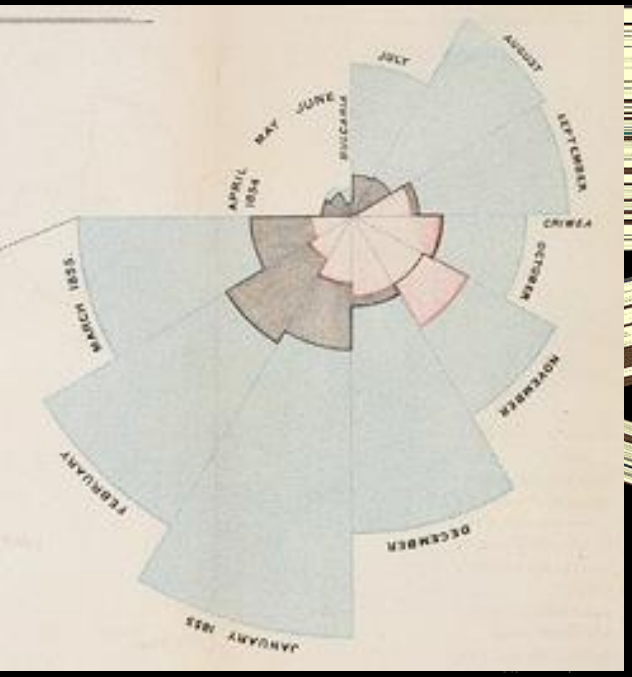
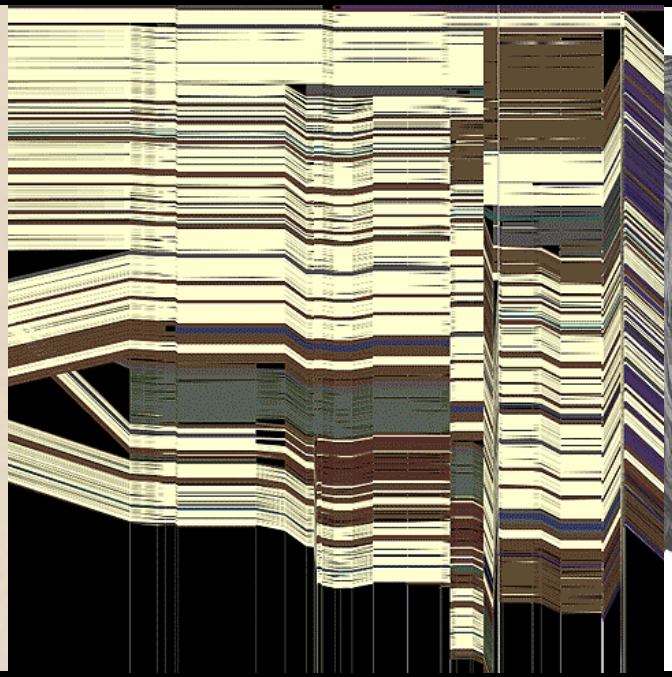
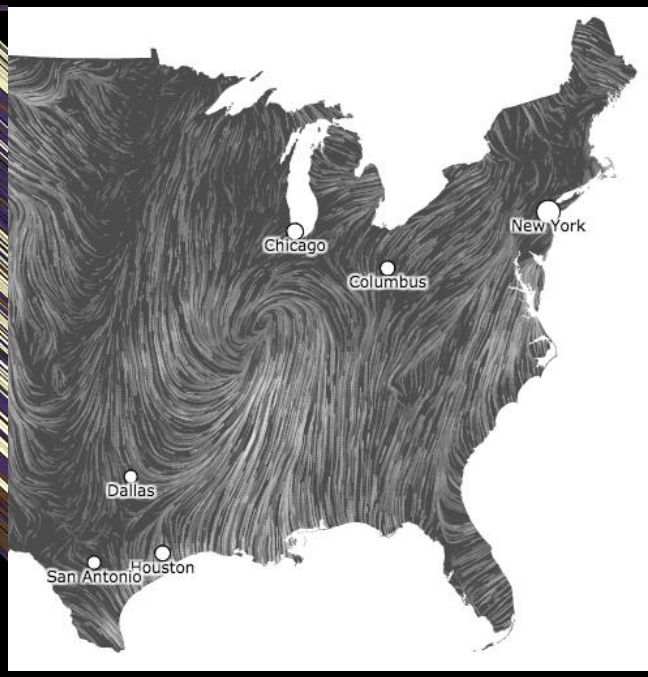
CSE 442 - Data Visualization

Visualization Tools







Leilani Battle University of Washington

Tutorial on Thursday

D3.js Deep Dive: Thursday 1/30 during lecture, Led by Lisa and Jiawen

- Be sure to read the D3, Part 1 notebook ahead of time. We'll work through Part 2 in class. Also read the JS/Observable primer if you're new to this!

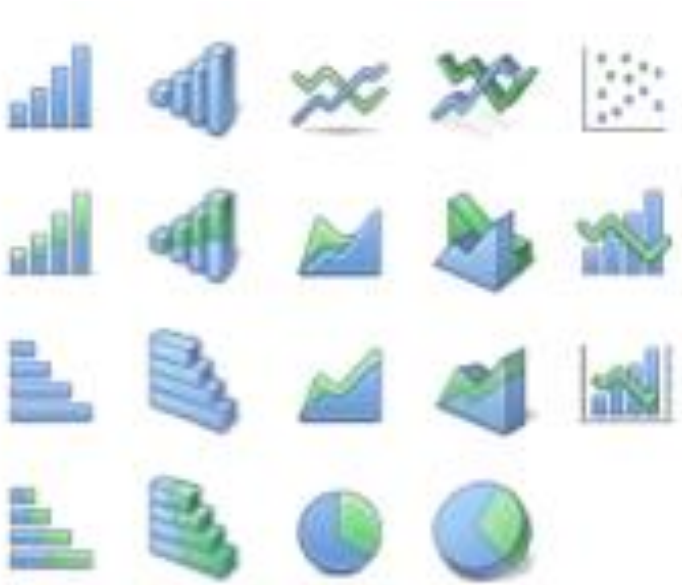
Learning Goals

There are many tools available to help people create visualizations.

What are the strengths and weaknesses of current tools?

What trade-offs must be considered when designing a tool for creating visualizations?

How do people create visualizations?



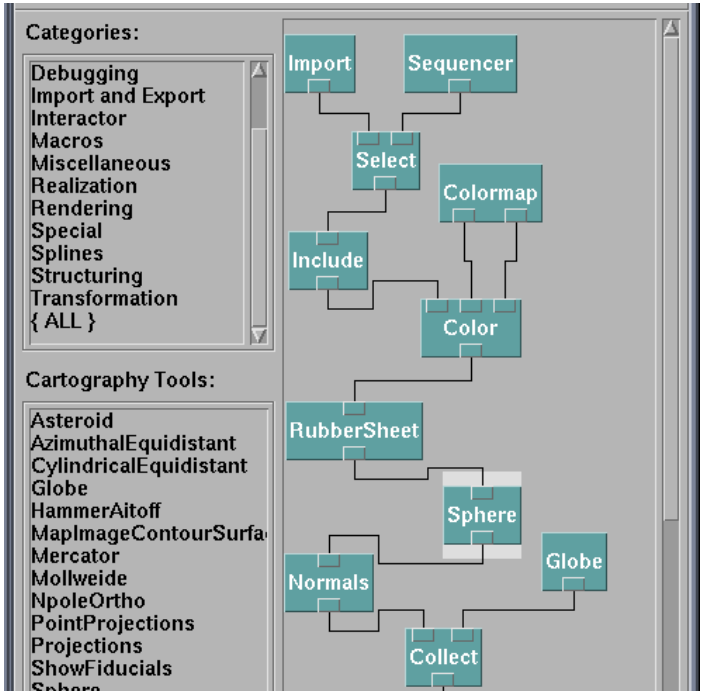


Chart Typology

Pick from a stock of templates

Easy-to-use but limited expressiveness

Prohibits novel designs, new data types

Component Architecture

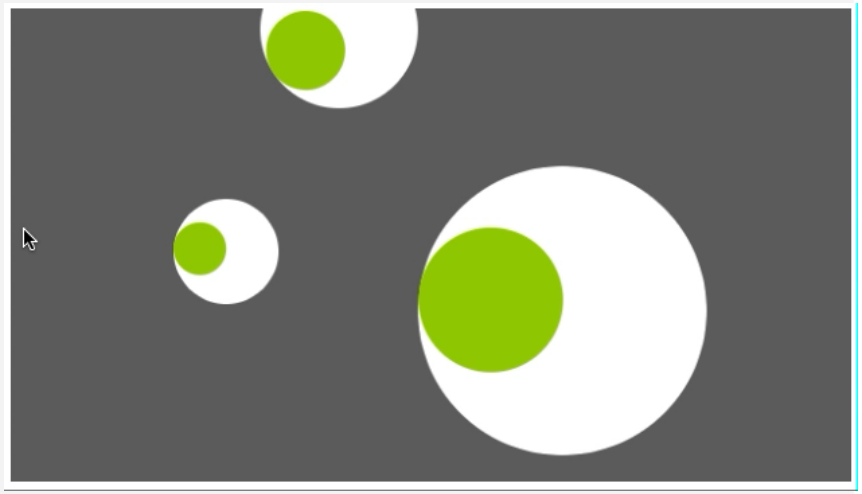
Permits more combinatorial possibilities

Novel views require new operators, which requires software engineering

Graphics APIs

Canvas, OpenGL, Processing

```
  Eye(int tx, int ty, int ts) {
    x = tx;
    y = ty;
    size = ts;
  }

  void update(int mx, int my) {
    angle = atan2(my-y, mx-x);
  }

  void display() {
    pushMatrix();
    translate(x, y);
    fill(255);
    ellipse(0, 0, size, size);
    rotate(angle);
    fill(153, 204, 0);
    ellipse(size/4, 0, size/2, size/2);
    popMatrix();
  }
}
```



http://processing.org


US Air Traffic, Aaron Koblin

Graphics APIs

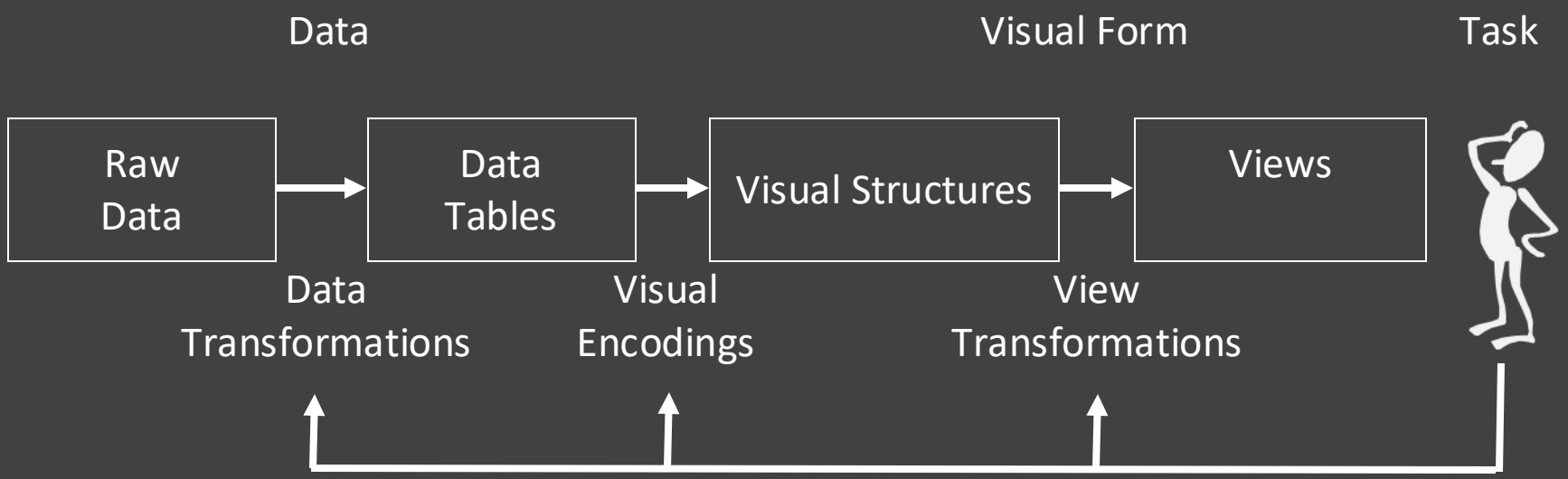
Canvas, OpenGL, Processing

Component Architectures

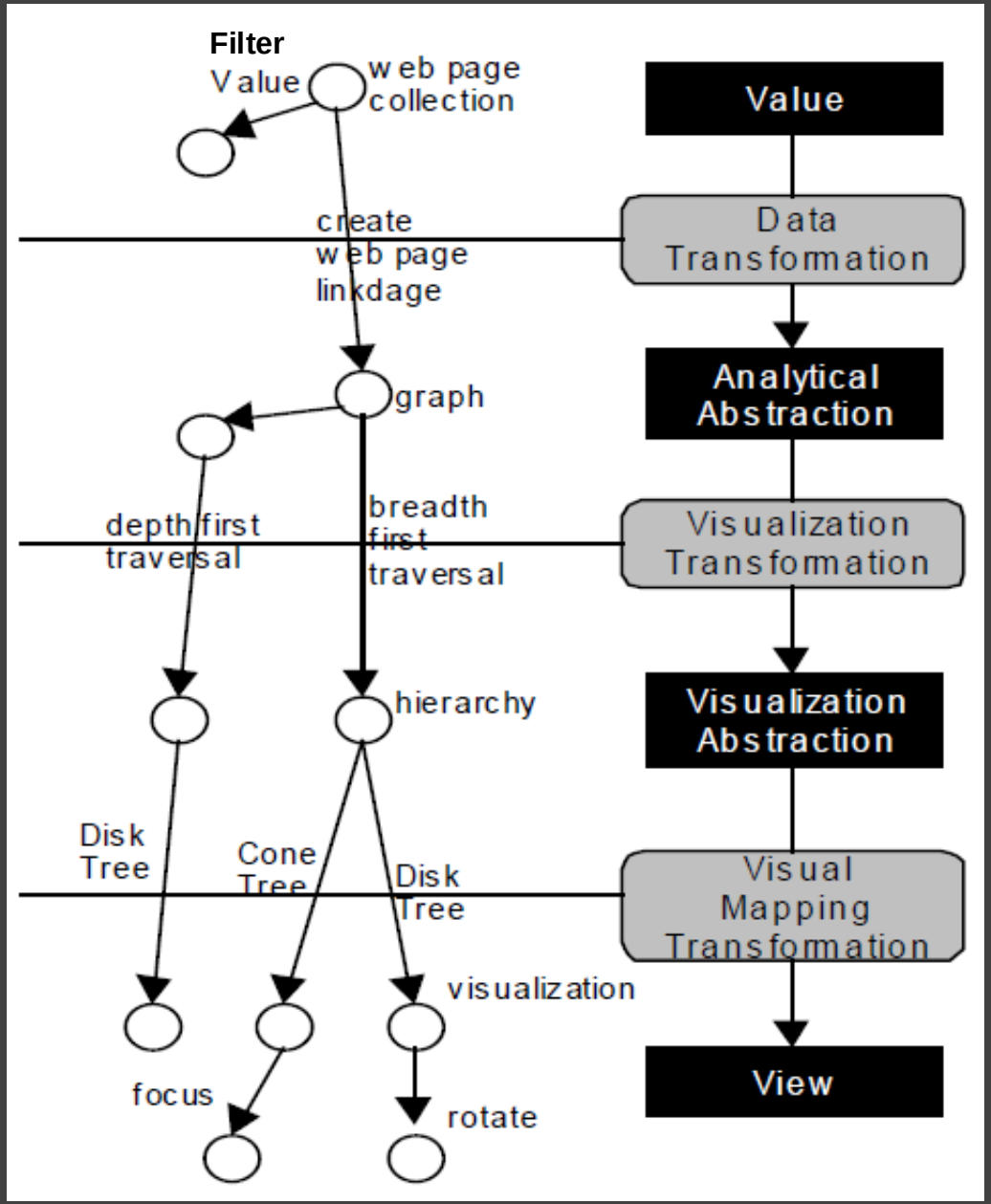
Prefuse, Flare, Improvise, VTK

Graphics APIs

Canvas, OpenGL, Processing


Data
Visual Form
Task
Raw Data
Data Tables
Visual Structures
Views
Data Transformations
Visual Encodings
View Transformations

Data State Model [Chi 98]



