

## Lecture 19: Information Visualization

Content in this lecture indicated as "All Rights Reserved" is excluded from our Creative Commons license. For more information, see <http://ocw.mit.edu/fairuse>.

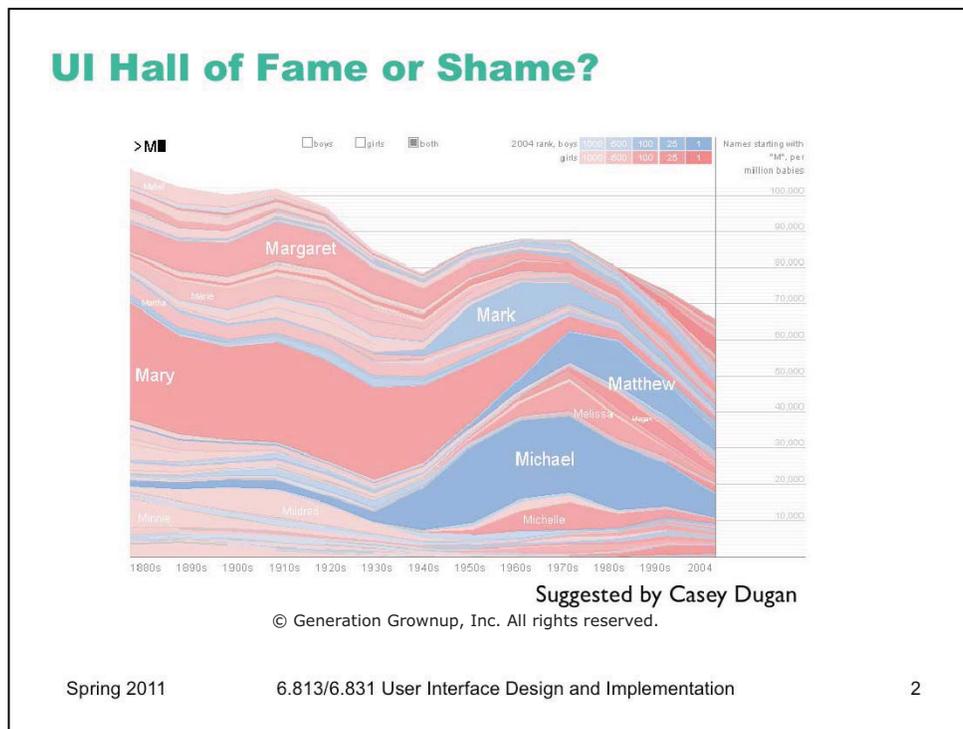
Spring 2011

6.813/6.831 User Interface Design and Implementation

1

Related reading for this lecture is Shneiderman, "The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations", *VL 1996*.

<http://cs.ubc.ca/~tmm/courses/cpsc533c-05-fall/readings/shneiderman96eyes.pdf>



This web site is the Baby Name Wizard’s Name Voyager ([www.babynamewizard.com](http://www.babynamewizard.com)).

It’s a great example of direct manipulation. The baby names have a continuous visual representation, which I can control by physical actions (clicking on a colored line). The text entry field is an incremental search, so it has rapid, incremental, and reversible effects.

Alas, some actions do *not* have easily reversible effects. Clicking on a name, like Michael, can’t be reversed by another click or single keypress; instead, I have to backspace to erase the whole word “Michael” in order to get back to where I was before. That makes it much harder to explore, and makes me (as a user) unwilling to click on something, because the cost of reversing the click isn’t worth the benefit I get from zooming in on one name.

Another unfortunate limitation is that I can’t zoom or filter the time line. Knowing how popular Mary was in the 1880’s is historically interesting, but if I’m trying to choose a name for my baby, the last 10 or even 5 years may be much more important to me, so I’d want to use as much more length of the y axis position for that range.

Another issue is that you can’t easily compare names that don’t share a common prefix – but comparing names is very important to new parents. This may be a failure of task analysis.

What visual variables are used here?

