INFORMATION ORGANIZATION LAB

IO LAB

INFORMATION ORGANIZATION LAB
As a reminder, we’re talking about visualization in this portion of the course because we want to have tools to analyze tagging practices, although you’re not required to do any kind of visualization as part of project 4.
VISUALIZATION GOALS

Exploration

- Trying to understand a data set using visual means

Communication

- Already understand a data, and want to communicate something about it visually

(This distinction is originally from J. Bertin)

Examples

Communication: http://vis.stanford.edu/protovis/ex/crimea-rose-full.html
KINDS OF DATA

- Nominal ("single", "married", "divorced", "widowed")
- Ordered ("dislike strongly", "dislike", "neutral", "agree", "agree strongly")
- Quantitative (64°, 32°, 18°, 105°)

- Nominal data can only be compared as “equal” or “not equal”
- Ordered data have a a logical progression, but may not correspond to quantitative values
- Quantitative data
Some of these features allow for “preattentive processing.” See [http://www.csc.ncsu.edu/faculty/healey/PP/index.html](http://www.csc.ncsu.edu/faculty/healey/PP/index.html).
How many times bigger is the circle on the right than the circle on the left?

**Left circle**
- diameter = 120
- radius = 60
- area = \(3.14 \times 60^2 = 11304\) sq. pixels

**Right circle**
- diameter: 300
- radius: 150
- area: \(3.14 \times 150^2 = 70650\) sq. pixels

\[\frac{70650}{11304} = 6.25\] times bigger
PERCEIVING LENGTH

How many times longer is the bottom line compared with the top line?

Top line
length: 100px

Bottom line
length: 600px
Another problem with area is that it’s so easy to get wrong, not just from the perception side, but from the creation side.

Let’s say we wanted to represent 100, 200, and 300 with area. How would we go about doing that?

\[ (A=100, 200, 300 \implies r = \sqrt{A/\pi}) \]
CREATION PROCESS

- What am I trying to show? If a visualization is a good way to show this
- What visual variables will I use to show these this?
- What data do I need? How do I get it?
- Data transformation (normalization, historical adjustment, etc.)
- Display

To make a basic visualization, you have to go through these steps.

Once you have the data, you need to get it into the format that the tool you're using wants. This usually requires creating an appropriate data structure, at least. It may also require some sampling transformation, like reducing the number of data points.
DO YOU NEED A VISUALIZATION?

**Blog Activity over 7 Days**
May 17 to May 24

1 comment
1,141 spam comments

“over 1000 spam comments per real comment”
Raster images are also called bitmaps.
It's said that “Raster is faster, but vector is better.” People also say, “Raster is faster, but vector is correcter.” But that sounds dumb.
GRANULARITY

Don’t store too many data points in too few pixels.

Too few pixels is a concern even if you’re using a vector format like SVG because ultimately the image is rendered at a certain screen size.

http://code.google.com/apis/chart/formats.html#granularity

This starts to get us into an issue that we’ll discuss more next week.
- SVG = scalable vector graphics, like EPS in XML.
- You can provide support for canvas in IE without the end-user installing a plugin by using the excanvas library.
- You can translate SVG to a Microsoft
- An advantage of SVG is that each element is part of the DOM, and so you can manipulate individual elements, have events for them, etc.
- Canvas is part of HTML5.
INTERFACE OPTIONS

- Write raw SVG or canvas code
- Use a visualization library
- Use a chart and graph library

Raw XML
Processing and Processing.js are raster graphics libraries. Processing.js. Uses the HTML `<canvas>` element. These are general visualization toolkits (as opposed to specialized graphing/charting toolkits)

http://processingjs.org/
Raphael is a vector graphics libraries.
Uses `<svg>` elements.
See [http://raphaeljs.com/icons/](http://raphaeljs.com/icons/) as a demonstration of SVG scalability
Google Chart API

http://code.google.com/apis/chart/

You can simply use image tags instead of either canvas or SVG.
Google Chart API

A simple bar chart

Doesn’t use SVG or <canvas> -- it just returns an image!

http://code.google.com/apis/chart/
Raphael and gRaphael are vector graphics libraries. Uses SVG.
r.g.barchart(10, 10, 300, 220,
  [[55, 20, 13, 32, 5, 1, 2, 10]],
  0,
  {type: "sharp"});

One line!
Bare bones.
PROTOVIS
Minard’s Napoleon

Uses SVG
Jeff Heer and Mike Bostock (from Stanford and (formerly) Berkeley)
vis.add(pv.Bar)
data([1, 1.2, 1.7, 1.5, .7, .3])
.width(20)
.height(function(d) d * 80)
.bottom(0)
.left(function() this.index * 25);

PROTOVIS
A simple bar chart

It’s chainable!
Inline functions!
Feature of Protovis.
pv is the namespace

Scale data: pv.Scale
Linear: pv.Scale.linear(domain, domain).range(range, range)
Return a function that lets you scale data
MORE ON VISUALIZATION

This has been a brief overview of the broad (and growing) field of visualization.

If you’re interested in more of the theory of visualization:
- CS-294, offered in the spring. See http://vis.berkeley.edu/
- Info 247, Information Visualization and Presentation. See http://www.ischool.berkeley.edu/courses/247
- Books by Edward Tufte, Stephen Few.