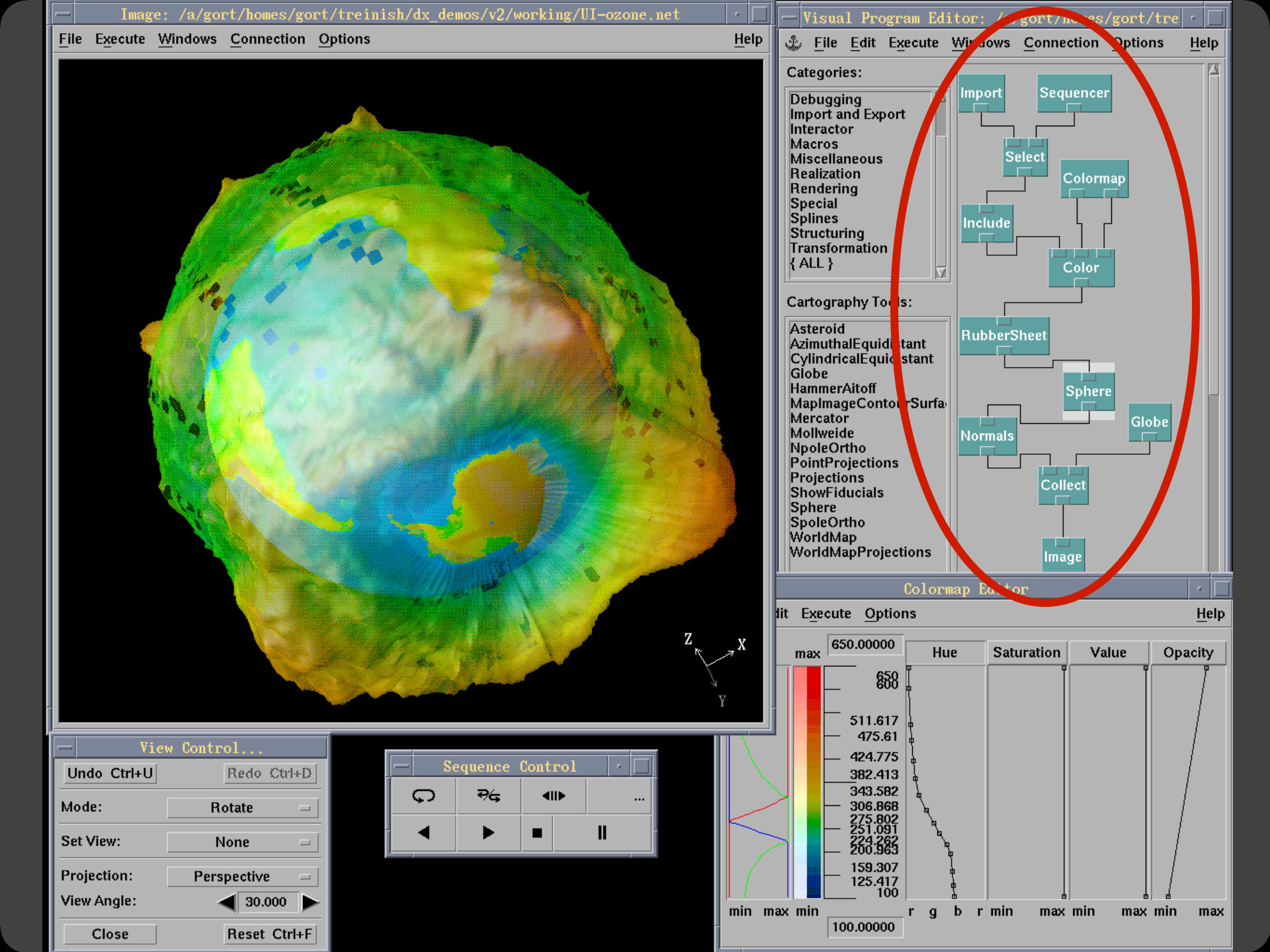
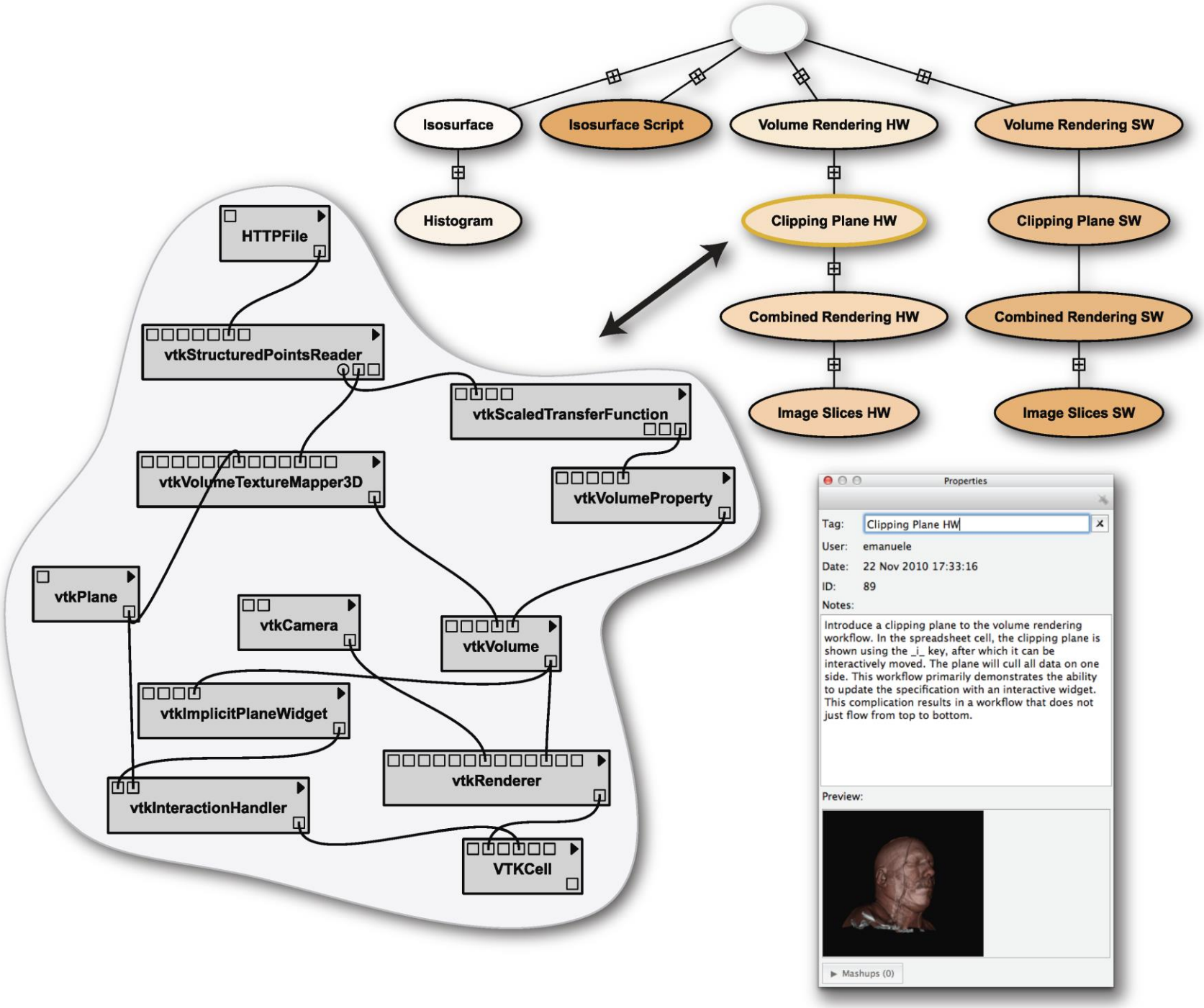


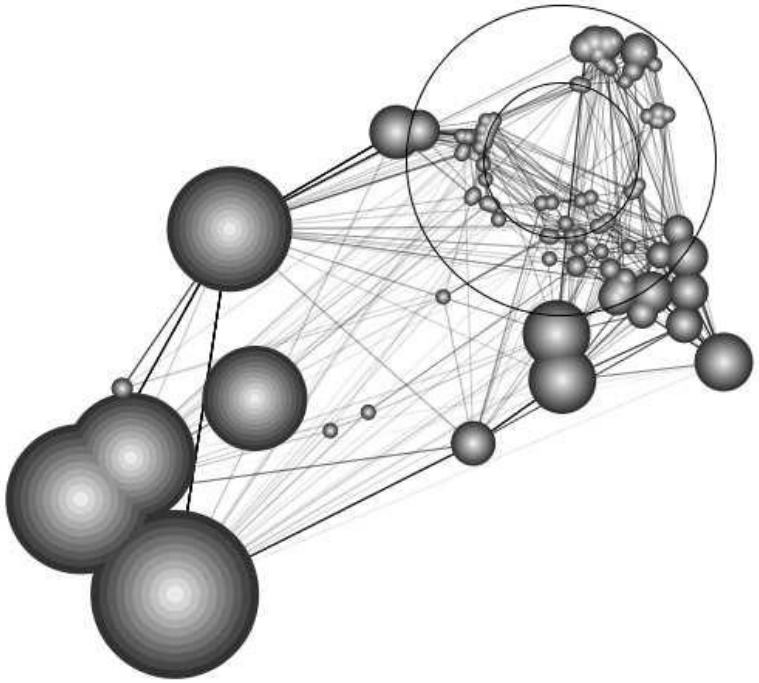
Image: /a/gort/homes/gort/treinish/dx_demos/v2/working/UI-ozone.net
File Execute Windows Connection Options Help
Visual Program Editor:
File Edit Execute Windows Connection Options Help
Categories:
Debugging
Import and Export
Interactor
Macros
Miscellaneous
Realization
Rendering
Special
Splines
Structuring
Transformation
{ ALL }
Cartography Tools:
Asteroid
AzimuthalEquidistant
CylindricalEquistant
Globe
HammerAitoff
MapImageContourSurface
Mercator
Mollweide
NpoleOrtho
PointProjections
Projections
ShowFiducials
Sphere
SpoleOrtho
WorldMap
WorldMapProjections
Import
Sequencer
Select
Colormap
Include
Color
RubberSheet
Sphere
Globe
Normals
Collect
Image
Z
X
Y
Colormap Editor
Edit Execute Options Help
max
650.00000
650
600
511.617
475.61
424.775
382.413
343.582
306.868
275.802
251.091
224.262
200.563
159.307
125.417
100
100.00000
Hue
Saturation
Value
Opacity
min max min r g b r min max min max min max
View Control...
Undo Ctrl+U
Redo Ctrl+D
Mode:
Rotate
Set View:
None
Projection:
Perspective
View Angle:
30.000
Close
Reset Ctrl+F
Sequence Control


Isosurface
Isosurface Script
Volume Rendering HW
Volume Rendering SW
Histogram
Clipping Plane HW
Clipping Plane SW
Combined Rendering HW
Combined Rendering SW
Image Slices HW
Image Slices SW
HTTPFile
vtkStructuredPointsReader
vtkScaledTransferFunction
vtkVolumeTextureMapper3D
vtkVolumeProperty
vtkPlane
vtkCamera
vtkVolume
vtkImplicitPlaneWidget
vtkRenderer
vtkInteractionHandler
VTKCell
Properties
Tag: Clipping Plane HW
User: emanuele
Date: 22 Nov 2010 17:33:16
ID: 89
Notes:
Introduce a clipping plane to the volume rendering workflow. In the spreadsheet cell, the clipping plane is shown using the _i_ key, after which it can be interactively moved. The plane will cull all data on one side. This workflow primarily demonstrates the ability to update the specification with an interactive widget. This complication results in a workflow that does not just flow from top to bottom.
Preview:
Mashups (0)

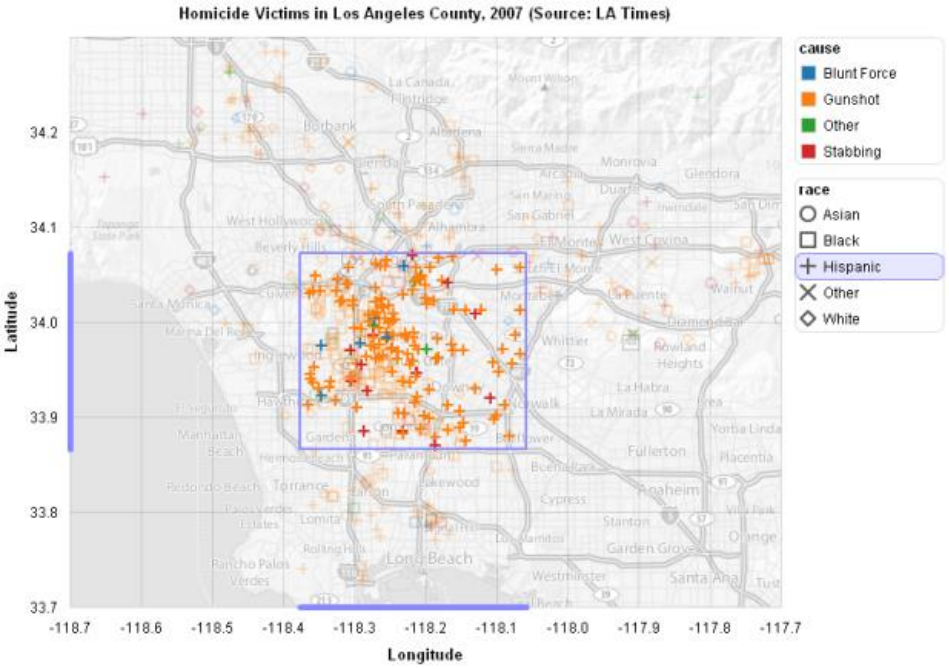
Prefuse & Flare

Operator-based toolkits for visualization design

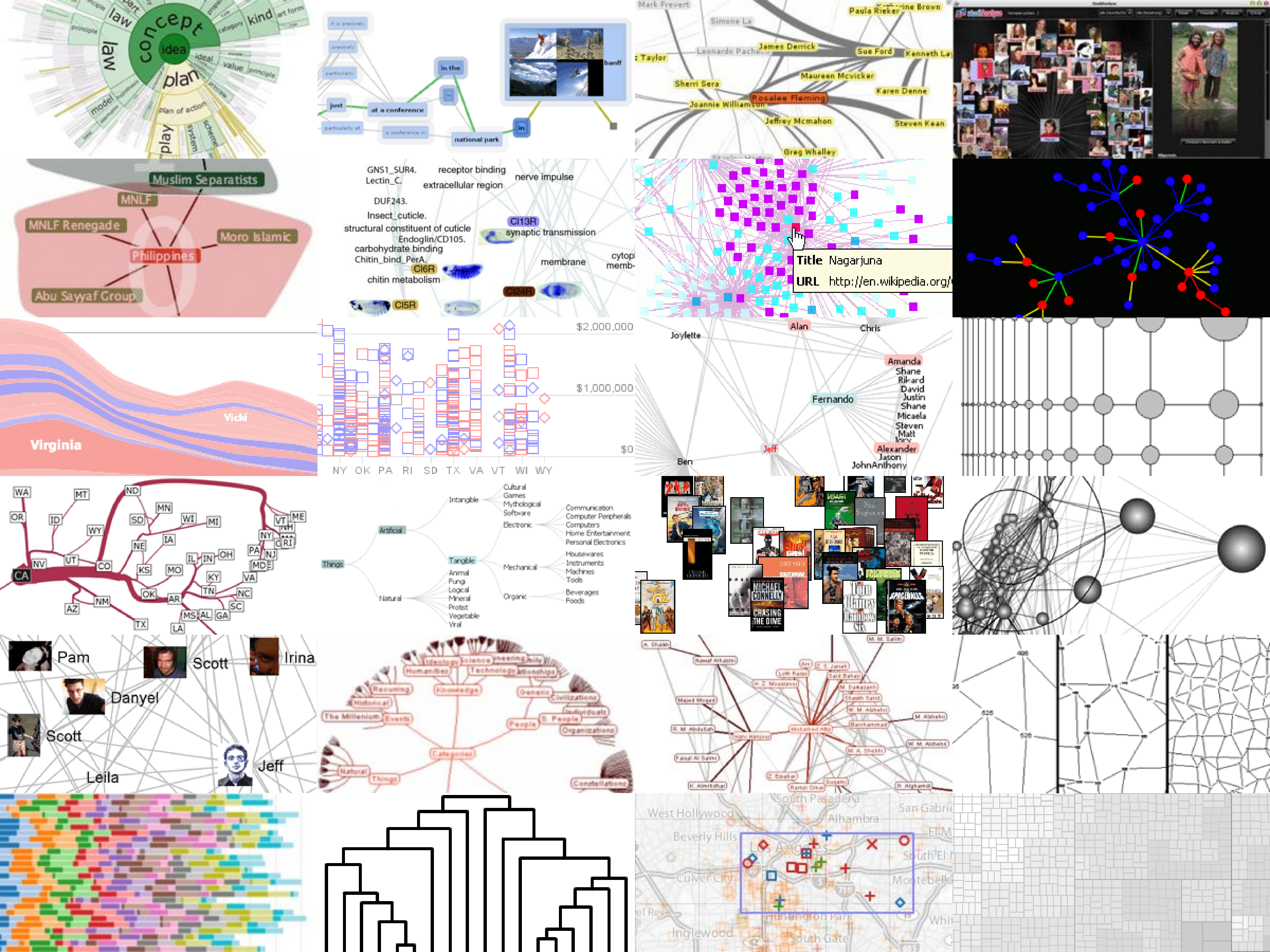
Vis = (Input Data -> Visual Objects) + Operators



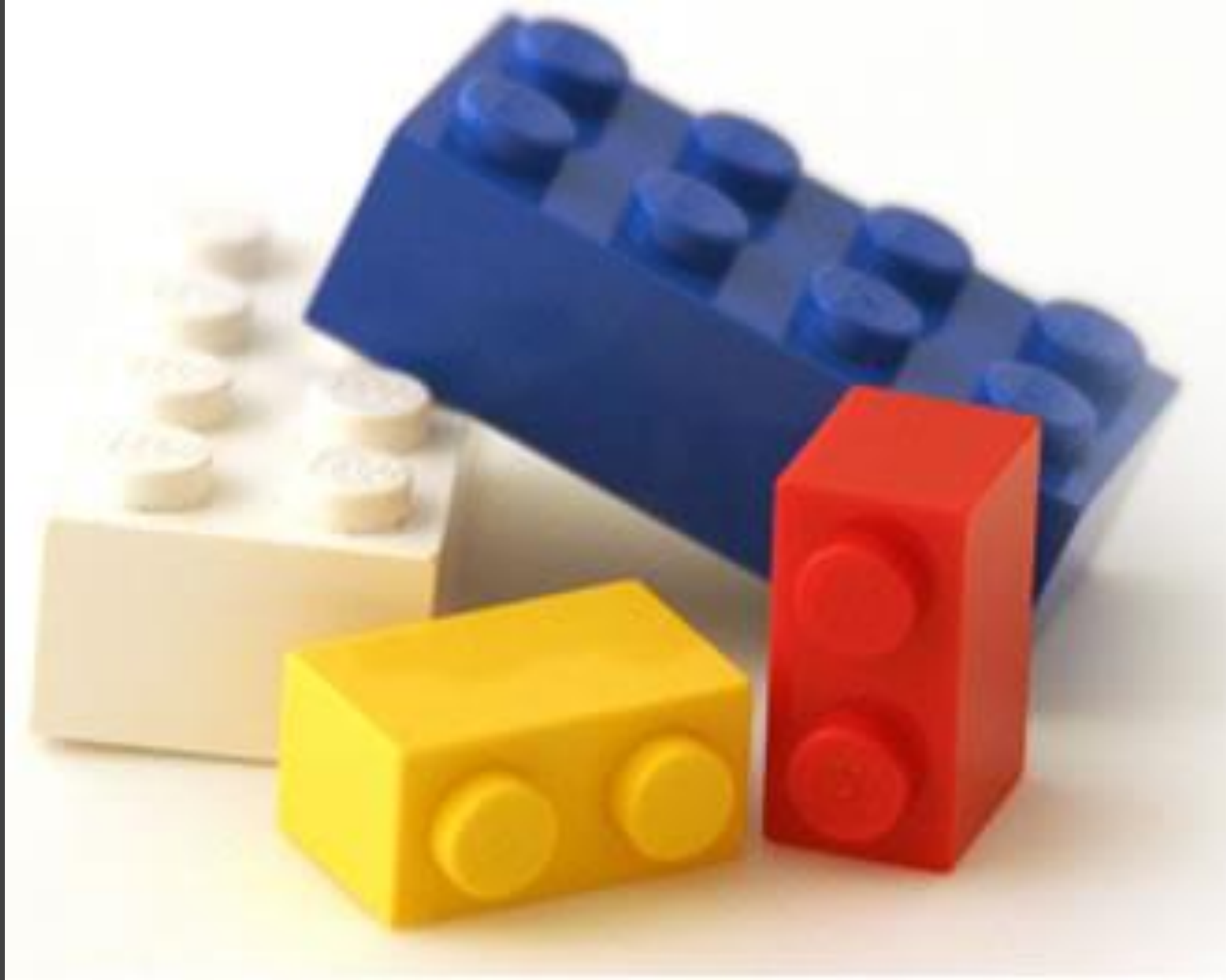
Prefuse (http://prefuse.org)