- **hue** for gender (a nominal variable)
- **saturation** for popularity rank (an ordinal variable)
- **size** (vertically) for frequency (a quantitative variable)
- **position** (horizontally) for time (a quantitative variable)

## Today's Topics

- Motivation
  - why we visualize, historical examples
- Using space well
  - scatterplots, treemaps, tag clouds
- Interactivity
  - overview, zoom & filter, details-on-demand
- Toolkits for visualization

Today's lecture is about information visualization. One way to define that is how to display large amounts of information in such a way that users can easily perceive it and reason about it – which typically means *visually*, rather than textually. We'll see that the visual variables we talked about last time will come back again today.

After motivating the problem and looking at some famous visualizations from history (well, both famous and infamous), we'll look at a few traditional visualizations and few that you haven't seen before, thinking in particular about how to pack information into limited space. We'll also think about how computer visualizations, unlike print, can be **interactive**, and what that should mean. We'll close with a list of useful resources, toolkits that provide some of the visualizations we've looked at as reusable widgets.

## Why Visualization?

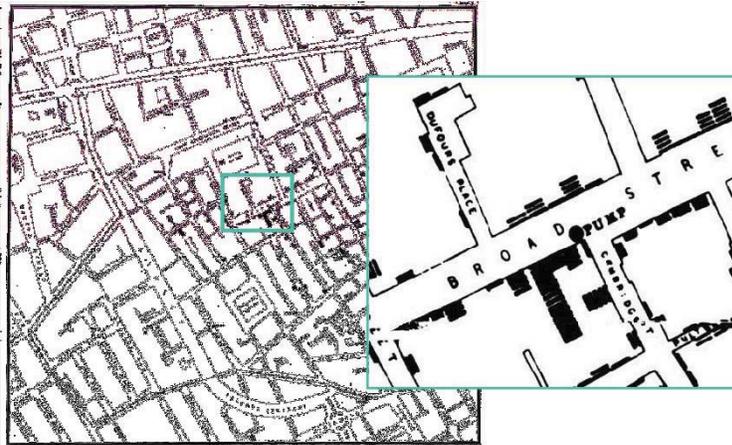
- Analyze
  - Think with your perceptual system
  - Play with the data to understand it
- Present
  - Convey information to others
  - Collaborate
  - Highlight important facts
  - Make a decision

Why do we want to visualize information? Basically two reasons: to **think**, and to **communicate**.

A visualization is a form of external cognition. Just like writing down a column of numbers helps you add them, drawing a picture of data in the right way helps you see patterns and relationships that would otherwise be invisible. Your perceptual system contributes to the thinking process when your data is visualized.

For a similar reason, visualization can be a great way to communicate information to other people, to think together, and to make a shared decision.

## Cholera Outbreak



© Edward Rolf Tufte. All rights reserved.

Spring 2011

6.813/6.831 User Interface Design and Implementation

7

Here's a classic visualization that did both: thinking and communication. In 1854, a cholera outbreak swept a neighborhood in London. The main theory of the disease at the time assumed that a "miasma", or bad air, was responsible for cholera. John Snow, a local doctor, believed otherwise. He created this simple plot on a map of the neighborhood, with each dot at a street address representing a case of cholera, and found a heavy cluster of cases around a public water pump on Broad Street. Using this plot, he persuaded local officials to remove the handle of the infected water pump, so that people would stop using it.

[Source: Tufte, *The Visual Display of Quantitative Information*, 1983.]

## Challenger Disaster: a Poor Display

HISTORY OF O-RING DAMAGE ON SRM FIELD JOINTS

SRM No.	Cross Sectional View			Top View		Clocking Location (deg)
	Friction Device (in.)	Parameter Affecting (deg)	Neutral Size (in.)	Length of Seal Friction (in.)	Heat Affected Length (in.)	
61A LH Center Field**	22A	None	None	0.280	None	161-165
61A LH Forward Field**	22A	None	None	0.280	None	330-334
61C LH Forward Field**	16A	0.110	114.0	0.280	4.25	143
61C RH Center Field (prim)**	15B	0.280	110.0	0.280	12.50	334
61C RH Center Field (sec)**	15B	None	45.0	0.280	None	29.50
41D RH Forward Field	33B	0.228	110.0	0.280	3.00	215
41C LH Aft Field*	11A	None	None	3.200	None	--
41B LH Forward Field	10A	0.040	117.0	0.280	3.00	311
575-2 RH Aft Field	2B	0.053	116.0	0.280	--	30

\*Hot gas path detected in putty. Indication of heat on O-ring, but no damage.  
 \*\*Soot behind primary O-ring.  
 \*\*\*Soot behind primary O-ring, heat affected secondary O-ring.  
 Clocking location of leak check part - 0 deg.

