

InfoVis Systems & Toolkits



CS 7450 - Information Visualization
September 14, 2015
John Stasko

Background



- In previous classes, we have examined different techniques for presenting multivariate data
 - We'll continue to show more later too
- Today we look at systems that implement these ideas and provide some of their own new visualization techniques

Agenda



- Toolkits that can be used to build systems
 - D3, Processing, ...
- Systems providing a view or views
 - Improvise, Many Eyes, Polaris, ...
- Commercial systems (to come our next class)
 - Spotfire, InfoZoom, Tableau, QlikView...

Toolkits & Infrastructures



- Set of components or capabilities that allow others to put together visualization systems
- Growing trend

Toolkit Design



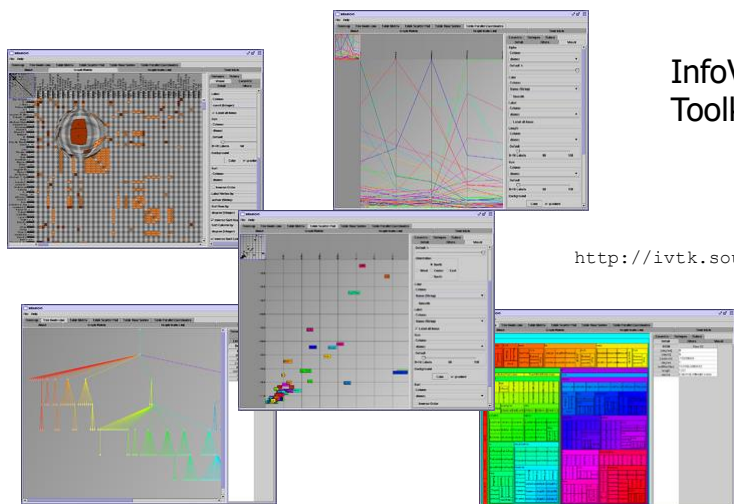
- What would you include in a toolkit like this if you designed it?

Fall 2015

CS 7450

5

History Lesson



InfoVis Toolkit

<http://ivtk.sourceforge.net>

Fekete
InfoVis '04

Fall 2015

CS 7450

6

Heer++ Series



Series of toolkits from Jeff Heer and his research group

Prefuse



Java2D

Heer et al, CHI '05

Fall 2015

CS 7450

7

Heer++ Series



Series of toolkits from Jeff Heer and his research group

Prefuse



What happened???

Web!!!

Java2D

Heer et al, CHI '05

Fall 2015

CS 7450

8

Heer++ Series



Series of toolkits from Jeff Heer and his research group

Prefuse

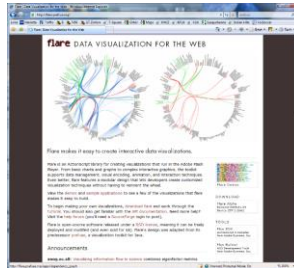


Java2D

Heer et al, CHI '05

Fall 2015

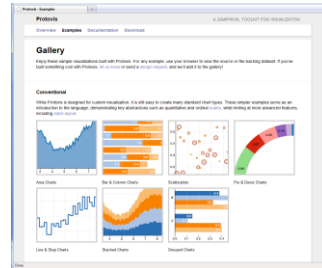
Flare



ActionScript & Flash

CS 7450

Protovis



Declarative spec.

Bostock & Heer, *TVCG* (InfoVis) '09

Heer & Bostock, *TVCG* (InfoVis) '10

9

D3: Data-Driven Documents



- Newest entry in the Heer-Bostock line of toolkits
- “Not just an infovis toolkit”
- Javascript-based
- Very similar to Protovis...
 - Except makes use explicitly of web standards such as Scalable Vector Graphics (SVG) rather than a proprietary “marks” graphics set

Fall 2015

CS 7450

10

D3 Design Pattern



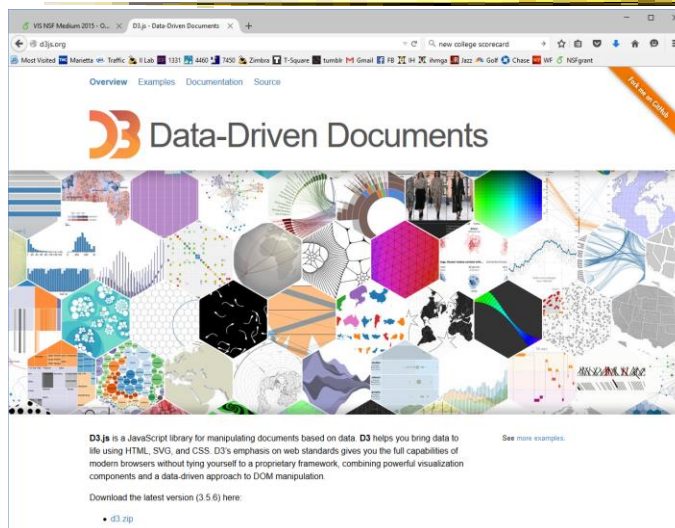
- Declarative Syntax like Protovis
- Creating/Modifying selections of the HTML DOM
- “An elegant for-loop with a bunch of useful helper functions”
- Excellent support for changing data
 - Taking advantage of CSS3 Transformations and Transitions
- Integrates seamlessly into any webpage

Fall 2015

CS 7450

11

D3 Website



Fall 2015

CS 7450

12

D3 Support



- Active community online
 - <https://github.com/mbostock/d3/wiki>
 - Including Mike Bostock often answering questions
- Tutorial from Chad in our last class

Fall 2015

CS 7450

13

Vega

<https://trifacta.github.io/vega/>



The screenshot shows a web browser window displaying the Vega website. The browser's address bar shows the URL <https://trifacta.github.io/vega/>. The website content includes the Vega logo, a list of example visualizations (bar chart, scatter plot, treemap, map), and text describing Vega as a visualization grammar. It mentions that Vega can describe data visualizations in JSON format and generate interactive views using HTML5 Canvas or SVG. Links are provided for the tutorial, documentation, discussion, and Vega Editor. A red banner in the top right corner of the website says "For more on GitHub".