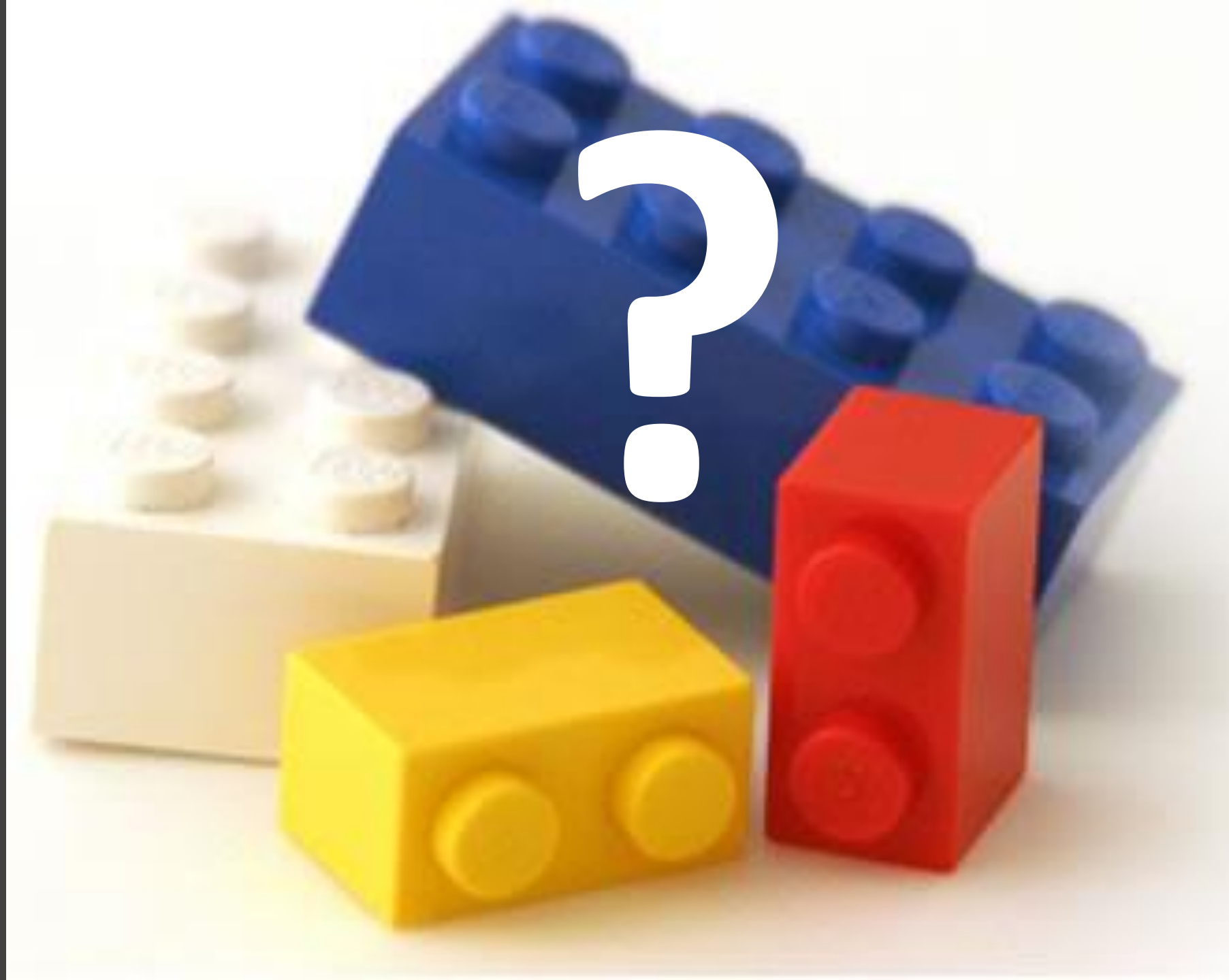


Flare (http://flare.prefuse.org)



concept
idea
law
plan
kind
plan of action
play
model
value
principle
at a conference
in the
just
national park
banff
Paula Rieker
James Derrick
Sue Ford
Sherri Sera
Maureen Mcvicker
Rosalee Fleming
Karen Denne
Joannie Williamson
Jeffrey Mcmahon
Steven Kean
Greg Whalley
Muslim Separatists
MNLF
MNLF Renegade
Moro Islamic
Philippines
Abu Sayyaf Group
receptor binding
extracellular region
nerve impulse
Insect_cuticle.
structural constituent of cuticle
Endoglin/CD105.
carbohydrate binding
synaptic transmission
membrane
chitin metabolism
Title Nagarjuna
URL http://en.wikipedia.org/
Vicki
Virginia
$2,000,000
$1,000,000
$0
NY OK PA RI SD TX VA VT WI WY
Joylette
Alan
Chris
Amanda
Fernando
Jeff
Ben
Alexander
Things
Artificial
Intangible
Tangible
Electronic
Mechanical
Natural
Organic
Pam
Scott
Irina
Danyel
Scott
Jeff
Leila
Knowledge
Categories
People
Things
West Hollywood
Beverly Hills
Culver City
South Pasadena
Alhambra
Inglewood





Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Canvas, OpenGL, Processing

Chart Typologies

Excel, Google Charts

Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Canvas, OpenGL, Processing

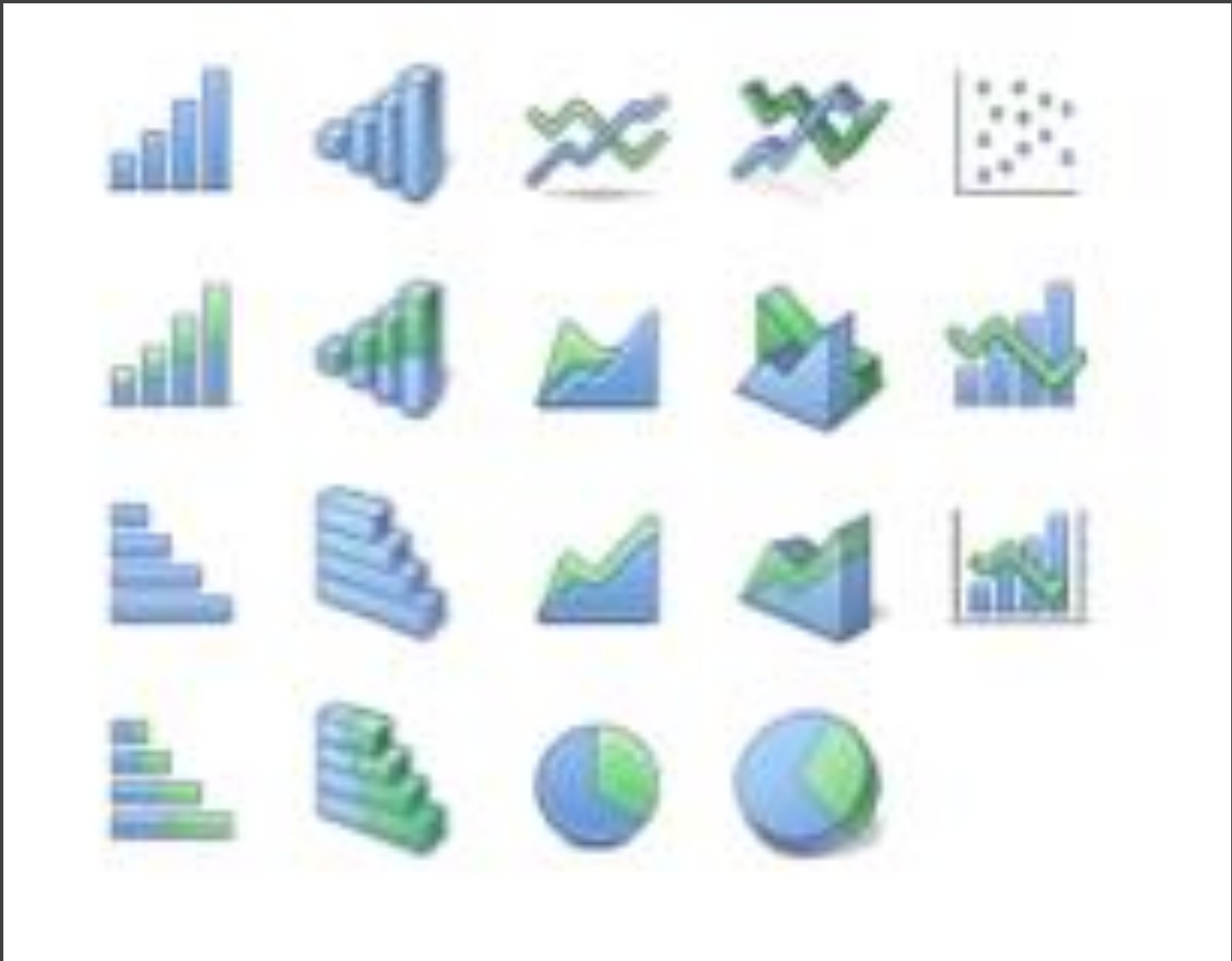


Chart Typologies

Data Sets : State Quick Facts

Uploaded By: zinggoat
Created at: Friday May 18, 3:08 PM
Data Source: US Census Bureau
Description:
Tags: people census

view as text | edit data set

	People QuickFacts	Population 2005 estimate	Population percent change April 1 2000 to July 1 2005	Population 2000	Population percent change 1990 to 2000	Persons under 5 years old percent 2004	Persons under 18 years old percent 2004	Persons 65 years old and over percent 2004
1	Alabama	4557808	0.03	4447100	0.1	0.07	0.24	0.13
2	Alaska	663661	0.06	626932	0.14	0.08	0.29	0.06
3	Arizona	5939292	0.16	5130632	0.4	0.08	0.27	0.13
4	Arkansas	2779154	0.04	2673400	0.14	0.07	0.25	0.14
5	California	36132147	0.07	33871648	0.14	0.07	0.27	0.11
6	Colorado	4665177	0.08	4301261	0.31	0.07	0.26	0.1
7	Connecticut	3510297	0.03	3405565	0.04	0.06	0.24	0.14
8	Delaware	843524	0.08	783600	0.18	0.07	0.23	0.13
9	Florida	17789864	0.11	15982378	0.24	0.06	0.23	0.17
10	Georgia	9072576	0.11	8186453	0.26	0.08	0.26	0.1
11	Hawaii	1275194	0.05	1211537	0.09	0.07	0.24	0.14
12	Idaho	1429096	0.1	1293953	0.29	0.07	0.27	0.11
13	Illinois	12763371	0.03	12419293	0.09	0.07	0.26	0.12

Choosing a visualization type for State Quick Facts

Analyze a text

Tag Cloud

How are you using your words? This enhanced tag cloud will show you the words popularity in the given set of text.

Learn more

Wordle

Wordle is a toy for generating "word clouds" from text that you provide. The clouds give greater prominence to words that appear more frequently in the source text.

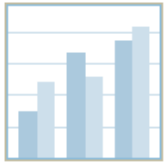
Learn more

Word Tree

See a branching view of how a word or phrase is used in a text. Navigate the text by zooming and clicking.

Learn more

Compare a set of values



Bar Chart

How do the items in your data set stack up? A bar chart is a simple and recognizable way to compare values. You can display several sets of bars for multivariate comparisons.

Learn more

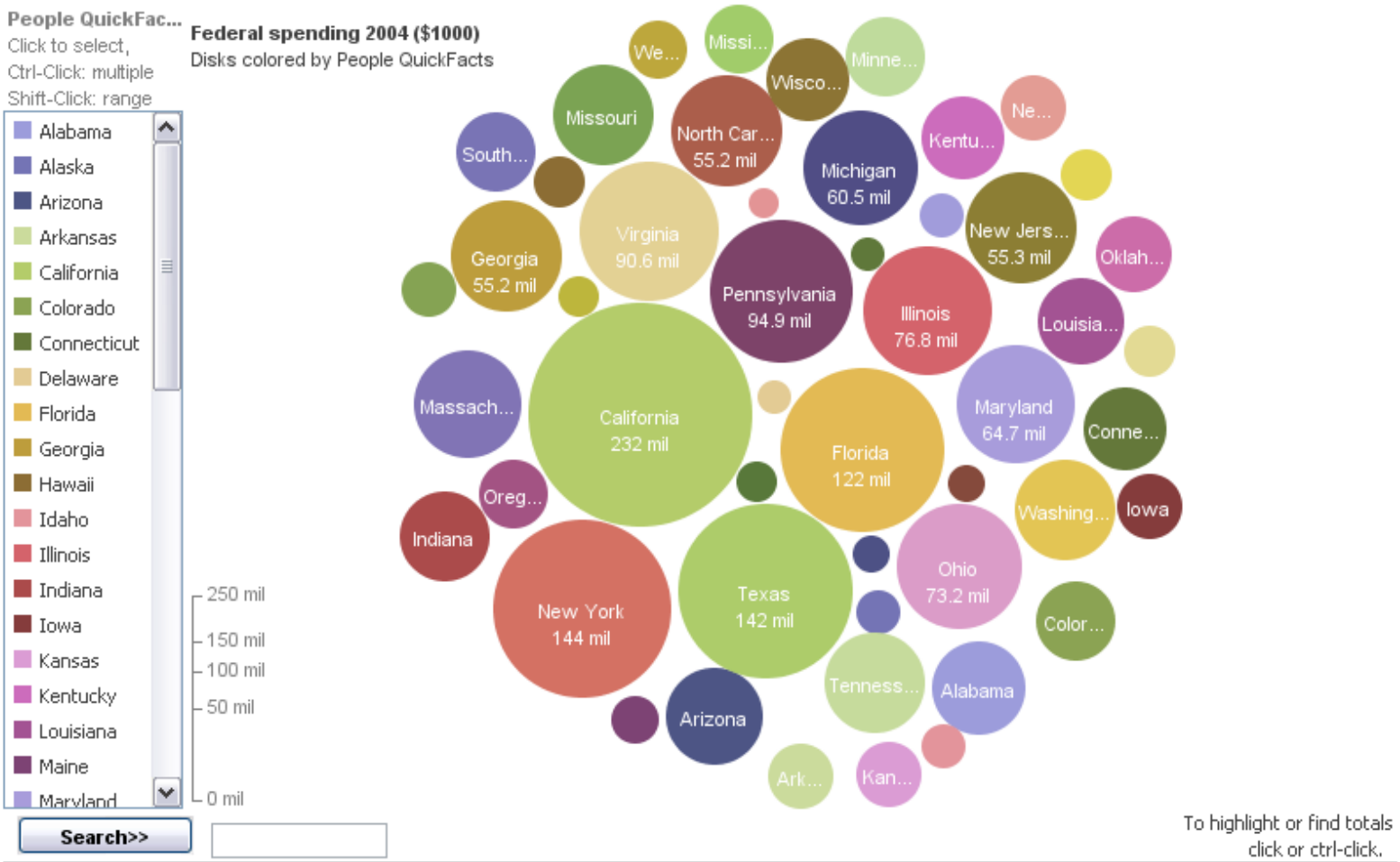
Block Histogram

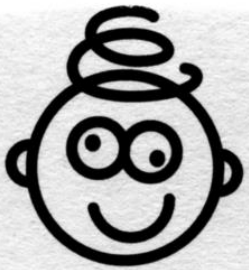
This versatile chart lets you get a quick sense of how a single set of data is distributed. Each item in the data is an individually identifiable block.

Learn more

Visualizations : Federal Spending by State, 2004

Creator: Anonymous
Tags: census people





MAD LIBS®

MY MUSIC LESSON

Every Wednesday, when I get home from school, I have a piano lesson. My teacher is a very strict house (NOUN). Her name is Hillary Clinton (CELEBRITY (FEMALE)). Our piano is a Steinway Concert tree (NOUN) and it has 88 ~~[illegible]~~ cups (PLURAL NOUN). It also has a soft pedal and a/an smily (ADJECTIVE) pedal. When I have a lesson, I sit down on the piano AIBERTO (NOUN) and play for 16 Minuts (PERIOD OF TIME). I do scales to exercise my cats (PLURAL NOUN), and then I usually play a minuet by Johann Sebastian Washington (CELEBRITY (LAST NAME)). Teacher says I am a natural Haunted House (NOUN) and have a good musical leg (PART OF THE BODY). Perhaps when I get better I will become a concert vet (PROFESSION) and give a recital at Carnegie hospital (TYPE OF BUILDING).

[M]ost charting packages channel user requests into a **rigid array of chart types**. To atone for this lack of flexibility, they offer a kit of post-creation editing tools to return the image to what the user originally envisioned. **They give the user an impression of having explored data rather than the experience**.

Leland Wilkinson
The Grammar of Graphics, 1999

Chart Typologies

Excel, Many Eyes, Google Charts

Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Canvas, OpenGL, Processing

Chart Typologies

Excel, Many Eyes, Google Charts

Visual Analysis Grammars

VizQL, ggplot2

Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Canvas, OpenGL, Processing

Tableau - Book2

File Edit View Format Data Analysis Table Bookmark Window Help

Show Me!