OTHER SRM-15 FIELD JOINTS HAD NO BLOWHOLES IN PUTTY AND NO SOOT NEAR OR BEYOND THE PRIMARY O-RING.  
 SRM-22 FORWARD FIELD JOINT HAD PUTTY PATH TO PRIMARY O-RING, BUT NO O-RING EROSION AND NO SOOT BLOWBY. OTHER SRM-22 FIELD JOINTS HAD NO BLOWHOLES IN PUTTY.

SRM No.	HISTORY OF O-RING TEMPERATURES (DEGREES - F)				WIND
	MOTOR	CRAT	AMB	O-RING	
SRM-15	DM-1	49	36	47	10 MPH
	DM-2	76	45	52	10 MPH
	DM-3	72.5	40	45	10 MPH
	DM-4	76	45	51	10 MPH
SRM-15	SRM-15	52	64	53	10 MPH
SRM-22	SRM-22	77	78	75	10 MPH
SRM-25	SRM-25	85	76	87	10 MPH
				27	25 MPH

Blow by history  
 SRM-15 worse blow-by  
 o 2 case joints (30'), (10') sec  
 o MUCH WORSE VIBRANT THAN SRM-22

SRM-22 Blow-by  
 o 2 case joints (30-40')

SRM-15A, 15, 16A, 15, 23A 24A  
 o NOZZLE Blow-by

[Tuft 97]

Spring 2011

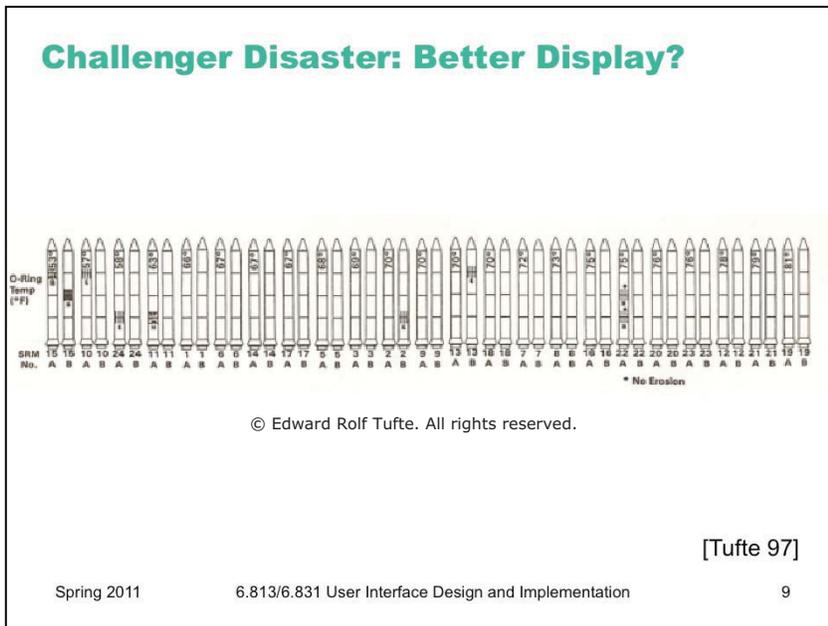
© Edward Rolf Tuft. All rights reserved.  
 6.813/6.831 User Interface Design and Implementation

8

A case where communication may have failed partly due to poor visualization was the Space Shuttle Challenger, which exploded on takeoff in 1986. The cause of the explosion was a breached O-ring in one of its rocket boosters. The O-ring failed because temperatures were well below freezing when the shuttle launched, and the material of the O-ring wasn't resilient enough to maintain the seal. The key mistake was the decision to launch the shuttle in such cold temperatures; and the key question is, did the decision makers have the information they needed, but just didn't look at it in a way that the risk would be clearly apparent?