Fall 2015

CS 7450

14

Characteristics



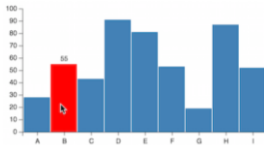
- Declarative visualization grammar like D3
- No CSS, DOM, etc., though
- Describe visualizations in JSON format
- Generate interactive views via HTML5 Canvas or SVG

Fall 2015

CS 7450

15

Example Code



```
{
  "width": 400,
  "height": 200,
  "padding": {"top": 10, "left": 30, "bottom": 20, "right": 10},
  "data": [
    {
      "name": "table",
      "values": [
        {"category": "A", "amount": 28},
        {"category": "B", "amount": 55},
        {"category": "C", "amount": 43},
        {"category": "D", "amount": 91},
        {"category": "E", "amount": 81},
        {"category": "F", "amount": 53},
        {"category": "G", "amount": 19},
        {"category": "H", "amount": 87},
        {"category": "I", "amount": 52}
      ]
    }
  ],
  "signals": [
    {
      "name": "tooltip",
      "init": {},
      "streams": [
        {"type": "rect:mouseover", "expr": "datum"},
        {"type": "rect:mouseout", "expr": "{}"}
      ]
    }
  ]
}
```

and more...

Fall 2015

CS 7450

16

Premise



- Not really for humans to write
- Intended for tools to generate & exchange it

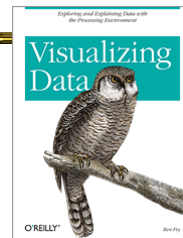
Fall 2015

CS 7450

17

Processing

<http://processing.org>



- Java based
- Unlike protovis & D3, not specifically designed for InfoVis
 - Data Reader? Layout algorithm?
 - But can definitely be used to build visualizations!
- Well documented, lots of tutorials with contributions from many people and even books

Ben Fry

Fall 2015

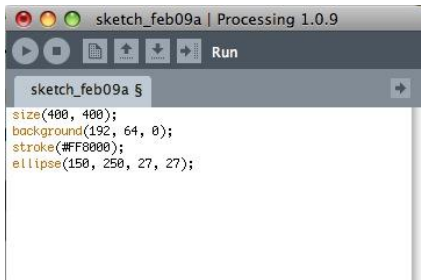
CS 7450

18

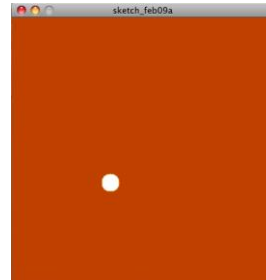
Processing: the idea



- Programming as scripting
 - PDE: processing development environment
 - A program is called a *sketch*
 - written as a list of statements

A screenshot of the Processing IDE window titled 'sketch_feb09a | Processing 1.0.9'. The window has a toolbar with icons for play, stop, refresh, and run. Below the toolbar is a text area containing the following code:

```
sketch_feb09a $  
size(400, 400);  
background(192, 64, 0);  
stroke(#FF8000);  
ellipse(150, 250, 27, 27);
```



Fall 2015

CS 7450

19

Processing: Useful Functions



```
void setup() {  
  //your own code here  
}
```

```
void draw() {  
  //your own code here  
}
```

- These are built-in functions that are called automatically.
 - The `setup()` block runs once.
 - The `draw()` block runs repeatedly: good for animation

Fall 2015

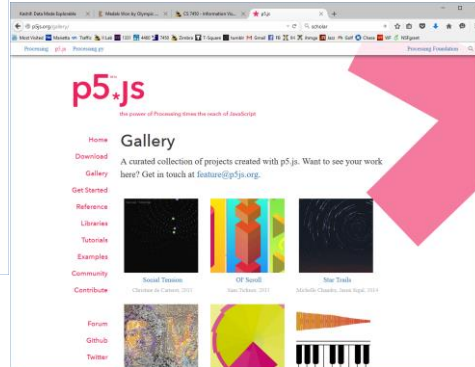
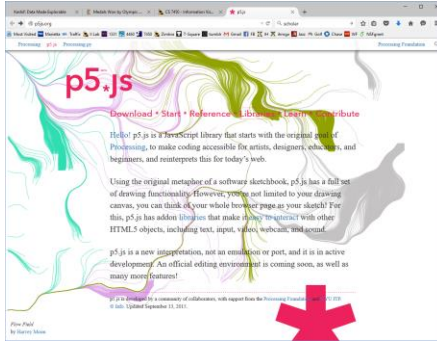
CS 7450

20

P5.js



- Interpretation of Processing in Javascript

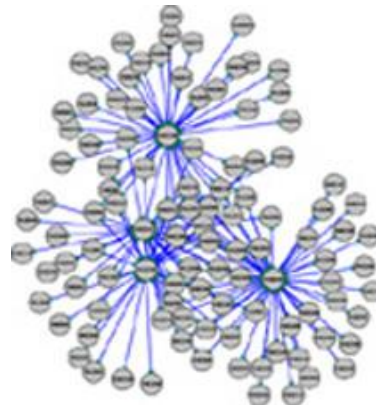


Fall 2015

CS 7450

21

Piccolo



Fall 2015

CS 7450

22

Characteristics



- Graphics toolkit with very nice built-in zooming and panning support
- Useful for implementing infovis too
- Will discuss more later in course...

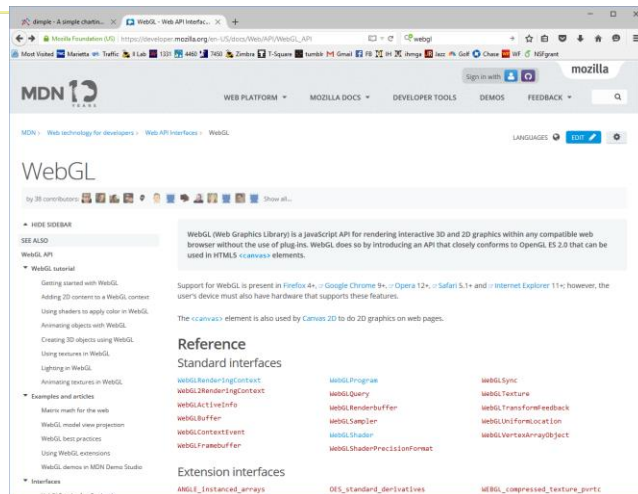
Fall 2015

CS 7450

23

https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API

WebGL



When you need 3D or really sophisticated graphics

Fall 2015

CS 7450

24



More toolkits

Fall 2015

CS 7450

25

<http://aperturejs.com/>

Aperture.js



Aperture JS agile visual analytics for big data

Aperture is a powerful, agile, and extensible visualization framework library project designed to produce tailored, best practice, scalable visualizations for analysts and decision makers in any common web browser. Aperture is an open source, standards based JavaScript library with supporting web services.