Schema

congress.csv Connection

Find:

Dimensions

- Abc Candidate
- Abc Candidate ID
- Abc General Elec Status
- Abc Incumbent/Challenger/Open-Seal
- # Party
- Abc Party Desig
- Abc Primary Elec Status
- Abc Runoff Elec Status
- Abc Spec Elec Status
- Abc State Code
- # Year
- Abc *Measure Names*

Measures

- # District
- # General Elec Pct
- # Total Receipts
- # *Measure Values*

Groups

Filters:

Level of Detail:

Mark: Automatic

Text:

Color: Party

Size:

Legend:

1
2
3

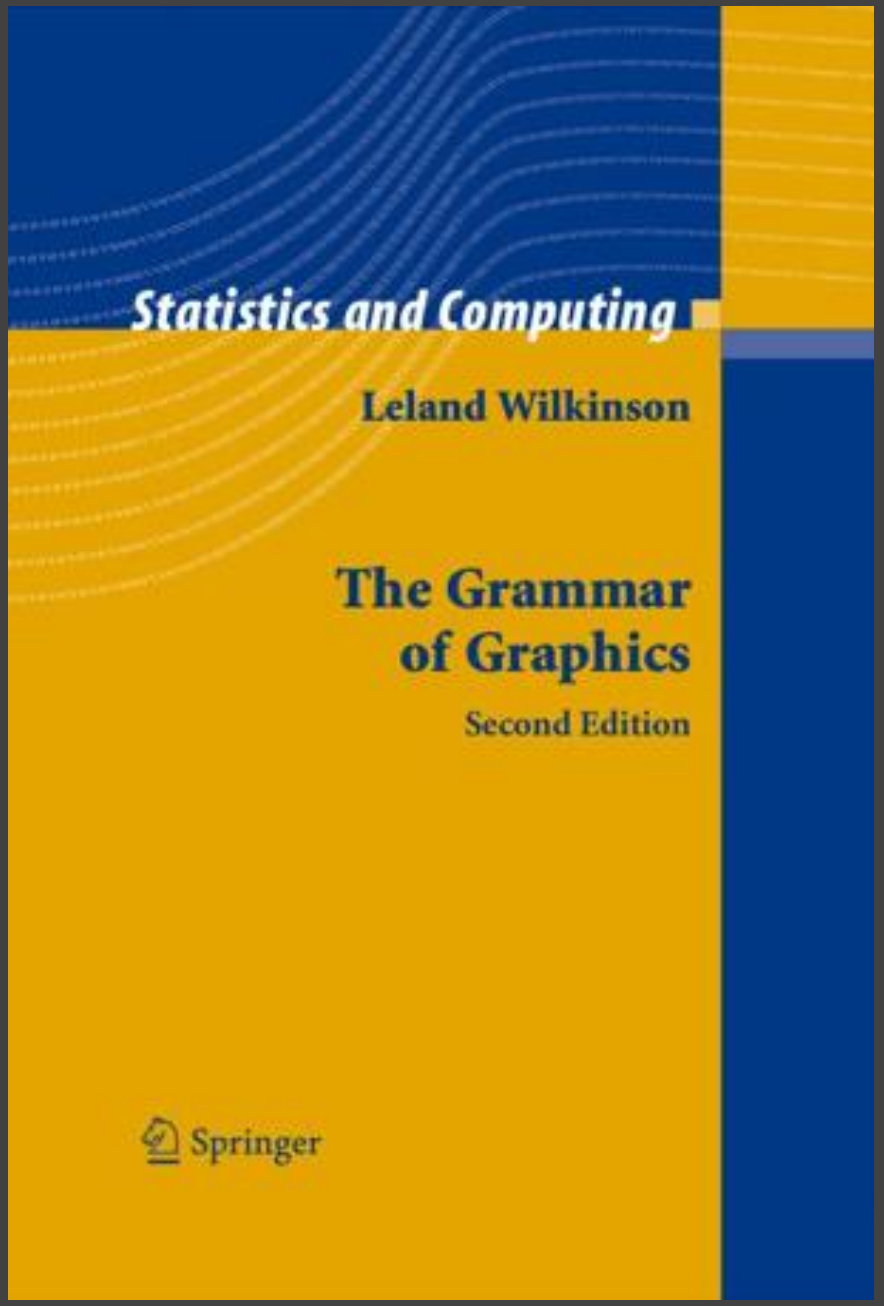
Size:

Columns: Party | Year

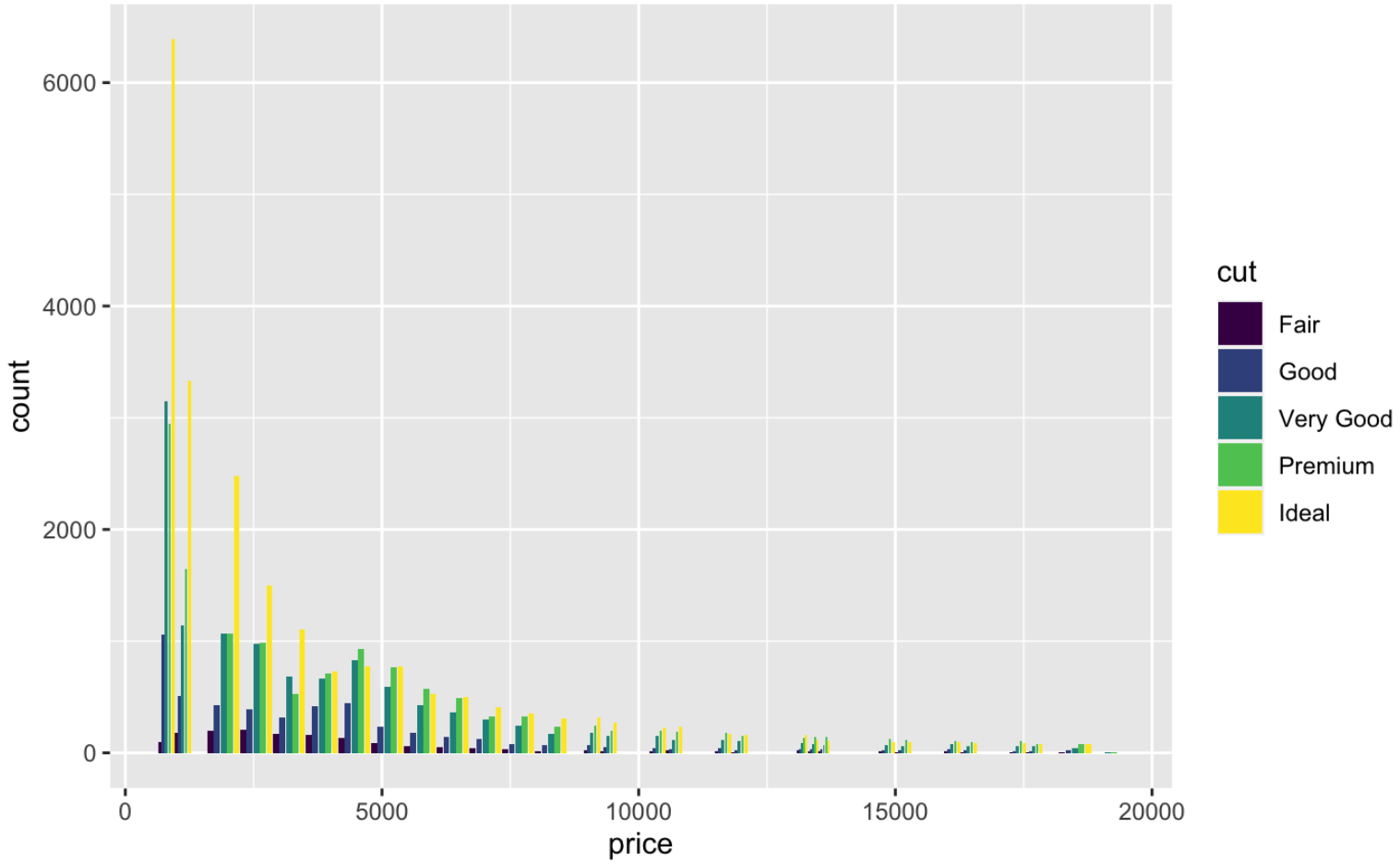
Rows: SUM(Total..

Sheet 1

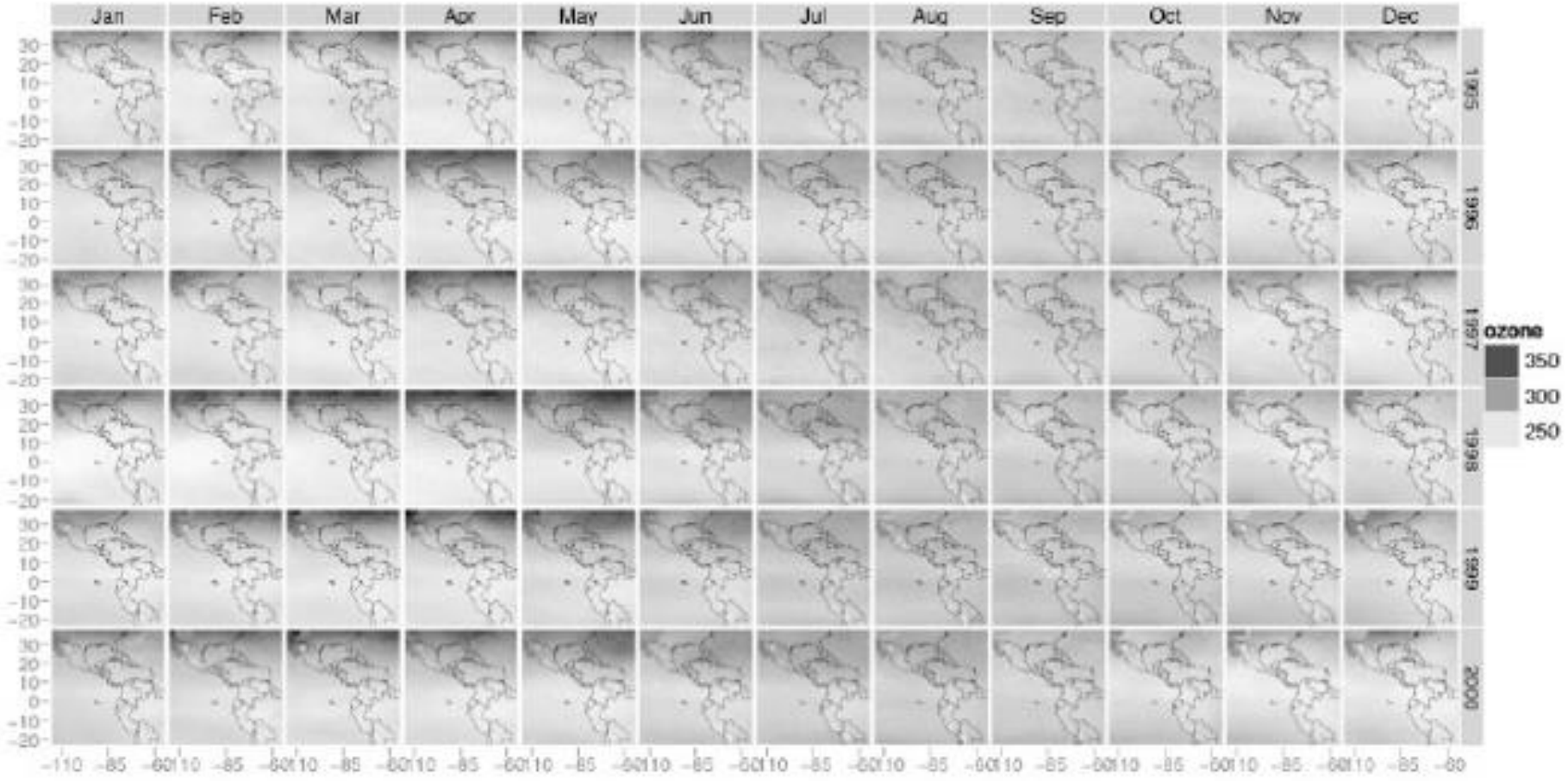
Ready


Statistics and Computing
Leland Wilkinson
The Grammar of Graphics
Second Edition
Springer

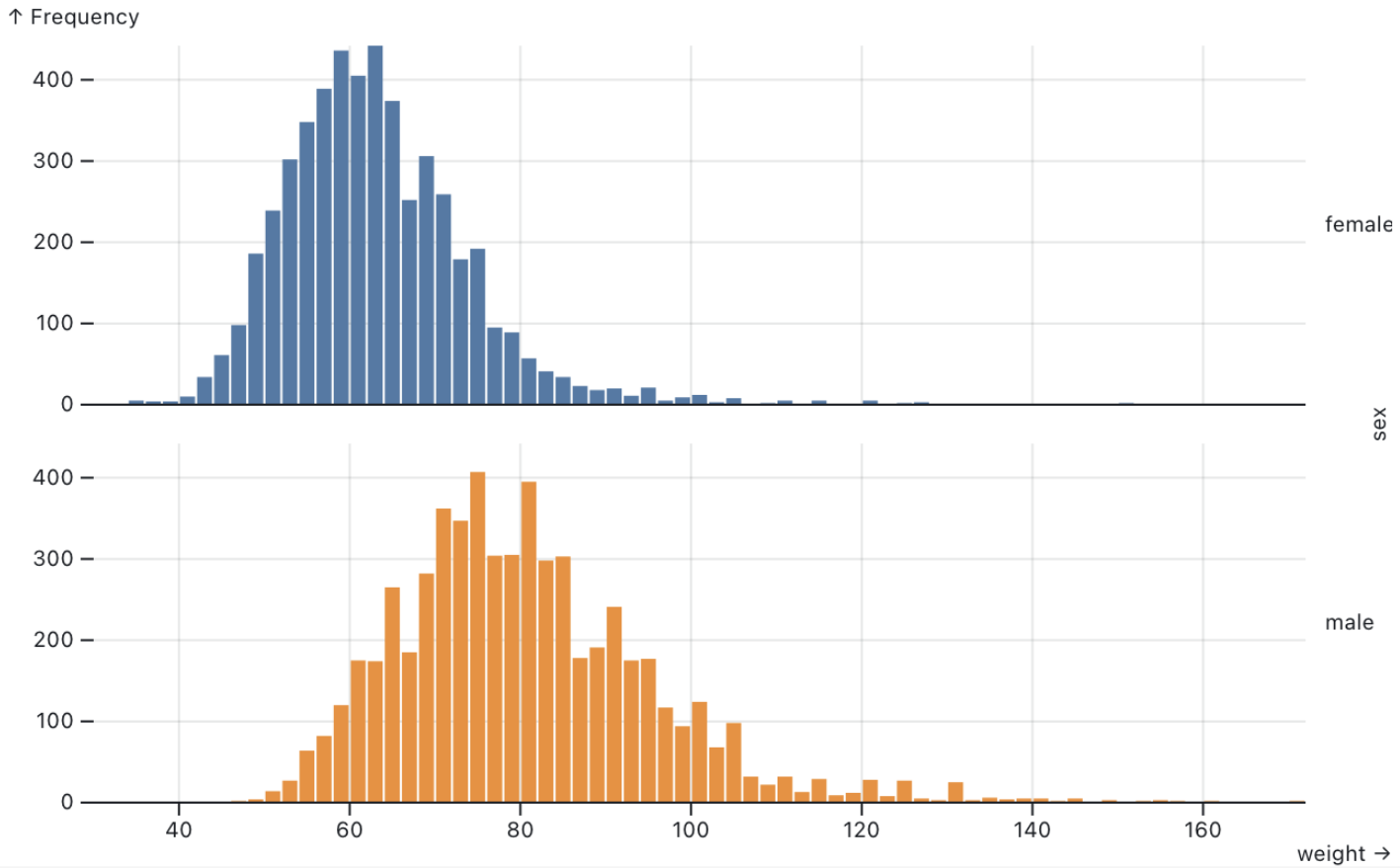
```
ggplot(diamonds, aes(x=price, fill=cut))
+ geom_bar(position="dodge")
```




```
ggplot(diamonds, aes(x=price, fill=cut))
+ geom_bar(position="dodge")
```




```
qplot(long, lat, data = expo, geom = "tile", fill = ozone,
  facets = year ~ month) +
scale_fill_gradient(low = "white", high = "black") + map
```




```
Plot.plot({
  grid: true,
  facet: {
    data: athletes,
    y: "sex"
  },
  marks: [
    Plot.rectY(athletes, Plot.binX({y: "count"}, {x: "weight", fill: "sex"})),
    Plot.ruleY([0])
  ]
})
```

Observable Plot

Chart Typologies

Excel, Many Eyes, Google Charts

Visual Analysis Grammars

VizQL, ggplot2

Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Canvas, OpenGL, Processing

Chart Typologies

Excel, Many Eyes, Google Charts

Visual Analysis Grammars

VizQL, ggplot2

Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Canvas, OpenGL, Processing

Chart Typologies

Excel, Many Eyes, Google Charts

Visual Analysis Grammars

VizQL, ggplot2

Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Canvas, OpenGL, Processing

Chart Typologies

Excel, Many Eyes, Google Charts

Visual Analysis Grammars

VizQL, ggplot2

?

Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Canvas, OpenGL, Processing

Chart Typologies

Excel, Many Eyes, Google Charts

Visual Analysis Grammars

VizQL, ggplot2

Visualization Grammars

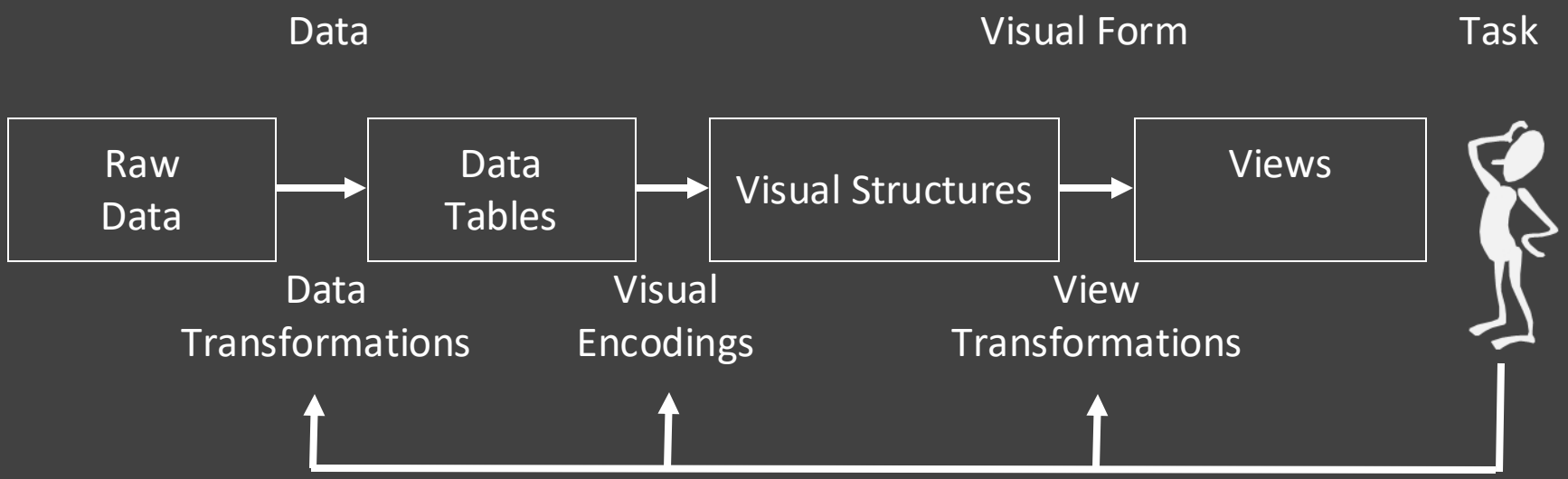
Protovis, D3.js

Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Canvas, OpenGL, Processing


Data
Visual Form
Task
Raw Data
Data Tables
Visual Structures
Views
Data Transformations
Visual Encodings
View Transformations