First, to establish a baseline, here's one way to present the data: in tabular, textual form. Some of the information used in the decision was presented this way. Tables are a form of visualization, but a weak one. They don't exploit all the capabilities of our visual systems to perceive and to think; they don't exploit all the dimensions of our visual systems.

Source: Tuft, *Visual Explanations*, 1997.



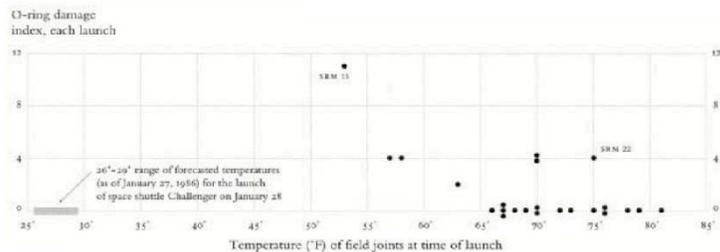
Here's another way to present it (a reproduction of a visualization that was actually used in the decision-making process before the Challenger launch). Each pair of rockets represents an actual pair of rockets (A and B) launched on a prior shuttle mission (SRM, numbered 1 to 24). The O-rings are the horizontal lines on the rockets, and smudges on some of the O-rings show where O-ring damage was observed after the rocket booster was recovered. (Shuttle rockets fall into the ocean where they are recovered and reused, so engineers are able to inspect them for damage after the launch.)

The launches are ordered by air temperature at launch (written sideways on the top of the rocket).

Let's discuss this visualization from a usability point of view. What's good or bad about it from a learnability point of view? What about visibility – what does it make visible, and what's still hidden or implicit? What kinds of errors might be made?

Source: Tufte, *Visual Explanations*, 1997.

## Challenger Disaster: A Better Viz



© Edward Rolf Tufte. All rights reserved.

[Tufte 97]

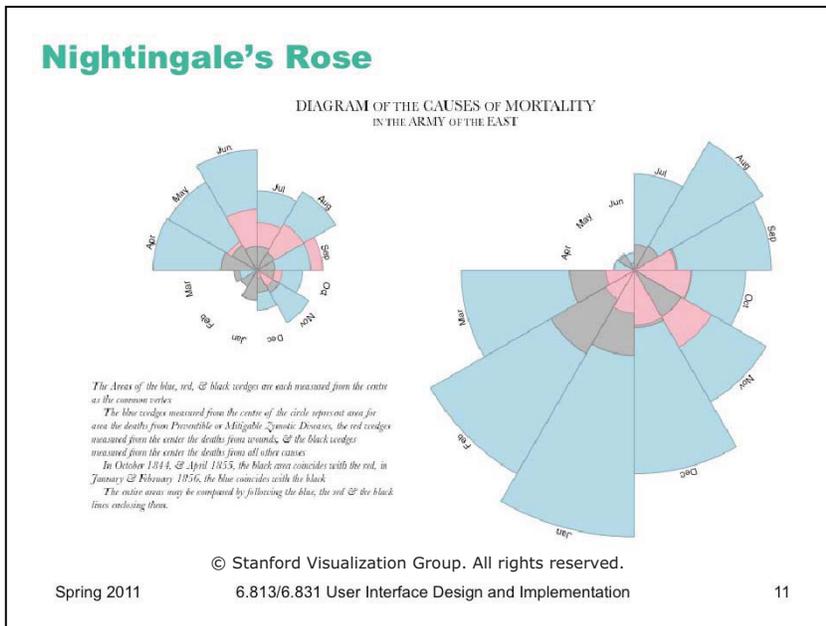
Spring 2011

6.813/6.831 User Interface Design and Implementation

10

Here is Edward Tufte's redesign of the visualization, showing the same data. A couple of things are worth noting. First, it dramatically simplifies the display, doing away with the rocket pictures (which were very suggestive, metaphorical, and learnable) in favor of much stronger visibility of the essential information (temperature and O-ring damage). Second, it changes the x-position from merely showing the **ordering** of the temperature variable, and instead uses it to show temperature in a **quantitative** way. These changes have a dramatic effect on the story told by the display, especially when we add the additional piece of information that the current temperature (when the decision makers were considering the launch) was well below freezing – far to the left on the graph. A shuttle had never before been launched in such cold temperatures, and O-ring damage was already showing a suggestive pattern of damage even at temperatures 20-30 F higher.