Declarative Grammar of Richly Layered Visual Forms

Aperture employs a unique and unified layer based approach to visualization assembly, enabling richer, more powerful combinations of visual forms than a standard widget based approach with many frames. Aperture's visual mapping and filter API provides adaptability to any data schema and transformation of data into succinct interactive visual forms with minimal effort.

Highly Interoperable using Web 2.0, SOA and GIS Standards

Aperture uses a non-proprietary Web 2.0, Service Oriented Architecture (SOA) approach with SVG or VML (IE7, 8) rendering and wide support for GIS standards, for greatest interoperability. Aperture *vizlets* (visualization forms) are easily embedded in web-based client frameworks such as the *Ozone Widget Framework* (OWF), or in any simple web page, where open JavaScript APIs provide full interoperability with other DOM elements.

Extensible and Open Source

The Aperture framework and API are designed for ease of extension, allowing a broad community to leverage and extend capabilities in creative ways. Aperture is under ongoing development and is freely available for download under The MIT License (MIT) open source licensing. Unlike GNU General Public License (GPL), MIT freely permits distribution of derivative work under proprietary license, without restricting the release of source code.

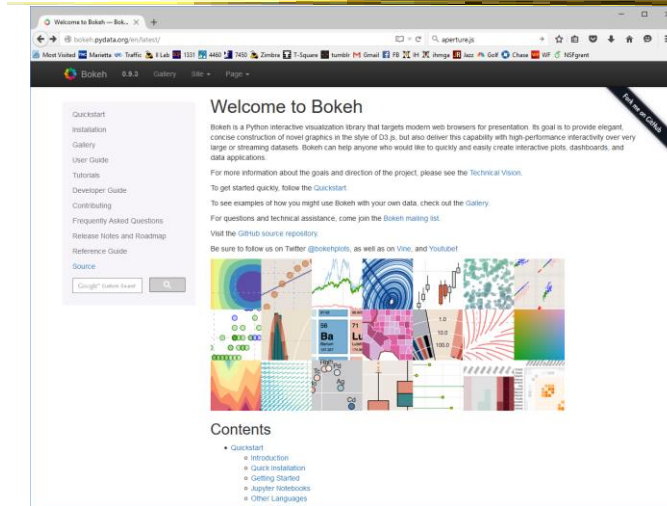
Fall 2015

CS 7450

26

Bokeh

<http://bokeh.pydata.org/en/latest/>



Python

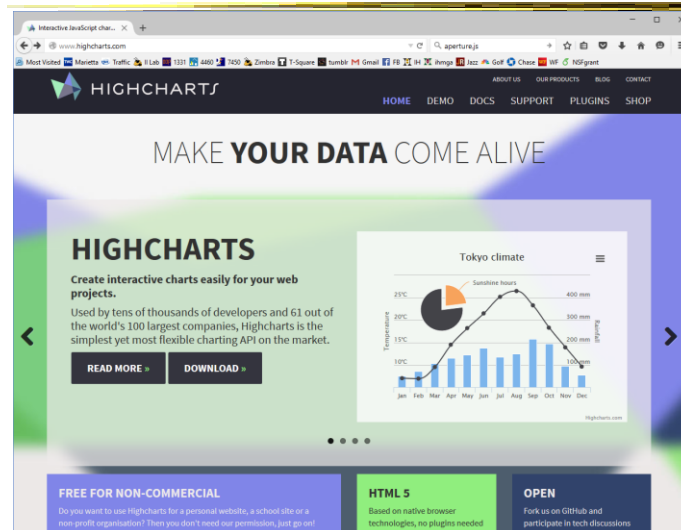
Fall 2015

CS 7450

27

HighCharts

<http://www.highcharts.com/>



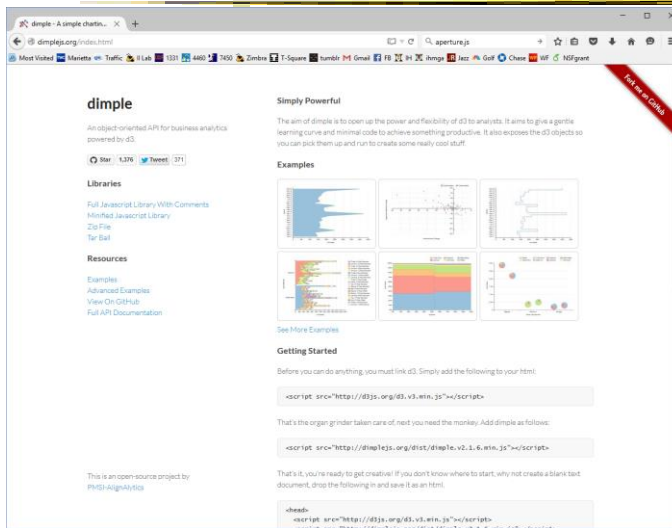
Fall 2015

CS 7450

28

dimple

<http://dimplejs.org>



On top of D3

Fall 2015

CS 7450

29

JavaScript InfoVis Toolkit



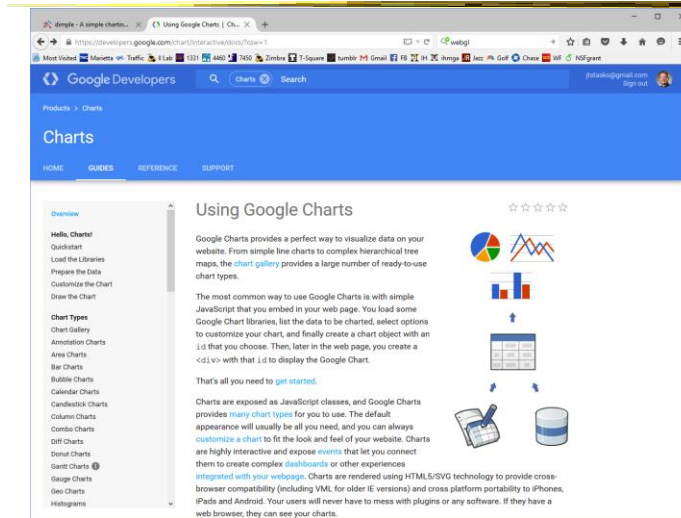