Visualization Grammars

Visualization Grammars

Data	Input data to visualize

Visualization Grammars

Data	Input data to visualize
Transforms	Group, aggregate, stats, layout

Visualization Grammars

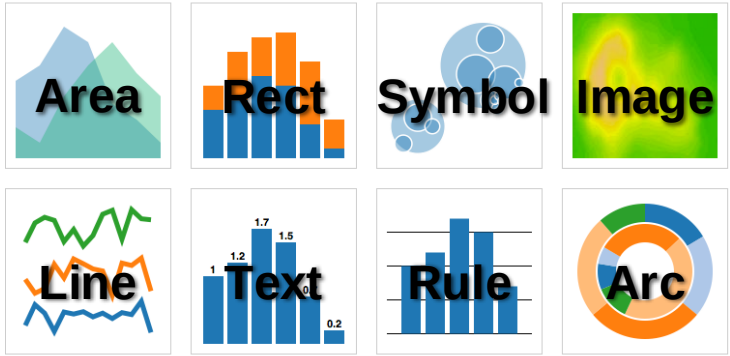
Data	Input data to visualize
Transforms	Group, aggregate, stats, layout
Scales	Map data values to visual values

Visualization Grammars

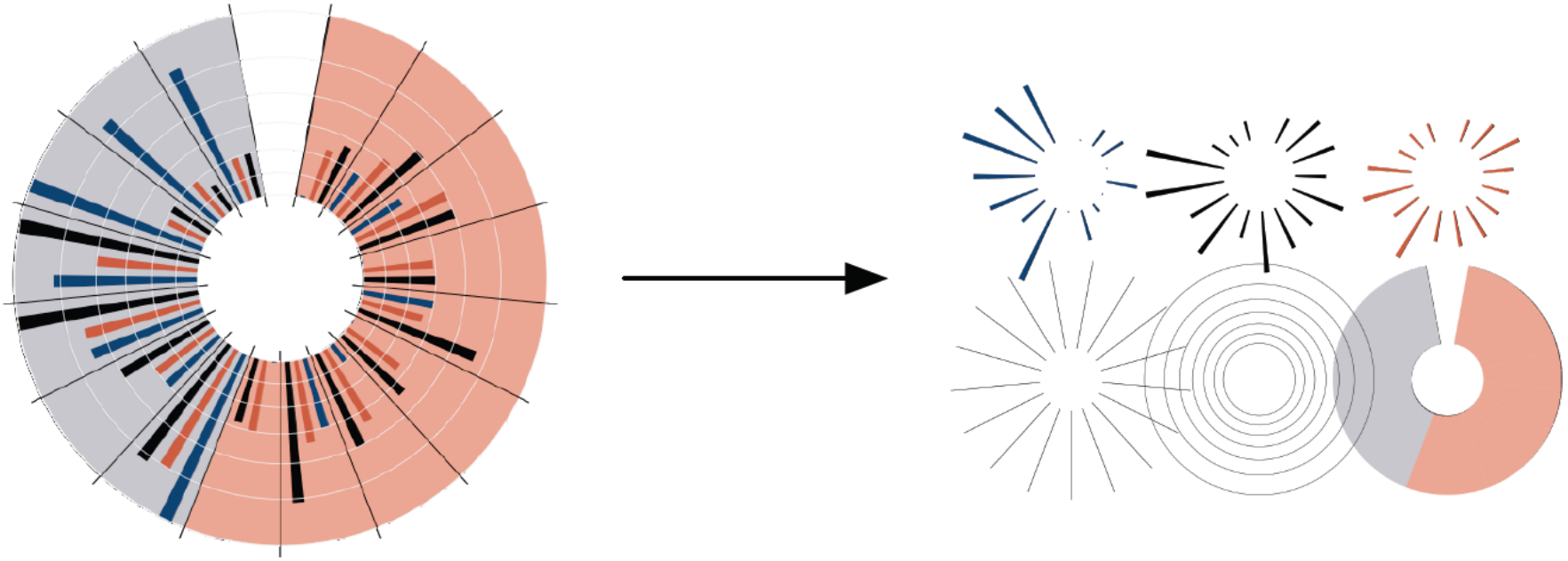
Data	Input data to visualize
Transforms	Group, aggregate, stats, layout
Scales	Map data values to visual values
Guides	Axes & legends visualize scales

Visualization Grammars

Data	Input data to visualize
Transforms	Group, aggregate, stats, layout
Scales	Map data values to visual values
Guides	Axes & legends visualize scales
Marks	Data-representative graphics

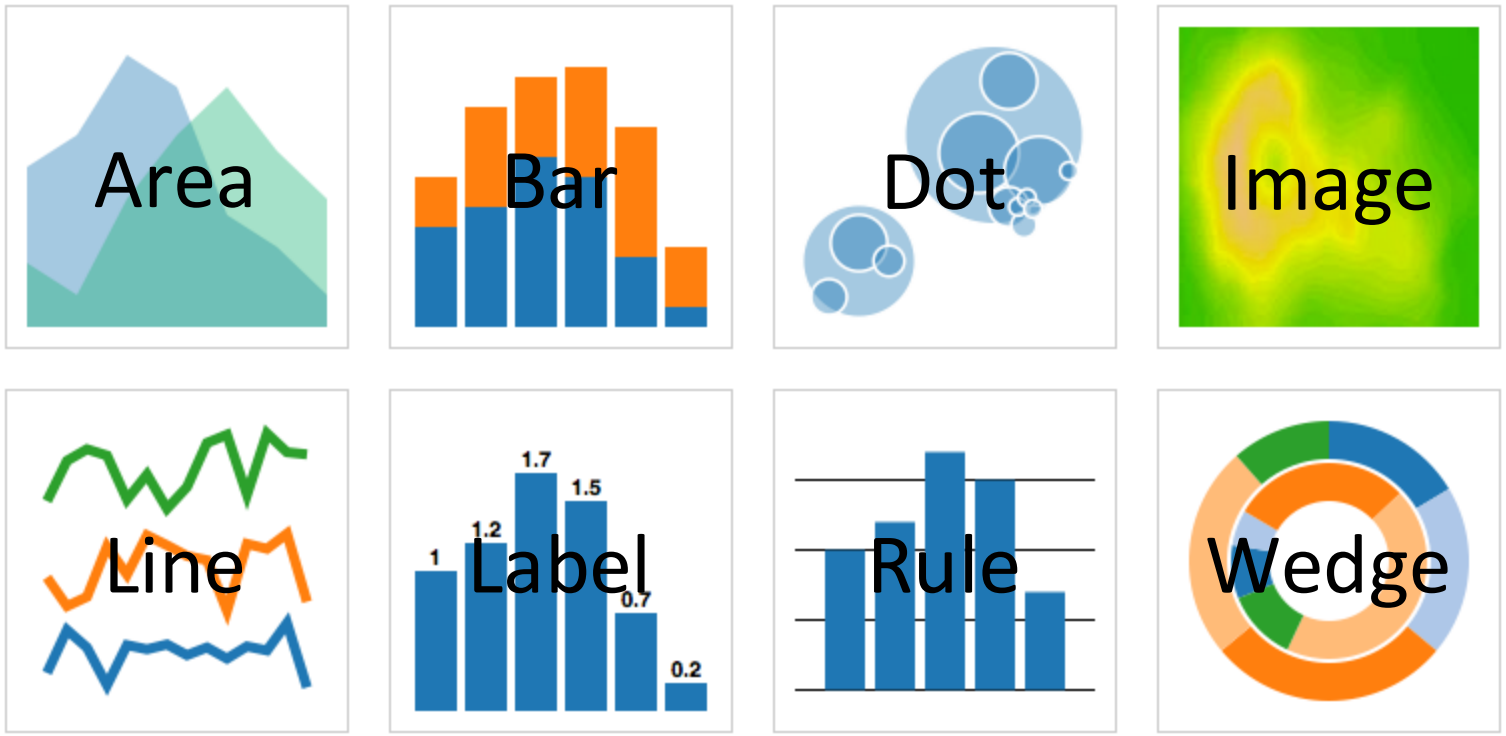


Protovis: A Grammar for Visualization



A graphic is a composition of data-representative marks.

with **Mike Bostock** & **Vadim Ogievetsky**



MARKS: Protovis graphical primitives

MARK

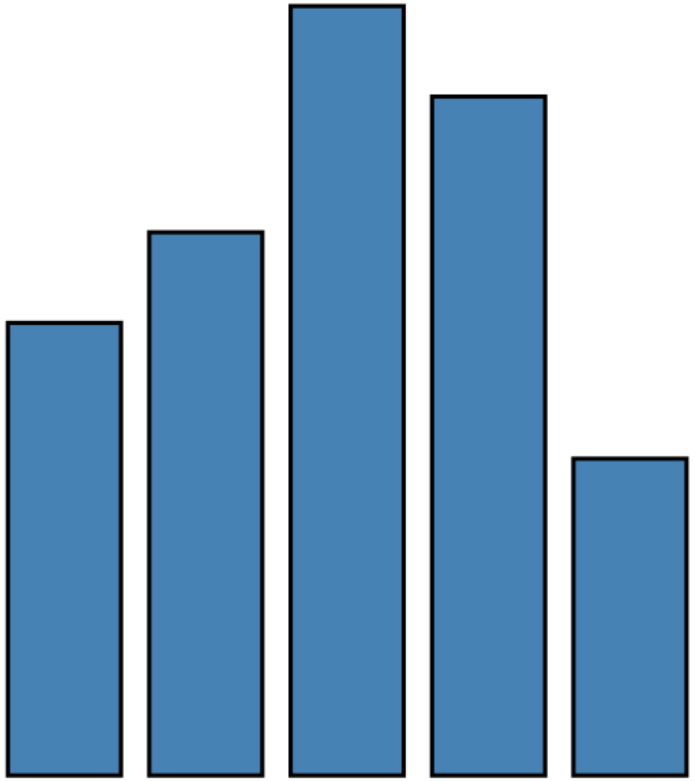
$\lambda : D \rightarrow R$

data	λ
visible	λ
left	λ
bottom	λ
width	λ
height	λ
fillStyle	λ
strokeStyle	λ
lineWidth	λ
…	λ

RECT

$\lambda : D \rightarrow R$

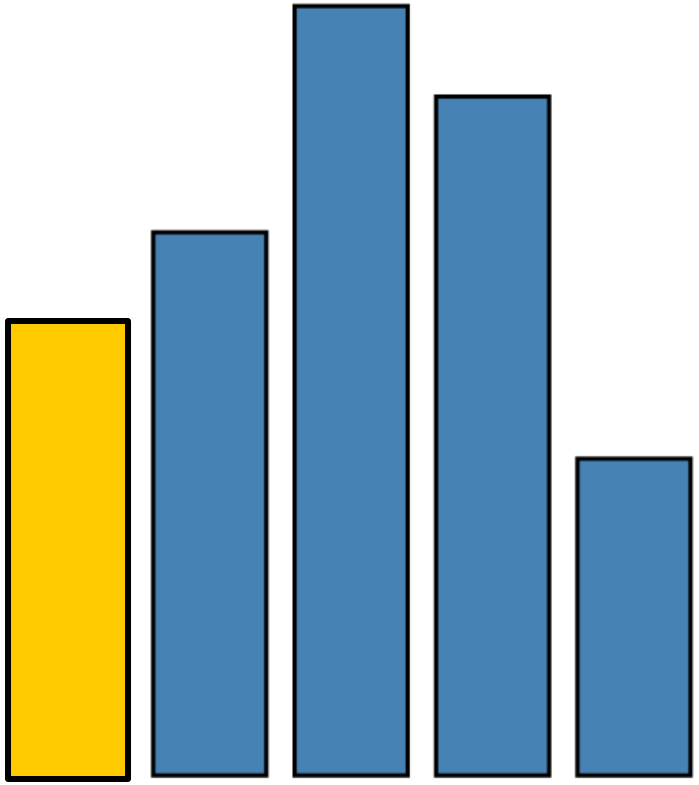
data	1	1.2	1.7	1.5	0.7
visible	true				
left	λ: index * 25				
bottom	0				
width	20				
height	λ: datum * 80				
fillStyle	blue				
strokeStyle	black				
lineWidth	1.5				
…	…				



RECT

$\lambda : D \rightarrow R$

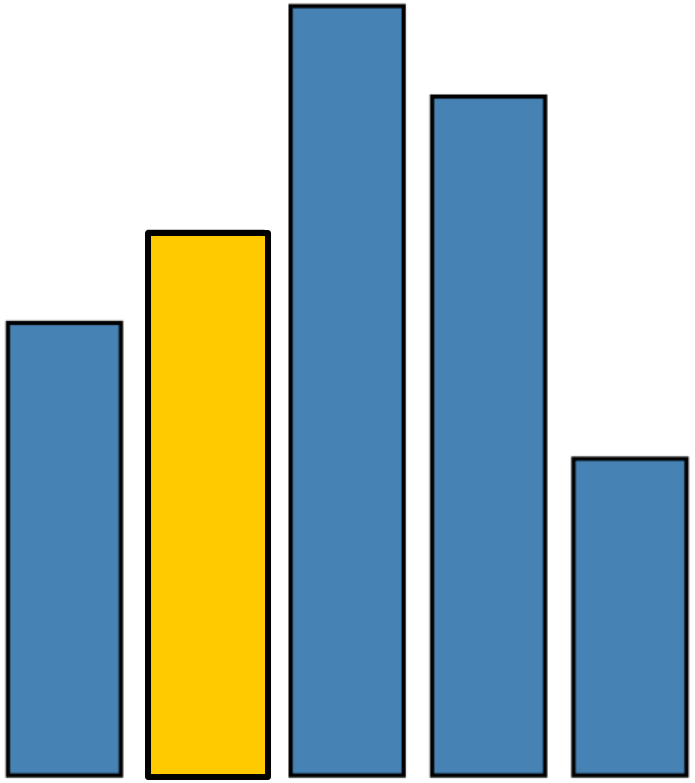
data	1	1.2	1.7	1.5	0.7
visible	true				
left	0 * 25				
bottom	0				
width	20				
height	1 * 80				
fillStyle	blue				
strokeStyle	black				
lineWidth	1.5				
…	…				



RECT

$\lambda : D \rightarrow R$

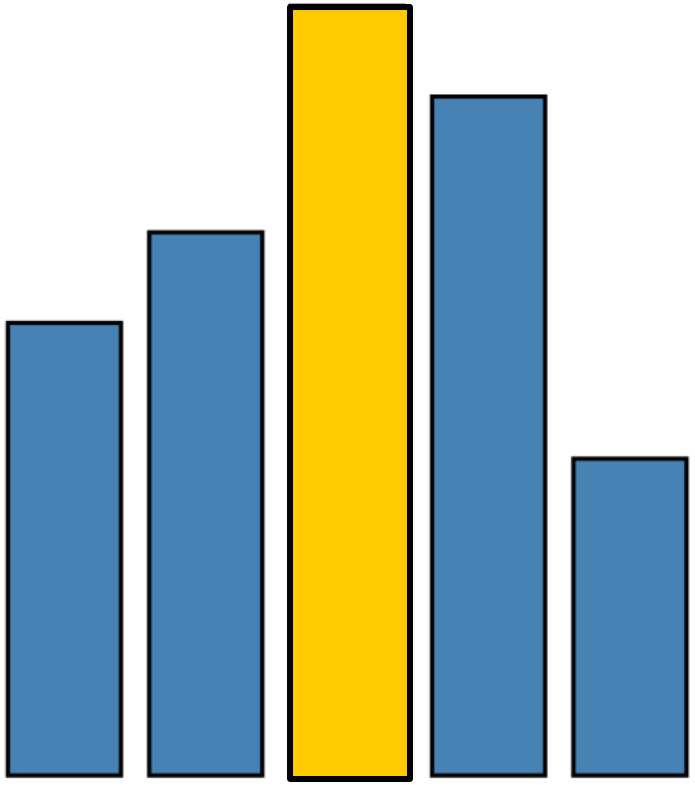
data	1	1.2	1.7	1.5	0.7
visible	true				
left	1 * 25				
bottom	0				
width	20				
height	1.2 * 80				
fillStyle	blue				
strokeStyle	black				
lineWidth	1.5				
…	…				



RECT

$\lambda : D \rightarrow R$

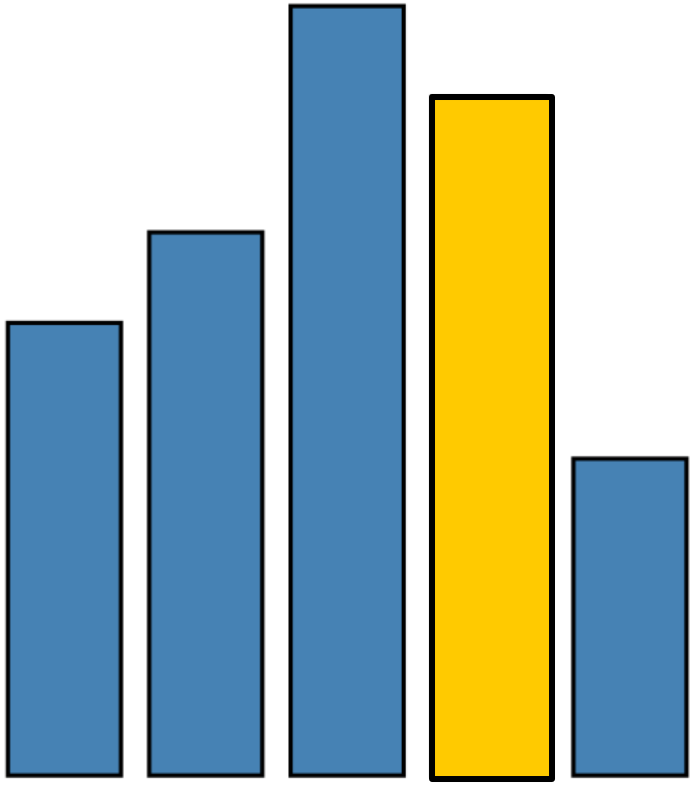
data	1	1.2	1.7	1.5	0.7
visible	true				
left	2 * 25				
bottom	0				
width	20				
height	1.7 * 80				
fillStyle	blue				
strokeStyle	black				
lineWidth	1.5				
…	…				