Source: Tufte, *Visual Explanations*, 1997.

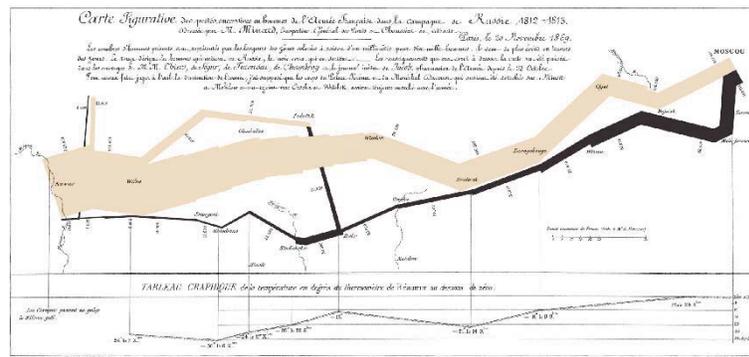


Florence Nightingale (who founded the Red Cross) invented this visualization in the 1850s to argue to the British Government that their army was suffering far more deaths from disease (blue) than from the battlefield (pink), and should be improving sanitation in its camps and hospitals. The right picture shows the first year of the war, when sanitation was particularly bad; the left side continues the second year of the war, after changes were made.

This kind of graph, now called a polar area diagram, coxcomb, or “Nightingale rose”, actually has a bit of deception in it. The number of deaths is actually proportional to the *radius* of a wedge, but the eye tends to interpret the whole area of a wedge as information-carrying. So putting the disease deaths on the outside of the coxcomb exaggerates their size.

Source: <http://vis.stanford.edu/protovis/ex/crimea-rose.html>

## Napoleon's March



This image is in the public domain.

Spring 2011

6.813/6.831 User Interface Design and Implementation

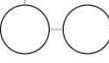
12

Our final historical example is this rich, multidimensional display of Napoleon's campaign into Russia (brown), and subsequent retreat (black). Made in 1869 by Charles Minard, this visualization packs in a lot of information:

- the path of the army (including several detachments) through an abstracted map of central Europe, with key cities and rivers marked;
- the size of the army (the width of the path)
- its direction (brown for attacking, black for retreating)
- the temperature during the retreat (because winter had set in, and cold weather was responsible for much of the further decimation of the army)

This display is widely admired as an information display that tells a remarkable story, more succinctly than words alone can do.

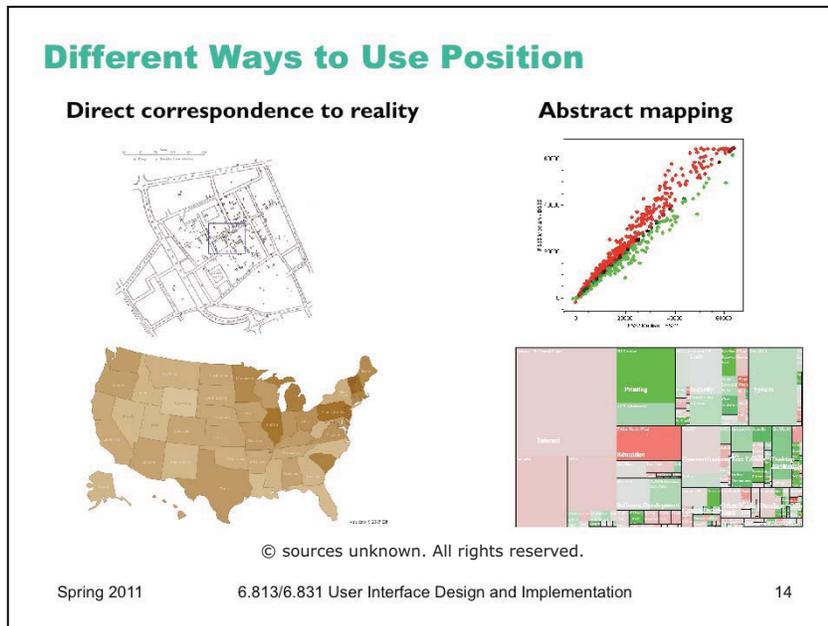
### Recall the Visual Variables

value	hue	texture	shape	position	orientation	size
						
						
						