<http://philogb.github.io/jit/>

Fall 2015

CS 7450

30

Google Chart Tools



Fall 2015

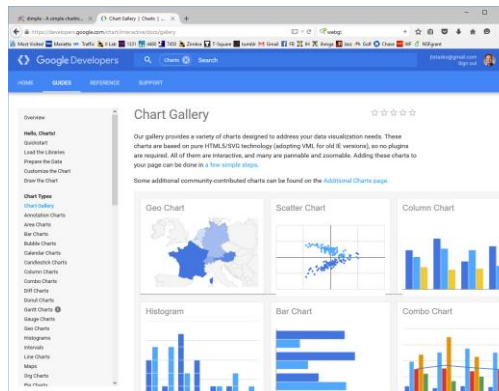
CS 7450

31

Characteristics



- Javascript-based
- Gallery of contributed code segments
- Visualizations are interactive
- Evolving API



Fall 2015

CS 7450

32

Reflection



- What would you seek in a good infovis system or toolkit?

Toolkit Tradeoffs



- InfoVis-focused
 - Many fundamental techniques built-in
 - Can be faster to get something going
 - Often more difficult to implement something “different”
 - Documentation?
- Generic graphics
 - More flexible
 - Can customize better
 - Big learning curve
 - Doc is often better
 - Can take a long time to (re)implement basic techniques

Design Patterns



- Heer used his experience with prefuse etc to describe common design patterns for infovis
 - Reference model
 - Data column
 - Cascaded table
 - Relational graph
 - Proxy tuple
 - Expression
 - Scheduler
 - Operator
 - Renderer
 - Production rule
 - Camera
 - Dynamic query binding

Heer & Agrawala
TVCG (InfoVis) '06

Fall 2015

CS 7450

35

Writing Code is Hard



- Why not just show what you want the visualization to look like?
 - What's the challenge?

Fall 2015

CS 7450

36

Lyra



- Interactive vis builder tool without needing to program
- Graphical “marks” are bound to data fields
- User shows what vis is to look like, the mapping from data
- Generates code (Vega) that can be run on the web

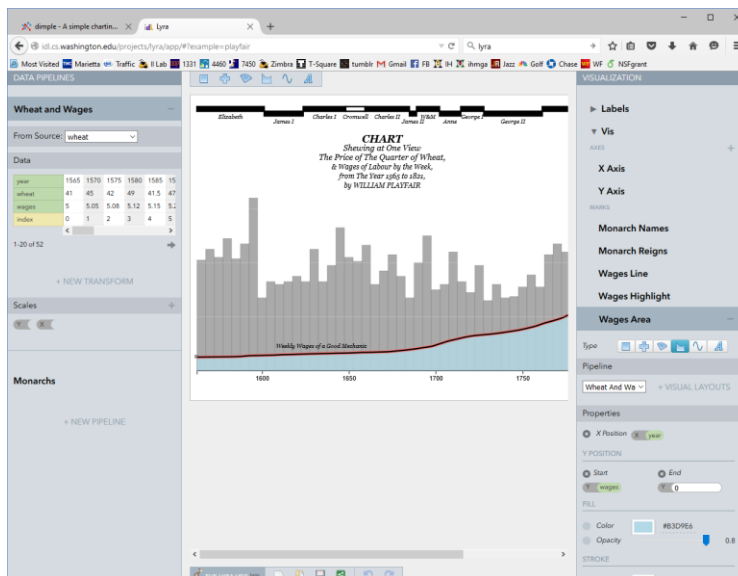
Satyanarayan & Heer
Computer Graphics Forum (EuroVis) '14

Fall 2015

CS 7450

37

<http://idl.cs.washington.edu/projects/lyra/>



Video

Fall 2015

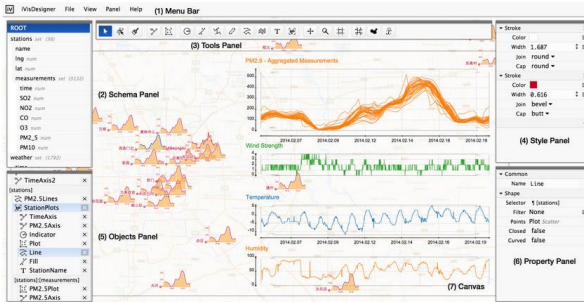
CS 7450

38

iVisDesigner



- Interactively create mappings from data elements to graphical elements
- Has scatterplot, timeline, graph templates