RECT

$\lambda : D \rightarrow R$

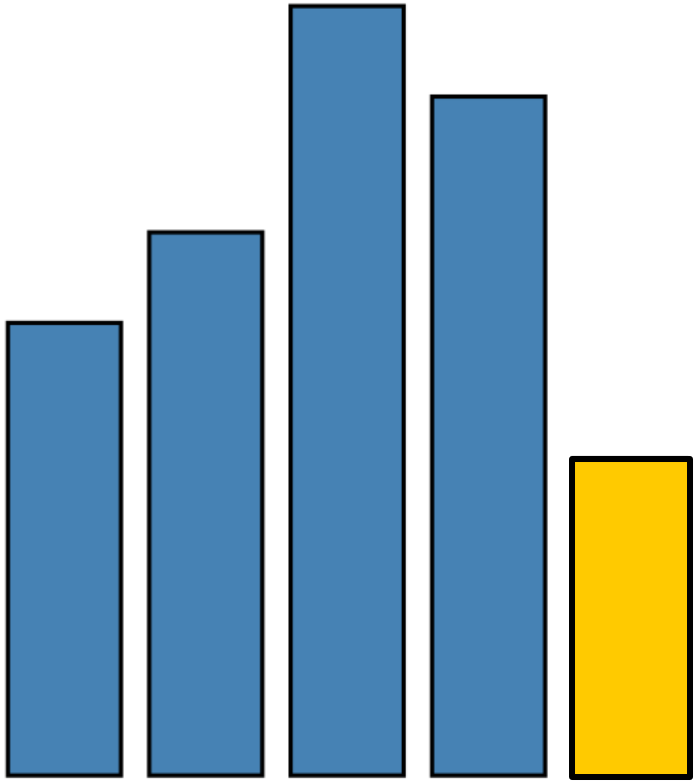
data	1	1.2	1.7	1.5	0.7
visible	true				
left	3 * 25				
bottom	0				
width	20				
height	1.5 * 80				
fillStyle	blue				
strokeStyle	black				
lineWidth	1.5				
…	…				



RECT

$\lambda : D \rightarrow R$

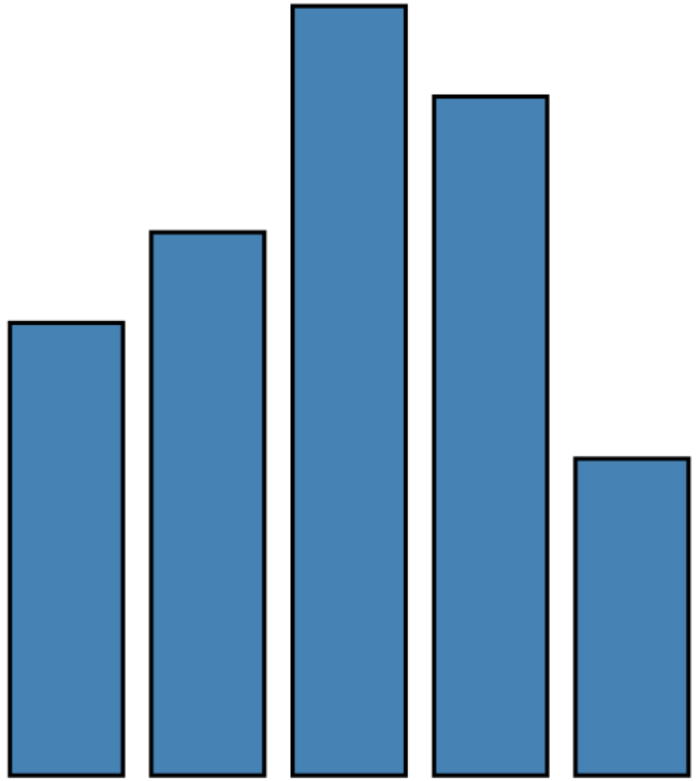
data	1	1.2	1.7	1.5	0.7
visible	true				
left	4 * 25				
bottom	0				
width	20				
height	0.7 * 80				
fillStyle	blue				
strokeStyle	black				
lineWidth	1.5				
…	…				



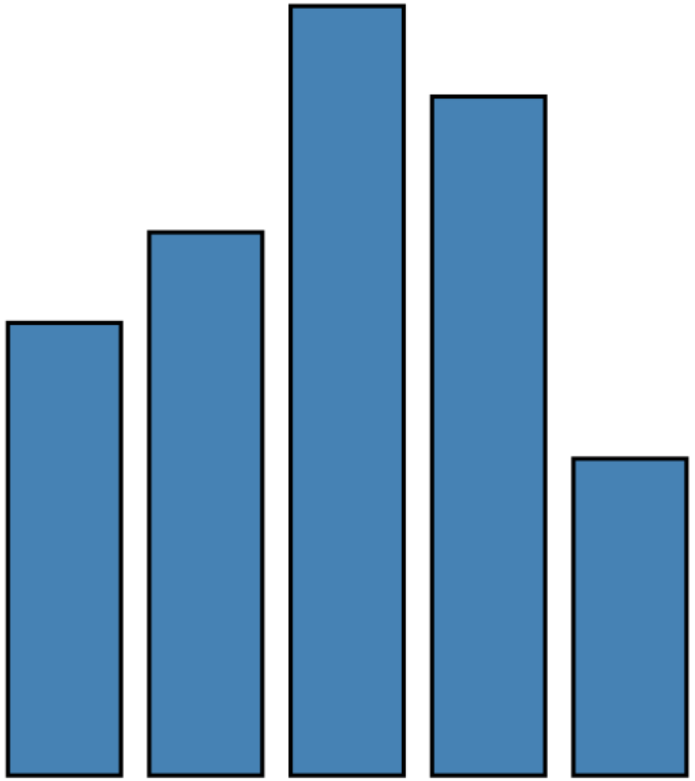
RECT

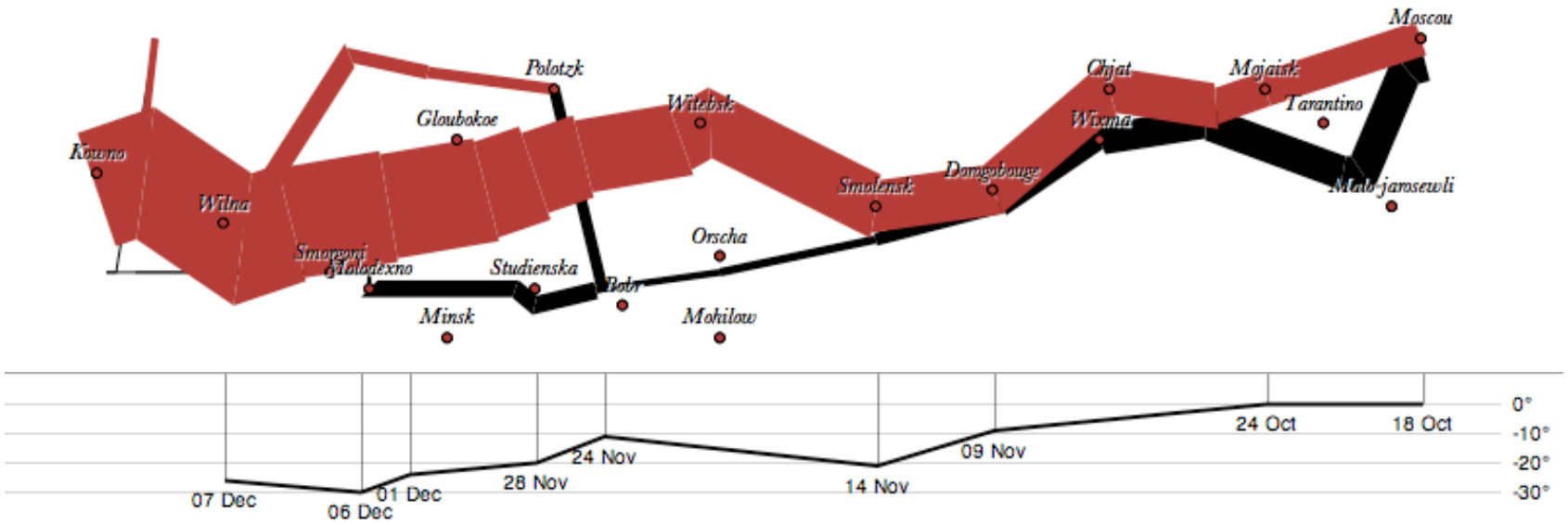
$$\lambda : D \rightarrow R$$

data	1	1.2	1.7	1.5	0.7
visible	true				
left	λ: index * 25				
bottom	0				
width	20				
height	λ: datum * 80				
fillStyle	blue				
strokeStyle	black				
lineWidth	1.5				
…	…				



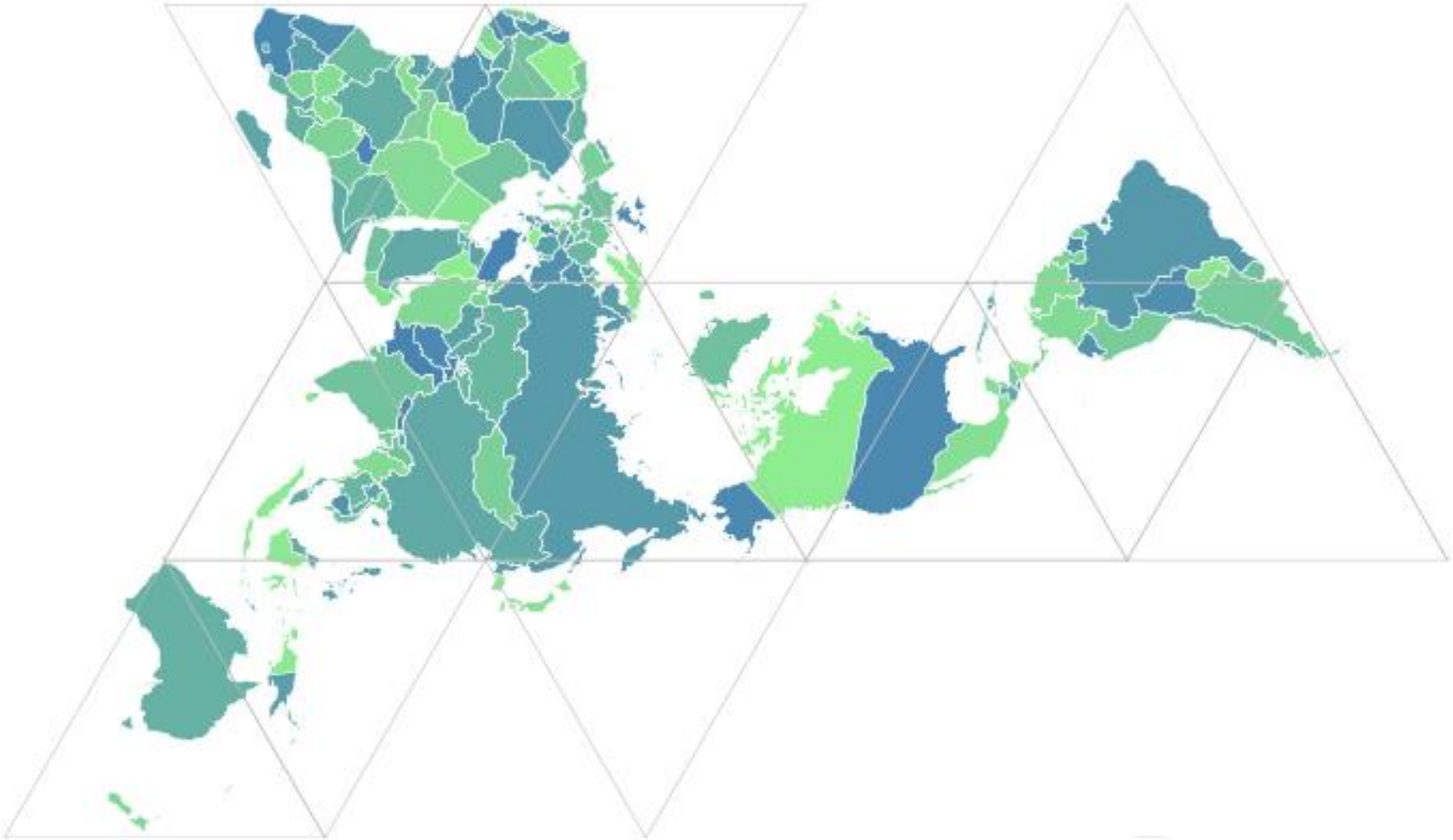
```
var vis = new pv.Panel();
vis.add(pv.Bar)
  .data([1, 1.2, 1.7, 1.5, 0.7])
  .visible(true)
  .left((d) => this.index * 25);
  .bottom(0)
  .width(20)
  .height((d) => d * 80)
  .fillStyle("blue")
  .strokeStyle("black")
  .lineWidth(1.5);
vis.render();
```





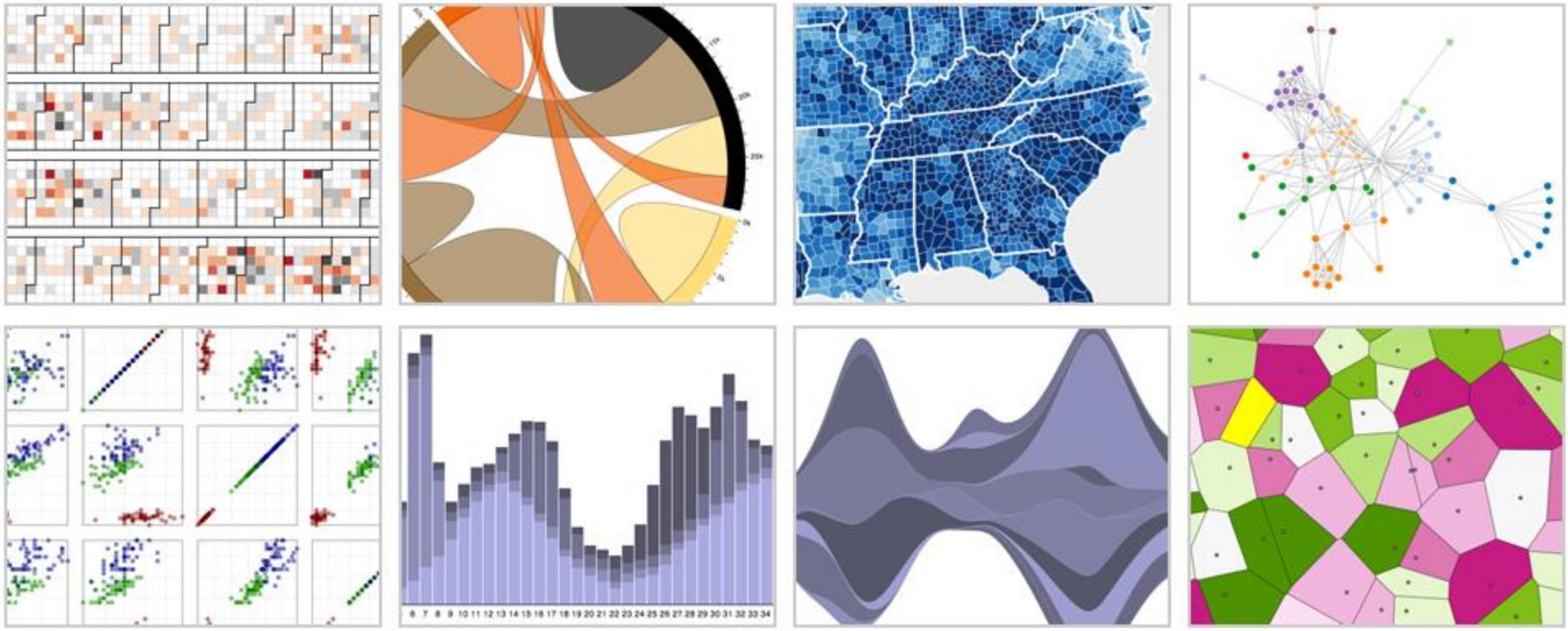

```
var army = pv.nest(napoleon.army, "dir", "group“);
var vis = new pv.Panel();

var lines = vis.add(pv.Panel).data(army);
lines.add(pv.Line)
  .data(() => army[this.idx])
  .left(lon).top(lat).size((d) => d.size/8000)
  .strokeStyle(() => color[army[paneIndex][0].dir]);

vis.add(pv.Label).data(napoleon.cities)
  .left(lon).top(lat)
  .text((d) => d.city).font("italic 10px Georgia")
  .textAlign("center").textBaseline("middle");
```

```
vis.add(pv.Rule).data([0,-10,-20,-30])
  .top((d) => 300 - 2*d - 0.5).left(200).right(150)
  .lineWidth(1).strokeStyle("#ccc")
  .anchor("right").add(pv.Label)
    .font("italic 10px Georgia")
    .text((d) => d+"°").textBaseline("center");

vis.add(pv.Line).data(napoleon.temp)
  .left(lon).top(tmp) .strokeStyle("#0")
 .add(pv.Label)
  .top((d) => 5 + tmp(d))
  .text((d) => d.temp+"° "+d.date.substr(0,6))
  .textBaseline("top").font("italic 10px Georgia");
```



Dymaxion Maps | Vadim Ogievetsky

d3.js Data-Driven Documents



with **Mike Bostock**, Jason Davies & Vadim Ogievetsky

Protovis

Specialized mark types

+ Streamlined design
- Limits expressiveness
- More overhead (slower)
- Harder to debug
- Self-contained model

Specify a scene (nouns)

+ Quick for static vis
- Delayed evaluation
- Animation, interaction are more cumbersome

Protovis

Specialized mark types

+ Streamlined design
- Limits expressiveness
- More overhead (slower)
- Harder to debug
- Self-contained model

Specify a scene (nouns)

+ Quick for static vis
- Delayed evaluation
- Animation, interaction are more cumbersome

D3

Bind data to DOM

- Exposes SVG/CSS/…
+ Exposes SVG/CSS/…
+ Less overhead (faster)
+ Debug in browser
+ Use with other tools

Transform a scene (verbs)

- More complex model
+ Immediate evaluation
+ Dynamic data, anim, and interaction natural

D3 Selections

The core abstraction in D3 is a ***selection***.

D3 Selections

The core abstraction in D3 is a ***selection***.

```
// Add and configure an SVG element (<svg width="500" height="300">)
var svg = d3.append("svg")      // add new SVG to page body
    .attr("width", 500)                 // set SVG width to 500px

    .attr("height", 300);               // set SVG height to 300px
```

D3 Selections

The core abstraction in D3 is a ***selection***.

```
// Add and configure an SVG element (<svg width="500" height="300">)
var svg = d3.append("svg")      // add new SVG to page body
    .attr("width", 500)             // set SVG width to 500px

    .attr("height", 300);           // set SVG height to 300px
// Select & update existing rectangles contained in the SVG element
svg.selectAll("rect")           // select all SVG rectangles
    .attr("width", 100)             // set rect widths to 100px
    .style("fill", "steelblue");    // set rect fill colors
```

Data Binding

Selections can ***bind*** data and DOM elements.

```
var values = [ {…}, {…}, {…}, … ]; // input data as JS objects
```

Data Binding

Selections can ***bind*** data and DOM elements.

```
var values = [ {…}, {…}, {…}, … ]; // input data as JS objects
// Select SVG rectangles and bind them to data values.
var bars = svg.selectAll(“rect.bars”).data(values);
```

Data Binding

Selections can ***bind*** data and DOM elements.