- Map data attributes to appropriate visual variables
  - Nominal, ordered, or quantitative (NOQ)
  - N: all variables
  - O: value, texture, position, size
  - Q: position, size

Spring 2011      6.813/6.831 User Interface Design and Implementation      13

It's worth at this point reviewing our discussion of visual variables from last time. An information visualization uses some or all of these variables to display data attributes. We had two important properties of variables (whether visual or data): their type (N, O, or Q) and their length (number of distinct values). Which visual variables were used in the Napoleon march display? In the Nightingale display?

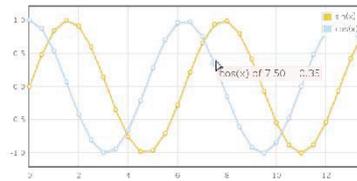
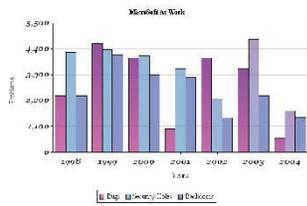


Position is the primal visual variable in infviz. It's the most versatile (in length and type), it pops out most strongly, and it's the most selective (since the spotlight of your attention is easy to move around your visual field). So it's often the scarcest resource as well. Infviz often involves figuring out how to pack as much information into the available pixels as you can.

Sometimes the most appropriate use of position is simply a direct mapping of real space, as in the maps on the left. If the data items have spatial coordinates, and physical location is actually relevant to the thinking or communicating task the user is trying to do, then this works.

When location is *not* relevant, however, we have to turn to abstract mappings – using x and y positions to represent other variables (as in the scatterplot on the top right), or simply using them to pack related information near each other (as in the treemap on the bottom right, which we'll say more about in a moment).

## Traditional Info Viz: Charts



© sources unknown. All rights reserved.

Spring 2011

6.813/6.831 User Interface Design and Implementation

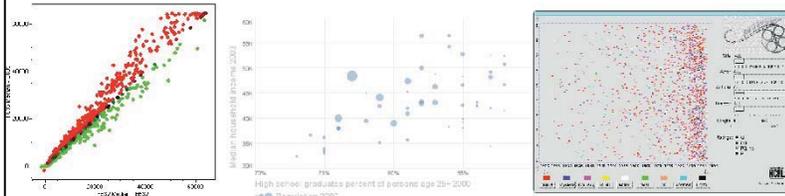
15

Traditional charts use position in exactly this abstract way. Bar charts, line charts, and pie charts have the advantage of learnability – they’re taught in schools and seen in a variety of media.

Let’s stop for a moment and think about the **tasks** that a user does with a visualization. What kinds of tasks do bar charts, line charts, and pies make easy or hard?

## Scatterplot

- maps any pair of attributes to x-y axes
  - usually quantitative or ordered attributes
  - usually the x-y attributes are related somehow
  - other attrs can be mapped to hue, value, size, shape



© sources unknown. All rights reserved.

Spring 2011

6.813/6.831 User Interface Design and Implementation

16

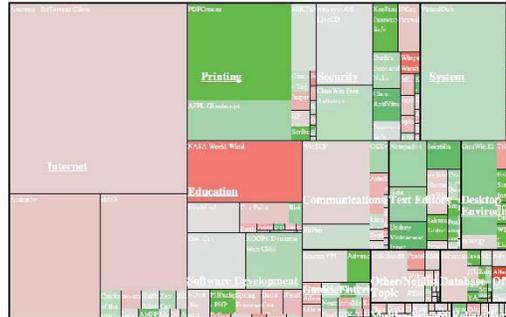
A more generic form of chart, widely used in information visualization, is a scatterplot, which represents each data item as a mark whose (x,y) position is determined by two of its attributes. Usually these two attributes are related somehow, so that the cloud of points will have a visible and useful high-level pattern that conveys information (like the upward-sloping line seen on the left). But that isn't strictly necessary – the scatterplot on the right shows movies, with the year of the movie as the x coordinate and the rating of the movie as the y coordinate. There's no correlation between those variables, but assigning them to position gives them a place on the screen and makes it possible for the user to get overviews of each variable independently.