Video

Ren, Hollerer, & Yuan
TVCG (InfoVis) '14

Fall 2015

CS 7450

39



Journalism-driven systems

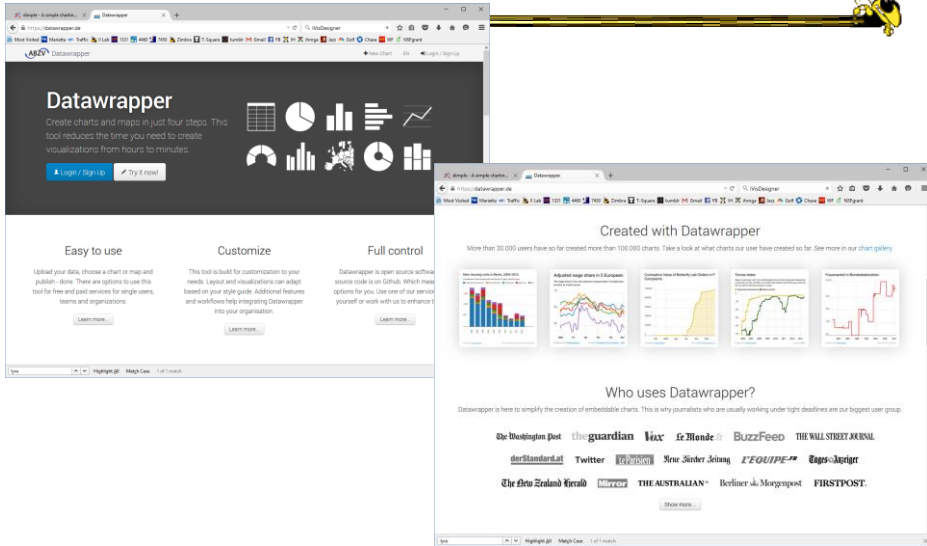
Fall 2015

CS 7450

40

<https://datavrapper.de/>

Datawrapper



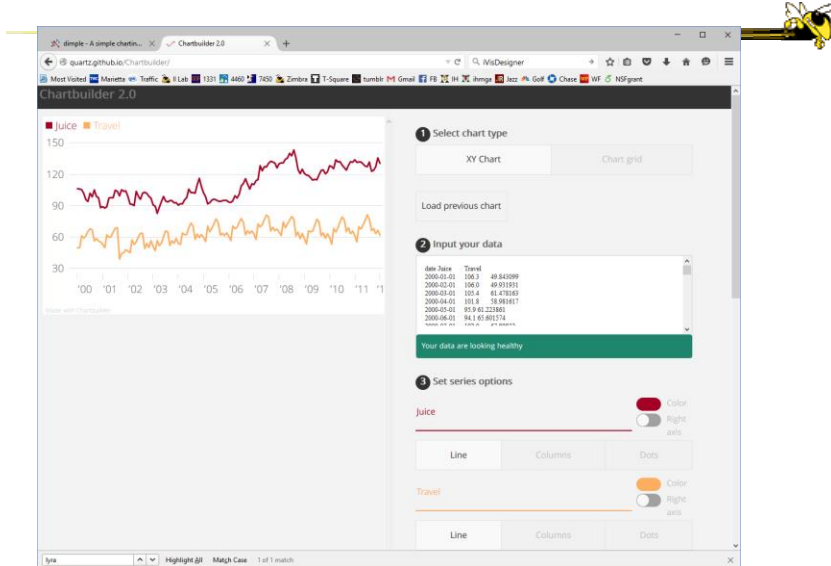
Fall 2015

CS 7450

41

<http://quartz.github.io/Chartbuilder/>

Chartbuilder



Fall 2015

CS 7450

42

Systems/Tools



- Primarily examining academic/research systems that provide preexisting views
 - Commercial systems next time

Fall 2015

CS 7450

43

<http://www.graphics.stanford.edu/projects//polaris/>

Polaris

<http://www.tableausoftware.com>



Database Schema: The user drags fields from the database schema to shelves to define the visual specification.

Layer Tabs: Each layer has its own tab; different transformations and mappings can be specified for each layer.

Axis Shelves: The fields placed here determine the structure of the table and the types of graphs in each table pane.

Context Menu: The context menu provides access to the data transformation and interaction capabilities of Polaris such as sorting, filtering, and aggregation.

Layer Shelf: The fields placed here determine how records are partitioned into layers.

Grouping and Sorting Shelves: The fields placed here determine how records are grouped and sorted within the table panes.

Mark Pulldown: Relations in each pane are mapped to marks of the selected type.

Retinal Property Shelves: The fields placed here determine how data is encoded in the retinal properties of the marks.

Legends: Legends enable the user to see and modify the mappings from data to retinal properties.

Video

Stolte et al
TVCG '02

Fall 2015

CS 7450

44

Basis



- Relational databases
- Pivot tables from spreadsheets
- N-dimensional data cubes

- Analytic approach is fundamental
- Provides visual representations of these concepts

Visualization



- Table of data (rows, columns)
- Each axis may have nested dimensions
- Table entry is a pane, and has visual marks to represent data

- Analyst drags and drops fields from database schema onto shelves of display
- Much interaction supported

Graphics



- Formal table algebra provided to describe data
- Visual mappings established from data types to appropriate (good) markings and encodings

Impact: Became Tableau

Fall 2015

CS 7450

47

Many Eyes

Viégas et al
TVCG (InfoVis) '07



- InfoVis on the web
- Website developed from IBM's infovis group
- Motivating infovis challenges:
 - Difficulty of creating new visualizations
 - How do you discuss the visualizations?
- Project goals:
 - Enable end-user creation of visualizations
 - Foster large-scale collaborative usage

Fall 2015

CS 7450

48

Features

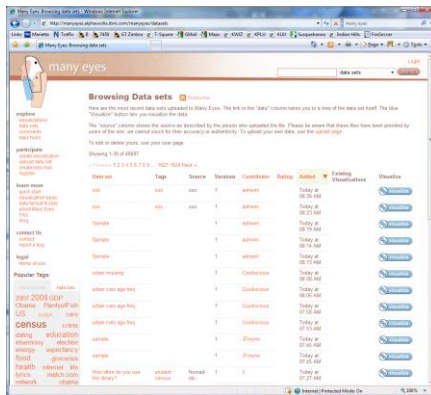


- Provides data upload capabilities and choice from library of visualizations
- Includes
 - Gallery of recently uploaded visualizations for browsing
 - Chosen highlighted visualizations
 - Attached discussion forums for each vis

Fall 2015

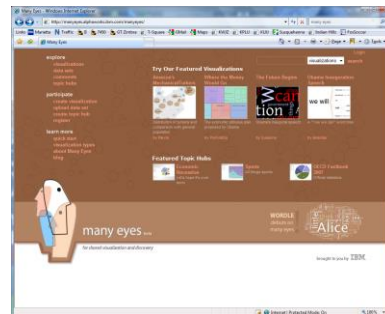
CS 7450

49



Recent gallery

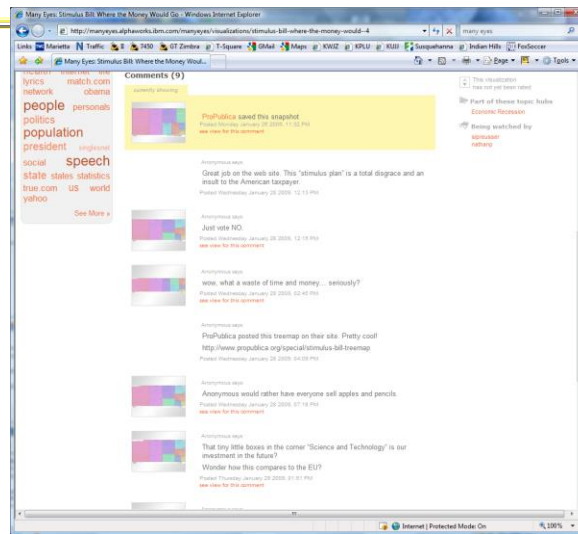
Featured visualizations



Fall 2015

CS 7450

50



Discussion
forums

Fall 2015

CS 7450

51

Data

- Users upload their own data sets
 - All become public
- Format: table or unstructured text
 - Metadata allowed
- Immutable once uploaded