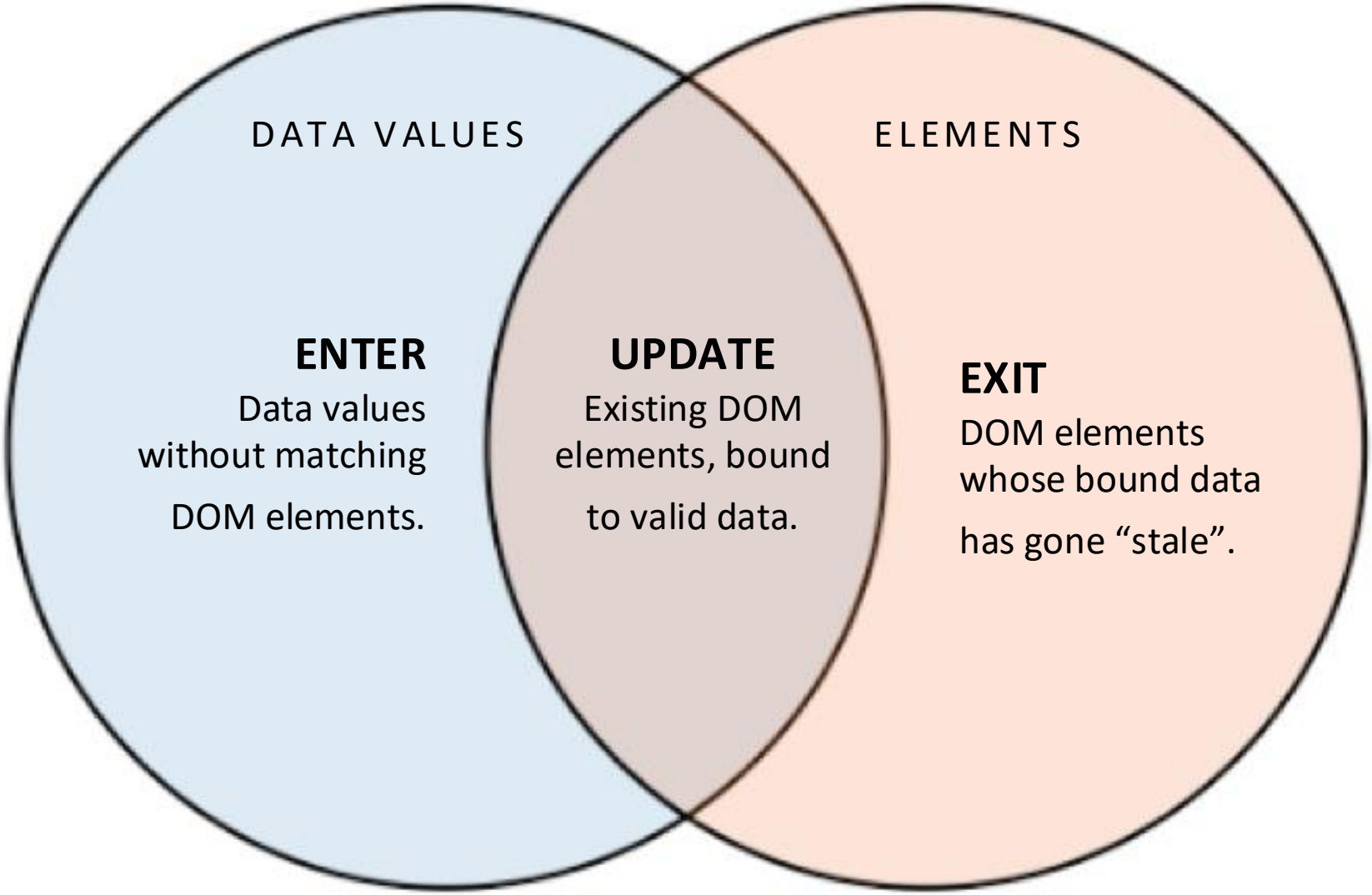
```
var values = [ {…}, {…}, {…}, … ]; // input data as JS objects
// Select SVG rectangles and bind them to data values.

var bars = svg.selectAll(“rect.bars”).data(values);
// What if the DOM elements don’t exist yet? The enter set represents data
// values that do not yet have matching DOM elements.

bars.enter().append(“rect”).attr(“class”, “bars”);
```

Data Binding

Selections can ***bind*** data and DOM elements.

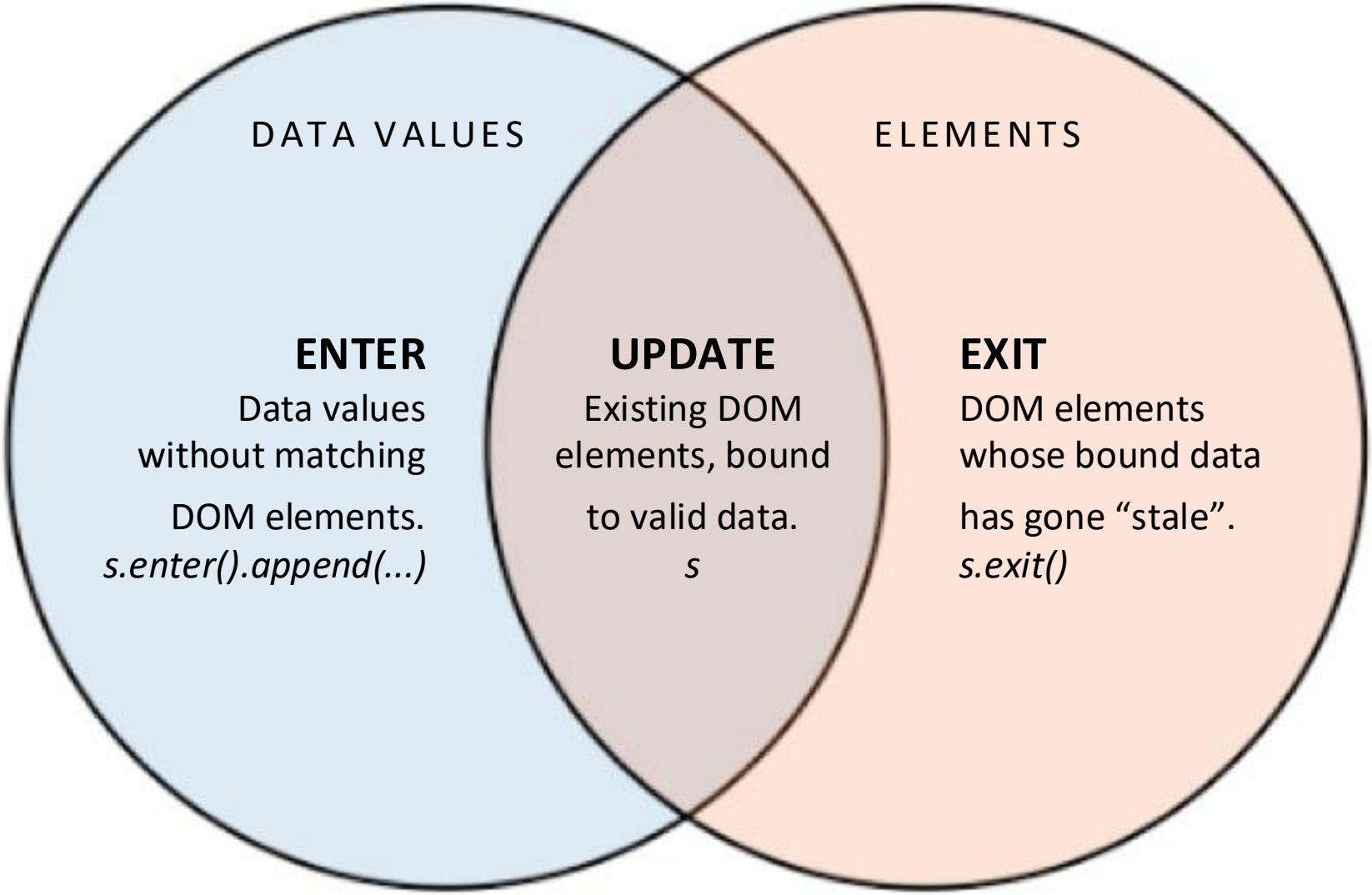
```
var values = [ {…}, {…}, {…}, … ]; // input data as JS objects
// Select SVG rectangles and bind them to data values.

var bars = svg.selectAll(“rect.bars”).data(values);
// What if the DOM elements don’t exist yet? The enter set represents data
// values that do not yet have matching DOM elements.

bars.enter().append(“rect”).attr(“class”, “bars”);
// What if data values are removed? The exit set is a selection of existing
// DOM elements who no longer have matching data values.
bars.exit().remove();
```

The Data Join



The Data Join

var s = d3.selectAll(...).data(...)



Data Binding

Selections can ***bind*** data and DOM elements.

```
var values = [ {…}, {…}, {…}, … ]; // input data as JS objects
// Select SVG rectangles and bind them to data values.
var bars = svg.selectAll(“rect.bars”).data(values)
    .join(
        enter => enter.append("rect"), // create new
        update => update,              // update current
        exit => exit.remove()          // remove outdated
    )
```

D3 Modules

Data Parsing / Formatting (JSON, CSV, …)
Shape Helpers (arcs, curves, areas, symbols, …)
Scale Transforms (linear, log, ordinal, …)
Color Spaces (RGB, HSL, LAB, …)
Animated Transitions (tweening, easing, …)
Geographic Mapping (projections, clipping, …)
Layout Algorithms (stack, pie, force, trees, …)

Interactive Behaviors (brush, zoom, drag, …)
Many of these correspond to future lecture topics!

Chart Typologies

Excel, Many Eyes, Google Charts

Visual Analysis Grammars

VizQL, ggplot2

Visualization Grammars

Protovis, D3.js

Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Canvas, OpenGL, Processing

Break Time!

Administrivia

A2 Peer Reviews

On Friday 1/31 you will be assigned two peer A2 submissions to review. For each:

- Try to determine which is earnest and which is deceptive
- Share a rationale for how you made this determination
- Share feedback using the “I Like / I Wish / What If” rubric

Assigned reviews will be posted on the A2 Peer Review page on Canvas, along with a link to a Google Form. You should submit two forms: one for each A2 peer review.

Due by **Wed 2/4 11:59pm**.

Tutorial on Thursday

D3.js Deep Dive: Thursday 1/30 during lecture, Led by Lisa and Jiawen

- Be sure to read the D3, Part 1 notebook ahead of time. We'll work through Part 2 in class. Also read the JS/Observable primer if you're new to this!

Web Publishing: Friday 2/7 4-5:30pm in Zoom, Led by Heer

A Visualization Tool Stack

Chart Typologies

Excel, Many Eyes, Google Charts

Visual Analysis Grammars

VizQL, ggplot2

Visualization Grammars

Protovis, D3.js

Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Canvas, OpenGL, Processing

Chart Typologies

Excel, Many Eyes, Google Charts

Charting Tools

Visual Analysis Grammars

VizQL, ggplot2

Visualization Grammars

Protovis, D3.js

Declarative Languages

Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Canvas, OpenGL, Processing

Programming Toolkits

Chart Typologies

Excel, Many Eyes, Google Charts

Charting Tools

Visual Analysis Grammars

VizQL, ggplot2

Visualization Grammars

Protovis, D3.js

Declarative Languages

Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Canvas, OpenGL, Processing

Programming Toolkits

What is a Declarative Language?

Programming by describing ***what***, not ***how***

Separate specification (*what you want*) from execution (*how it should be computed*)

In contrast to imperative programming, where you must give explicit steps.

What is a Declarative Language?

Programming by describing ***what***, not ***how***

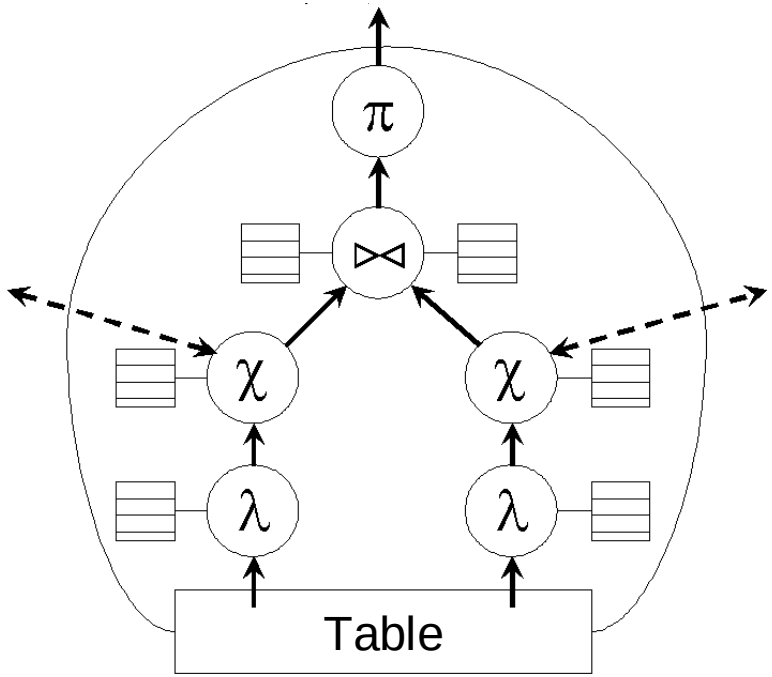
Separate specification (*what you want*) from execution (*how it should be computed*)

In contrast to imperative programming, where you must give explicit steps.

```
d3.selectAll("rect")
  .data(my_data)
 .join("rect")
  .attr("x", d => xscale(d.foo))
  .attr("y", d => yscale(d.bar))
```


```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
 "http://www.w3.org/TR/html4/loose.dtd">
<!--[if IE]><![endif]-->
▼<html>
  ▶<head>…</head>
  ▼<body id="home" style="visibility: visible; ">
    <script src="http://connect.facebook.net/en_US/all.js"></scrip
    <div id="fb-root"></div>
    <a name="top"></a>
    ▼<div id="shell">
      ▶<ul id="memberTools">…</ul>
        <!-- ADXINFO classification="text_ad" campaign="nyt2010-circ-
      ▶<div class="tabsContainer">…</div>
        <!-- close .tabsContainer -->
      ▶<div id="page" class="tabContent active">…</div>
        <!--close page -->
      </div>
      <!--close shell -->
    ▶<script type="text/javascript" language="JavaScript">…</script
      <img src="/adx/bin/clientside/
      1b1a1755Q2F,.tWdC,KiQ2BCQ5C6iSM.KCYQ2FMadYYQ24xN!Q2B!xQ24WN4aN
      <script src="http://graphics8.nytimes.com/js/test/wto/wt_capi.
      charset="utf-8"></script>
      <span id="wt_script"></span>
    ▶<script type="text/javascript">…</script>
      <img src="http://graphics8.nytimes.com/ads/blank.gif">
      <script type="text/javascript" src="http://graphics8.nytimes.c
```

HTML / CSS




```
SELECT customer_id, customer_name,
    COUNT(order_id) as total
FROM customers
INNER JOIN orders ON
    customers.customer_id
    = orders.customer_id
GROUP BY customer_id, customer_name
HAVING COUNT(order_id) > 5
ORDER BY COUNT(order_id) DESC
```

SQL

Why Declarative Languages?

Faster iteration, less code, larger user base?

Better visualization. *Smart defaults.*

Reuse. *Write-once, then re-apply.*

Performance. *Optimization, scalability.*

Portability. *Multiple devices, renderers, inputs.*

Programmatic generation.
Write programs which output visualizations.

Automated search & recommendation.

Chart Typologies

Excel, Many Eyes, Google Charts

Charting Tools

Visual Analysis Grammars

VizQL, ggplot2

Visualization Grammars

Protovis, D3.js

Declarative Languages

Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Processing, OpenGL, Java2D

Programming Toolkits

Chart Typologies

Excel, Many Eyes, Google Charts

Charting Tools

Visual Analysis Grammars

VizQL, ggplot2, ***Vega-Lite***

Visualization Grammars

Protovis, D3.js, ***Vega***

Declarative Languages

Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Processing, OpenGL, Java2D

Programming Toolkits

Chart Typologies

Excel, Many Eyes, Google Charts

?!

