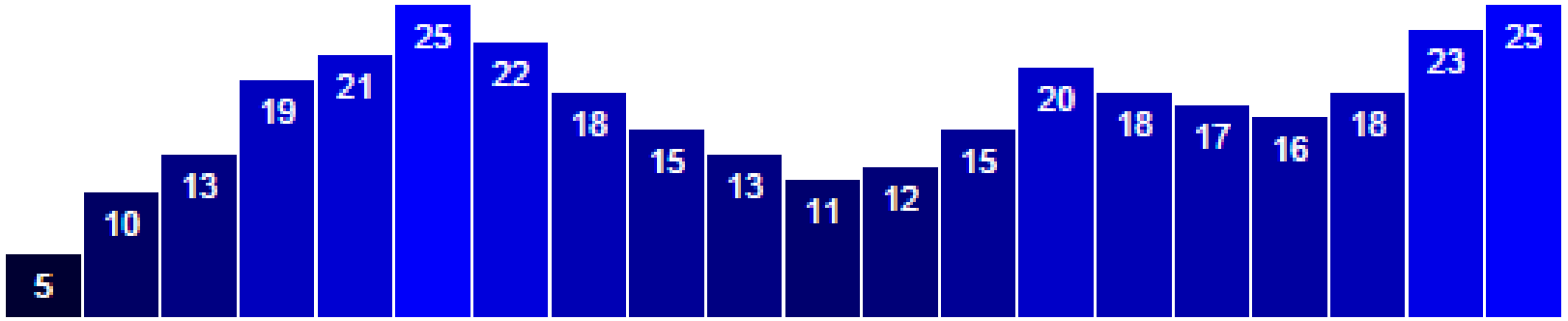
Run this code (still using var dataset = [5, 10, 15, 20, 25];)

```
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?

Bar Charts



Code

Updates

This will update the bar chart on a mouse click:

```
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);
           });

    });
```

Adding Animated Transitions

Smooth animations are desirable:

```
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) + ")";
   });
```

Control Duration

Now run this code:

```
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) + ")";
   });
```

Interaction

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

Directing Action to a Specific Item

Assume you selected a certain item by mouseover

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

Keyword “this” maps the action to the selected item

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

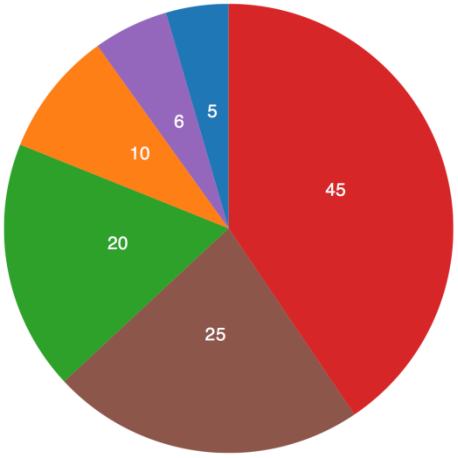
Layouts

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



Force-Directed Layout

```
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).

Demo

Then, we create an SVG circle for each node:

```
var nodes = svg.selectAll("circle")
        .data(dataset.nodes)
        .enter()
        .append("circle")
        .attr("r", 10)
        .style("fill", function(d, i) {
                return colors(i);
        })
        .call(force.drag);
```

Project #2.1: Data Preparation

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

Submission via conference website
Due Tuesday, Oct. 11, 2016

You will earn 10 pts for whatever option you pursue

- 10 pts are regular credit, additional options earn 10 pts extra credit each

Data Files

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 HistoricalBatte, Horror, InteractiveFiction (which are genres in this case) into a single attribute and give them an ID instead, such as Horror・1, HistoricalBattle・2, …,

Project #2.2: data Visualization

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. cerate a drop-down menu to allow users to select the variable shown

An additional 10 pts for elegant implementation/function

Submission via conference website
Due Tuesday, Oct. 11, 2016

Extra credit (10 pts):

- mouse moves left (right) should decrease (increase) bin width/size

Deliverables

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

Seeking Outside Help

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

Strongly Disagree	Disagree	Undecided	Agree	Strongly Agree
(1)	(2)	(3)	(4)	(5)