Fall 2015

CS 7450

52

Visualizations



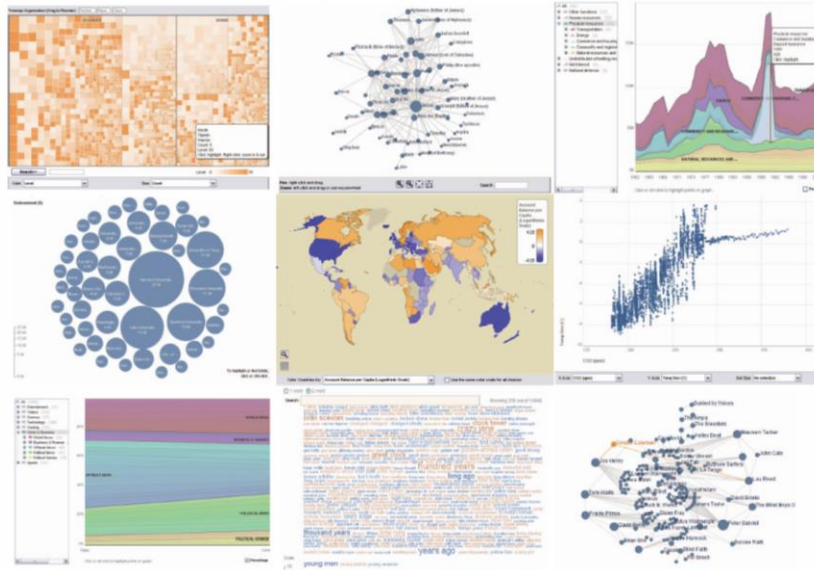
- Preloaded visualization types
 - Has grown over time
- User chooses one to combine with their data
- Provides named, typed slots that the user maps particular pieces of data to
 - System makes some reasonable guesses too

Fall 2015

CS 7450

53

Originals

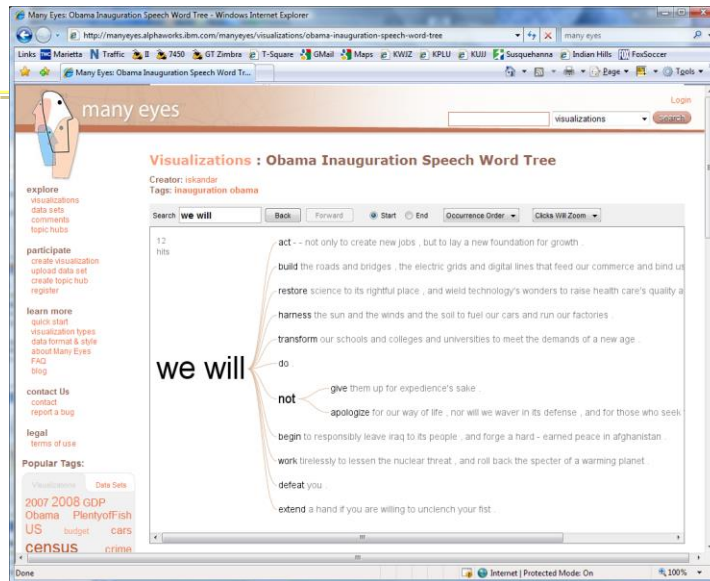


Fall 2015

CS 7450

54

WordTree



Fall 2015

CS 7450

55

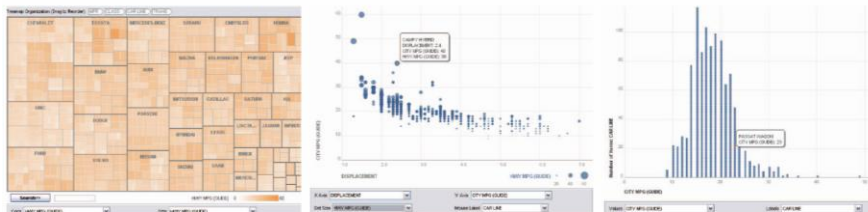


Fig. 3. Three user generated visualizations offering different perspectives on the same dataset on car fuel economy. The grey areas on the top and bottom are automatically generated by the application and allow the user to browse through different dimensions in the data.

Allows the user to control the mapping from data to image

Fall 2015

CS 7450

56

Social Aspects



- Users identified by login ID
- Can leave comments about different visualizations
- Can take snapshot of visualization state
 - Unique URL
- “Blog this” button

Fall 2015

CS 7450

57

Evaluation



- Quantitative, objective
- 1895 posts of March '07
- Wide variety of topics of visualizations and motivations for creating visualizations
- Does seem to be fostering discussion

Viégas et al
HICSS '08

Fall 2015

CS 7450

58

Evaluation



- Qualitative, subjective
- In-depth interviews with some ME users
- Visualizations used largely for communication and collaboration (not necessarily analysis)
 - Privacy and audience management a concern
- Highlights a number of interesting, non-expected uses of the technology

Danis et al
CHI '08

Fall 2015

CS 7450

59

https://www-304.ibm.com/connections/blogs/predictiveanalytics/entry/six_years_of_many_eyes_a_personal_retrospective?lang=en_us

Status



- Shut down by IBM on June 12, 2015



Fall 2015

CS 7450

60

Thoughts?



- What do you think of the design choices they made?