Charting Tools

Visual Analysis Grammars

VizQL, ggplot2, ***Vega-Lite***

Visualization Grammars

Protovis, D3.js, ***Vega***

Declarative Languages

Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Processing, OpenGL, Java2D

Programming Toolkits

Visual Analysis Grammars

VizQL, ggplot2, ***Vega-Lite***

Visualization Grammars

Protovis, D3.js, ***Vega***

Declarative Languages

Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Processing, OpenGL, Java2D

Programming Toolkits

Interactive Data Exploration

Tableau, ***Lyra, Voyager***

Graphical Interfaces

Visual Analysis Grammars

VizQL, ggplot2, ***Vega-Lite***

Visualization Grammars

Protovis, D3.js, ***Vega***

Declarative Languages

Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Processing, OpenGL, Java2D

Programming Toolkits

UW Interactive Data Lab PROJECTS

The Lyra Visualization Design Environment (VDE) alpha

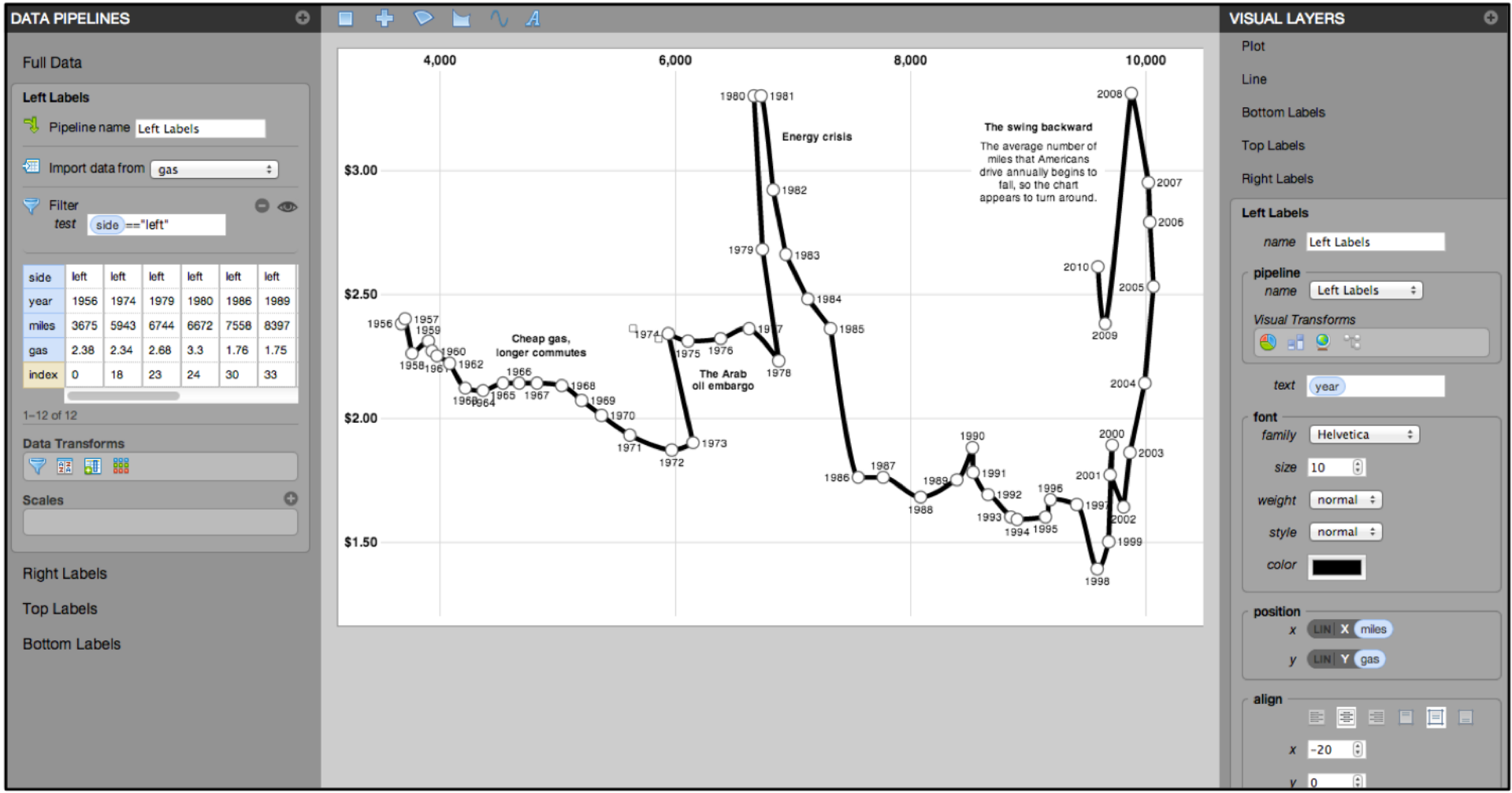
Arvind Satyanarayan, Kanit "Ham" Wongsuphasawat, Jeffrey Heer

William Playfair's classic chart comparing the price of wheat and wages in England recreated in the Lyra VDE.

idl.cs.washington.edu/projects/lyra

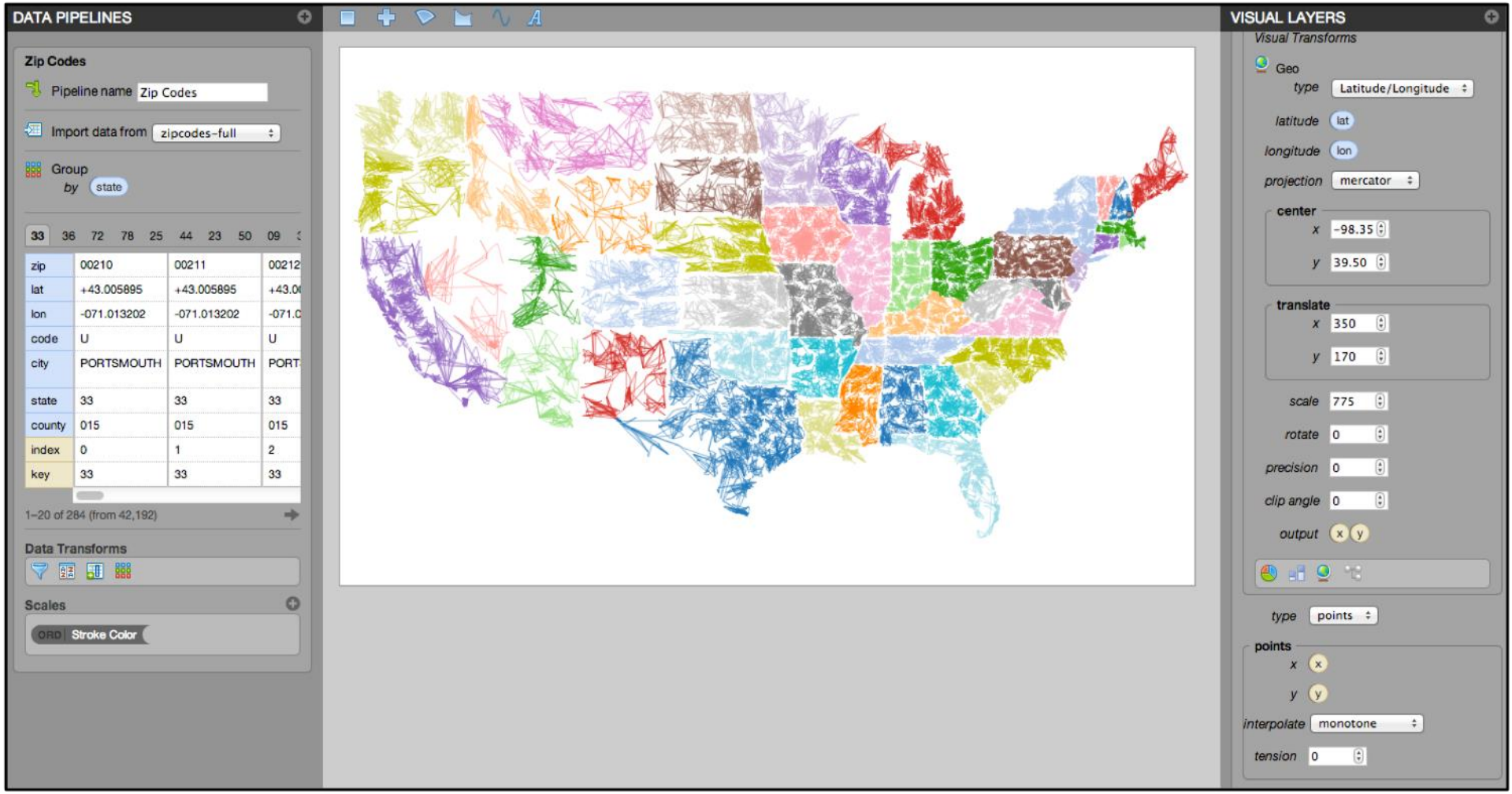
See also: Charticulator, Data Illustrator

Lyra A Visualization Design Environment



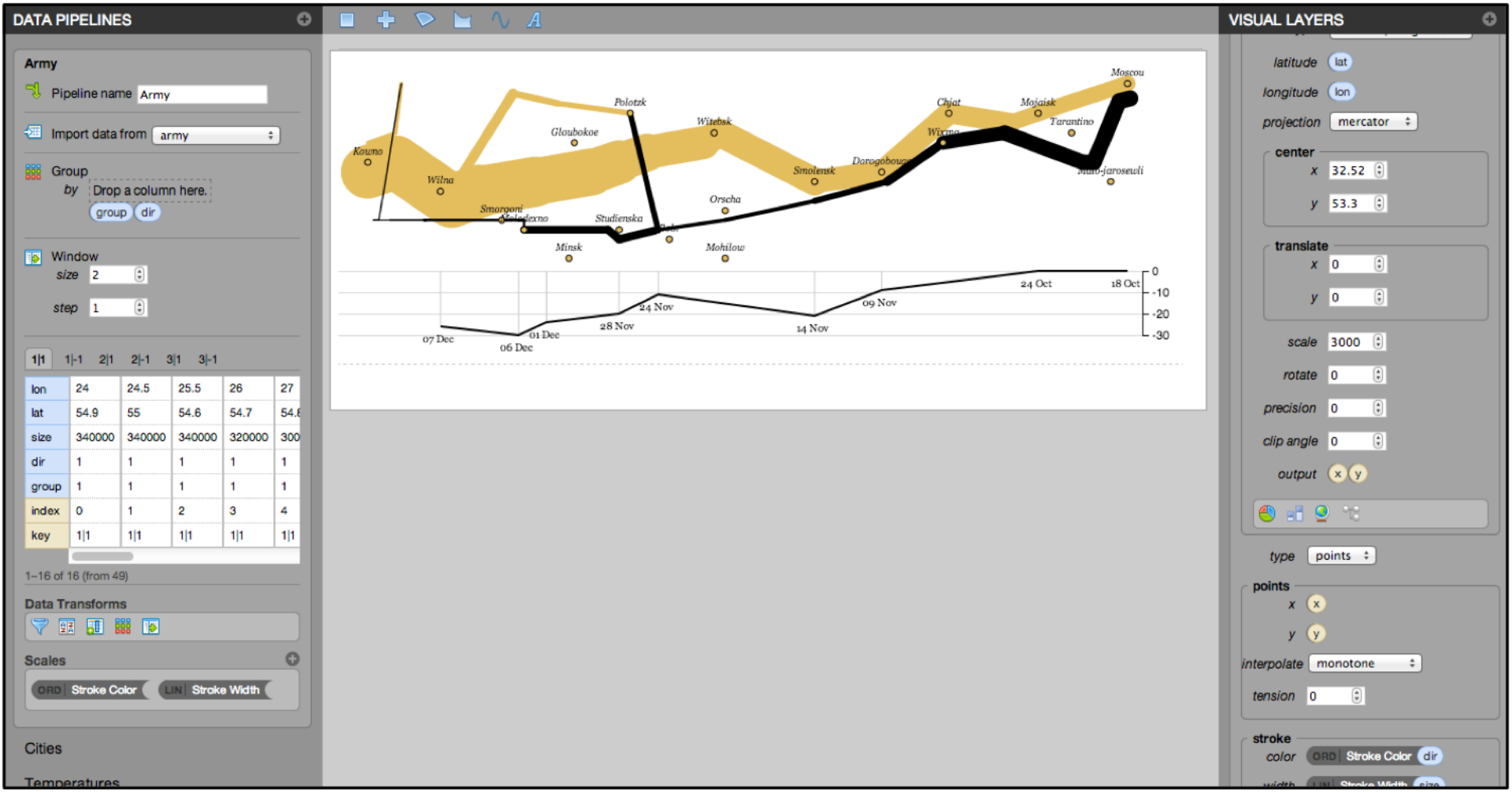
Driving Shifts into Reverse by Hannah Fairfield, NYTimes

Lyra A Visualization Design Environment

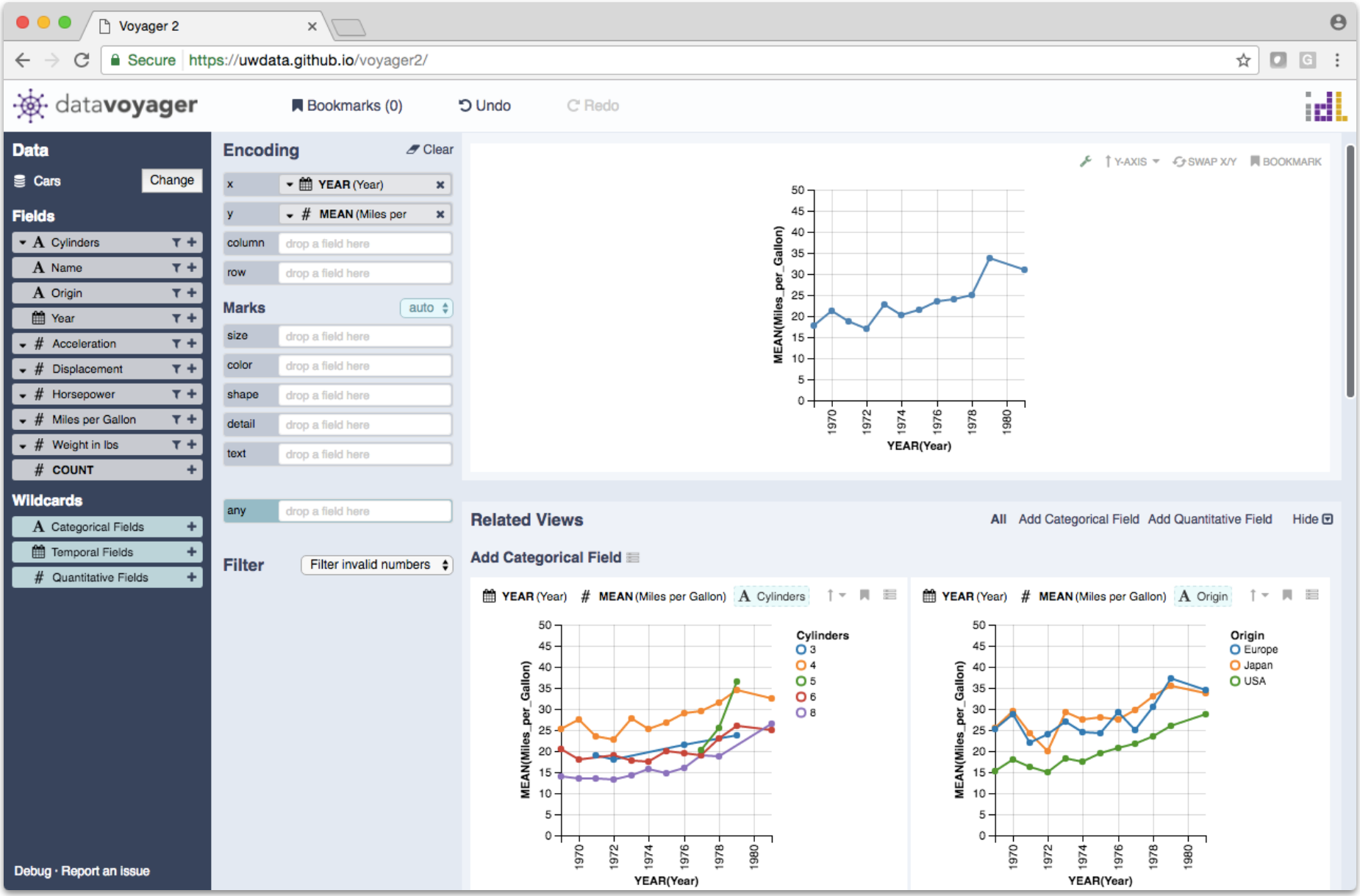


ZipScribble by Robert Kosara

Lyra A Visualization Design Environment



Napoleon's March by Charles Minard



Voyager. Wongsuphasawat et al. *InfoVis'15, CHI'17*

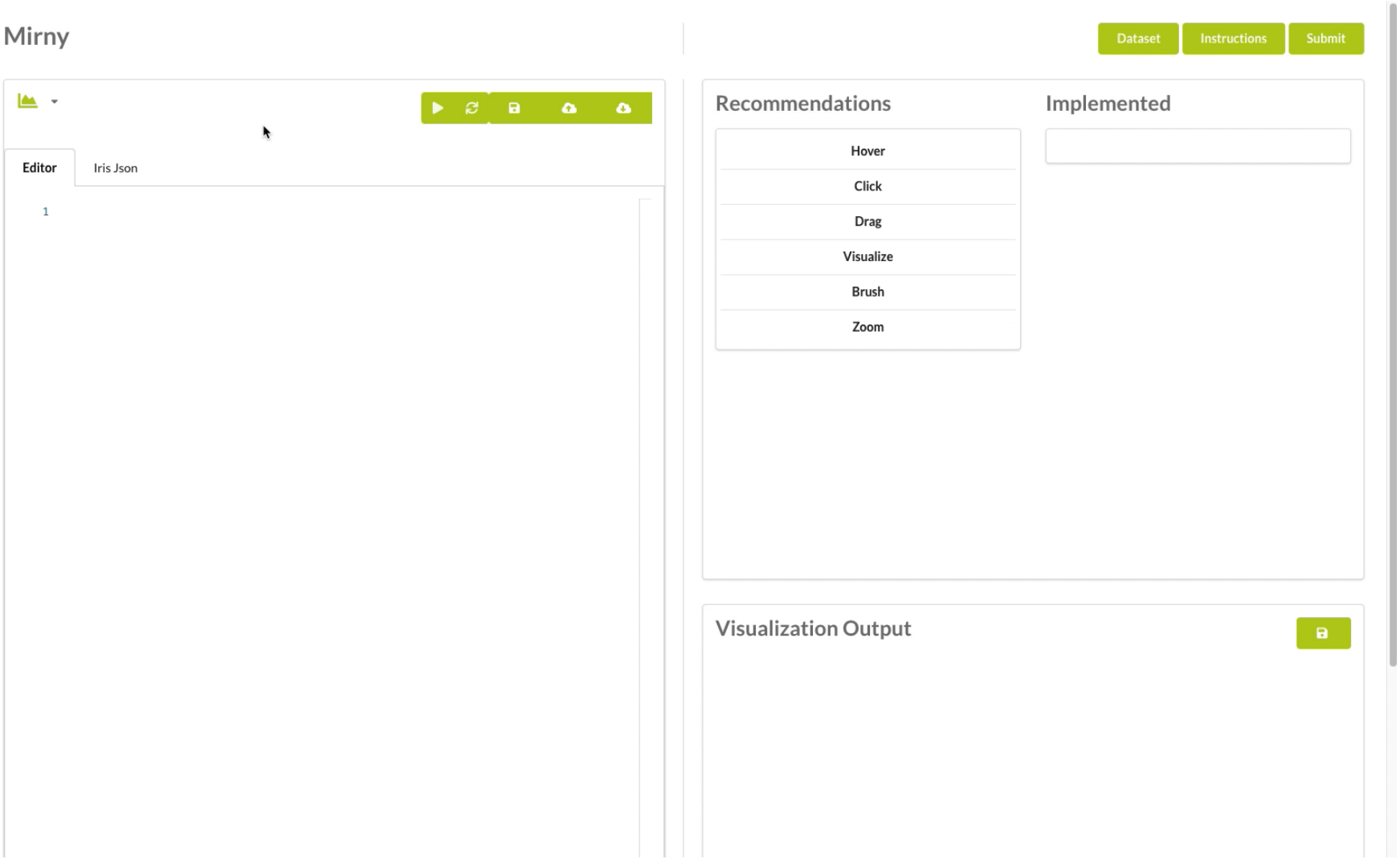
Key Idea: Augment manual exploration with visualization recommendations sensitive to the user's current focus.

The goal is to support *systematic consideration* of the data, without exacerbating *false discovery*.

To model a user's search frontier, we *enumerate related Vega-Lite specifications*, seeded by the user's current focus.

Candidate charts are pruned and ranked using models of estimated *perceptual effectiveness*.

Voyager. Wongsuphasawat et al. *InfoVis'15, CHI'17*



Mirny. Bako et al. *IUI '23*

Summary

There is no one-size-fits-all tool for visualization.

Instead, visualization tools fall along a spectrum ranging from graphical interfaces to advanced programming toolkits.

Visualization tools make deliberate tradeoffs between ease of use and expressiveness, placing them at specific points along the spectrum.

Users often select and switch between various tools to meet their current needs.