Other visual variables are used to map other data attributes into each point.

When the display gets too cluttered and points fall on top of each other, one approach is jitter or point displacement, moving overlapping points slightly apart, sacrificing positional precision for a more accurate display of density. (See Ellis & Dix, "A Taxonomy of Clutter Reduction for Information Visualisation", *IEEE Transactions on Visualization & Computer Graphics*, 2007.)

## Treemap

- Space-filling visualization of a tree
  - each leaf has a quantitative attribute => size and other attribute => hue or value



© source unknown. All rights reserved.

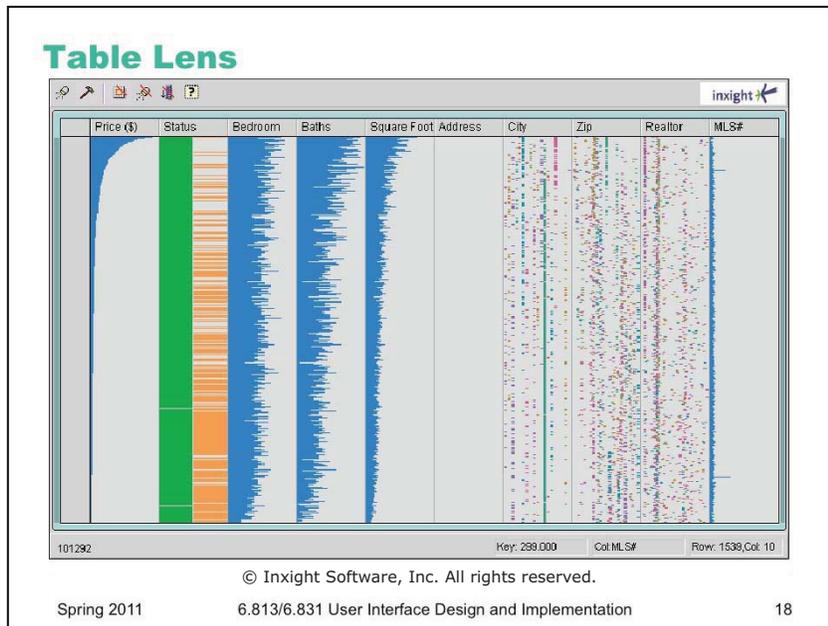
Spring 2011

6.813/6.831 User Interface Design and Implementation

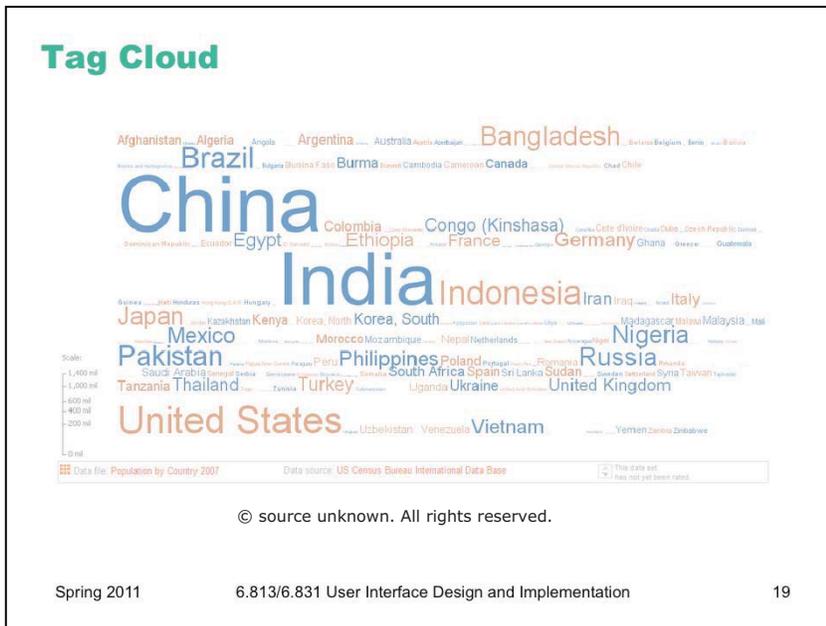
17

A treemap is one approach for displaying a tree in a compact rectangular space. It's designed for hierarchical data where the leaves have an important quantitative attribute that should be mapped to size (size as in area, not just linear dimension). Consider visualizing the space used by a filesystem, where the tree is the folder structure and the size of a node is its size in bytes. Or consider the treemap shown here, which visualizes the popularity of open source projects on SourceForge. The tree has two levels: application types (like Education) contain applications (like NASA World Wind). The size of a node is the number of downloads in a particular month, and the hue (green or red) reflects the change in downloads since the previous month. Treemaps have also been used to visualize changes in the whole stock market in a similar way (sectors, companies, market value, with color showing value changes).