Sense.us



- Related, follow-up to Many Eyes
- Discussion and visualization of US Census data
- Go beyond Many Eyes in terms of annotation, collaboration, and discussion

Components



- Doubly-linked discussion
 - Can go from visualization to threaded discussion items or vice-versa
- Graphical annotation
 - Simple graphics editor and comments (like transparent layer)
- Bookmark trail
 - Small strip of views
- Comment listings and social navigation
 - Searchable, sortable indices and links

Fall 2015

CS 7450

63

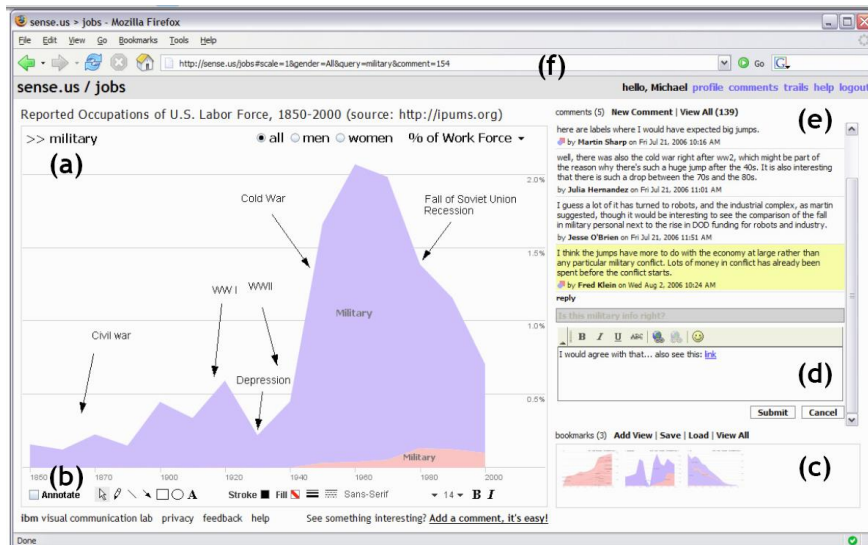


Figure 1. The sense.us collaborative visualization system. (a) An interactive visualization applet, with a graphical annotation for the currently selected comment. The visualization is a stacked time-series visualization of the U.S. labor force, broken down by gender. Here the percentage of the work force in military jobs is shown. (b) A set of graphical annotation tools. (c) A bookmark trail of saved views. (d) Text-entry field for adding comments. Bookmarks can be dragged onto the text field to add a link to that view in the comment. (e) Threaded comments attached to the current view. (f) URL for the current state of the application. The URL is updated automatically as the visualization state changes.

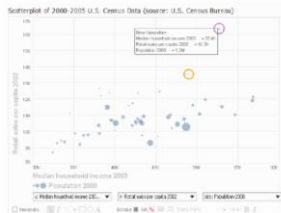


Figure 5. Scatterplot of U.S. states showing median household income (x-axis) vs. retail sales per capita (y-axis). New Hampshire and Delaware have the highest retail sales.



Figure 6. Visualization of the number of teachers. Annotations indicate the start of compulsory education and the rise of teachers in the post World War II era.

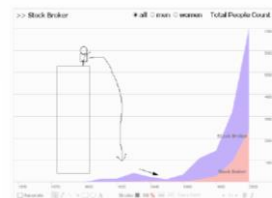


Figure 7. Annotated view of stock brokers. The attached comment reads "Great depression 'killed' a lot of brokers".

Sample annotations

Fall 2015

CS 7450

65

Statwing

<https://www.statwing.com/>

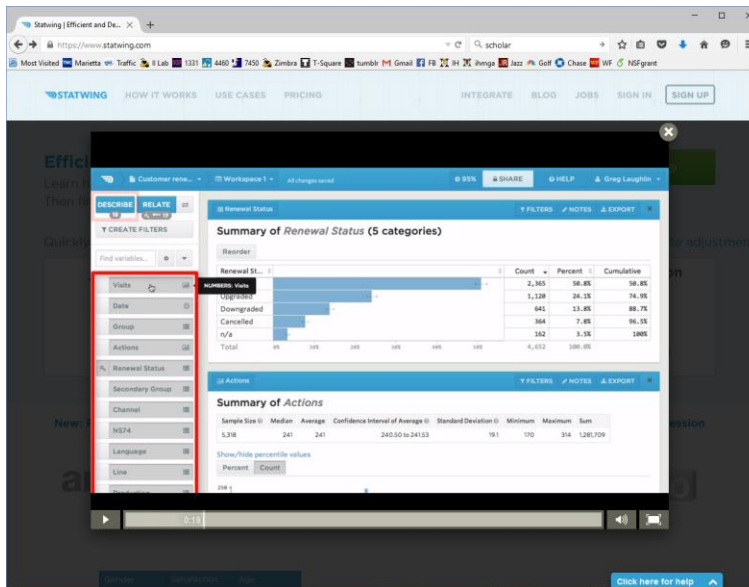


- Data analysis tool
- Upload spreadsheet or dataset, pick relationships you want to explore
- Basic graph types generated