The treemap layout algorithm takes a tree (with a quantitative attribute) and a rectangular space and tiles the tree's nodes into the space. Variants of the algorithm try to make the tiles more square (to avoid long skinny tiles whose areas are harder to compare) or maintain ordering relationships among siblings.



TableLens was a research project that combines the table representation with visualization. It displays thousands of rows of a table using only one pixel (or less) per row – converting quantitative data into a bar, and nominal data into position and color. Just like a typical table, you can click on a column heading to sort by that column, which in turn reveals relationships between that column and multiple other columns at the same time. In this display of data about homes for sale, sorting by price shows a clear correlation between price and bedrooms, bathrooms, and square footage – and correlations in cities, zipcodes, and realtors as well. This is a good demonstration of how packing lots of variables onto the screen at once can make interesting facts pop out.

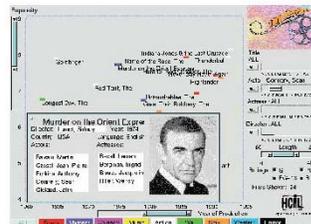
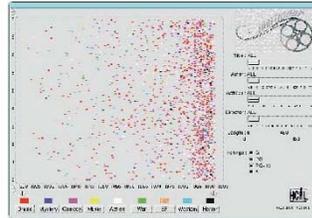


Tag clouds are another space-filling visualization, where the label of the data item is used to represent it, and visual variables of size and hue or value encode other information. In this example, size encodes the country’s population, and hue is just used to separate adjacent words (it doesn’t actually carry data).



## Shneiderman Mantra

- A good info viz provides three things:
  - Overview
  - Zoom & filter
  - Details on demand



© Human-Computer Interaction Laboratory,  
University of Maryland. All rights reserved.

Spring 2011

6.813/6.831 User Interface Design and Implementation

22

Most of our discussion to this point hasn't paid much attention to interaction, just focused on mapping data variables to visual variables. But in a graphical user interface, the user can interact with and change the visualization. Ben Shneiderman (a professor at University of Maryland who did seminal work in infoviz) has a good mantra for the design of an interactive visualization. "Overview, zoom & filter, details on demand." If you include those three elements, you're on the right track. Provide an **overview** that lets the user see all the data, get a big picture view of what's where, and see important relationships at a glance. The scatterplot display in Film Finder (which was developed by Shneiderman and Chris Ahlberg) is exactly that overview.

Second, provide the ability to **zoom and filter** the visualization, so that the user can reduce the (probably overwhelming) overview down to a subset of the data that they want to study more carefully. Good interaction means direct manipulation, so zooming might be done directly on the scatter plot, and filtering in a query interface alongside. In Film Finder, once you zoom or filter to few enough points, you start to see more detail, such as actual movie titles.

Third, provide more **details on demand**. Hovering over a point displays its title. Clicking on a point brings up a more detailed, probably more textual representation of a data item.

## Infoviz Toolkits

- Flot, Flotr
  - Javascript and <canvas>
  - conventional viz
- Protovis
  - Javascript & SVG
  - very powerful and flexible visualizations: treemaps, stacked time series, etc.
  - can reproduce Napoleon's March and Nightingale's Rose!

Here are a few toolkits (for HTML/Javascript) that provide some of the visualizations we've discussed in a reusable library.

## Summary

- Infoviz supports thinking and communicating
- Map data variables to visual variables
- Use space wisely
- Overview, zoom & filter, details on demand

MIT OpenCourseWare  
<http://ocw.mit.edu>

6.831 / 6.813 User Interface Design and Implementation  
Spring 2011

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.