Fall 2015

CS 7450

66



Demo video

Fall 2015

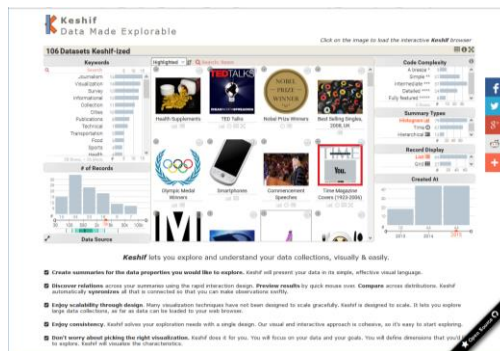
CS 7450

67

Keshif

<http://keshif.me/>

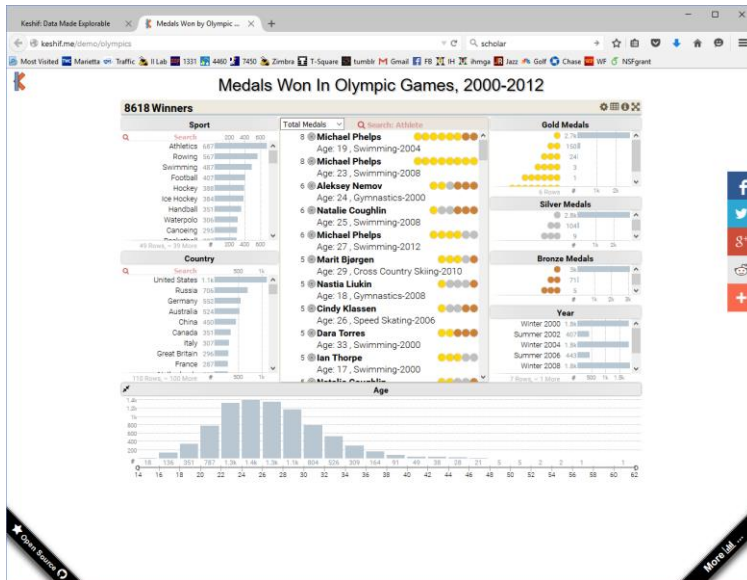
- Import data from spreadsheets and csv/text files
- Provides multiple coordinated bar charts and histograms
- Large set of examples on website



Fall 2015

CS 7450

68



Demo

Fall 2015

CS 7450

69

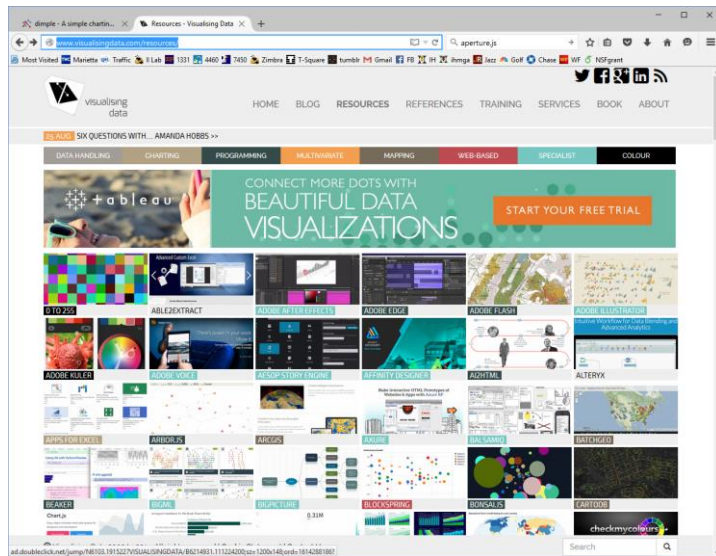
Some web collections

Fall 2015

CS 7450

70

<http://www.visualisingdata.com/resources/>



Fall 2015

CS 7450

71

<http://www.creativebloq.com/design-tools/data-visualization-712402>

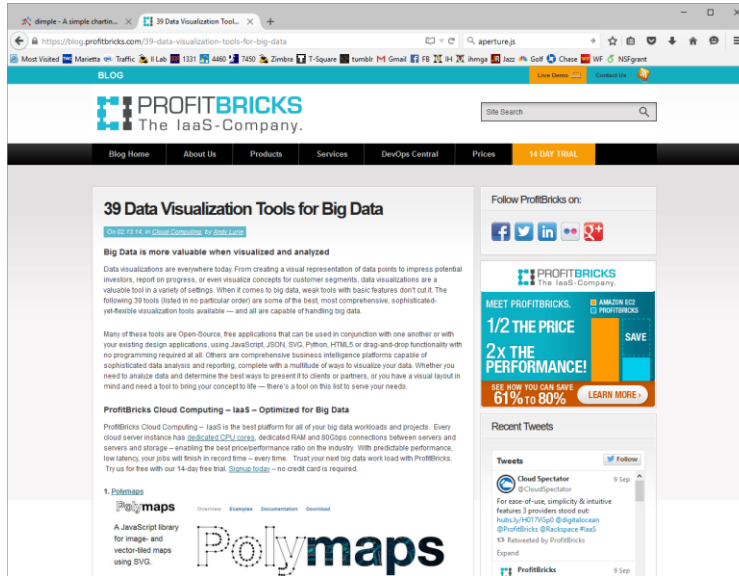


Fall 2015

CS 7450

72

https://blog.profitbricks.com/39-data-visualization-tools-for-big-data

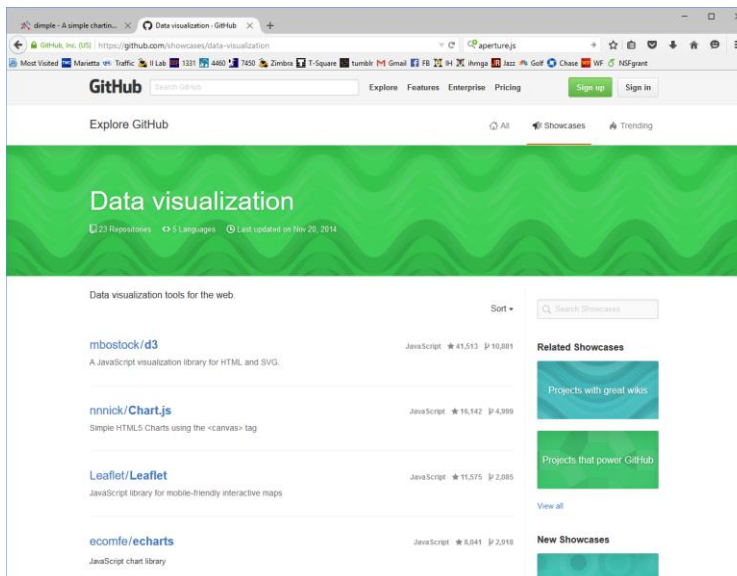


Fall 2015

CS 7450

73

https://github.com/showcases/data-visualization



Fall 2015

CS 7450

74

More Next Time...



- Demos of commercial infovis systems
 - Spotfire
 - Table Lens/Eureka
 - SeeIt
 - Qlikview
 - InfoZoom
 - Tableau

HW 2 Feedback



- Things we noticed
- Example solutions

Project Topics Feedback



- Will try to return your proposals soon
- Get to work – Poster session is a few weeks away

Fall 2015

CS 7450

77

Upcoming



- Commercial systems
 - Reading
Spenke & Beilken '00
- Interaction
 - Reading
Yi et al '07

Fall 2015

CS 7450

78