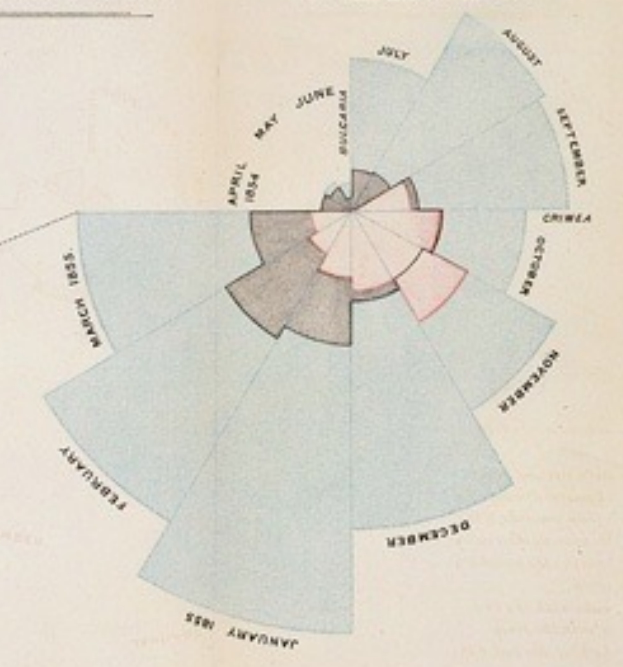
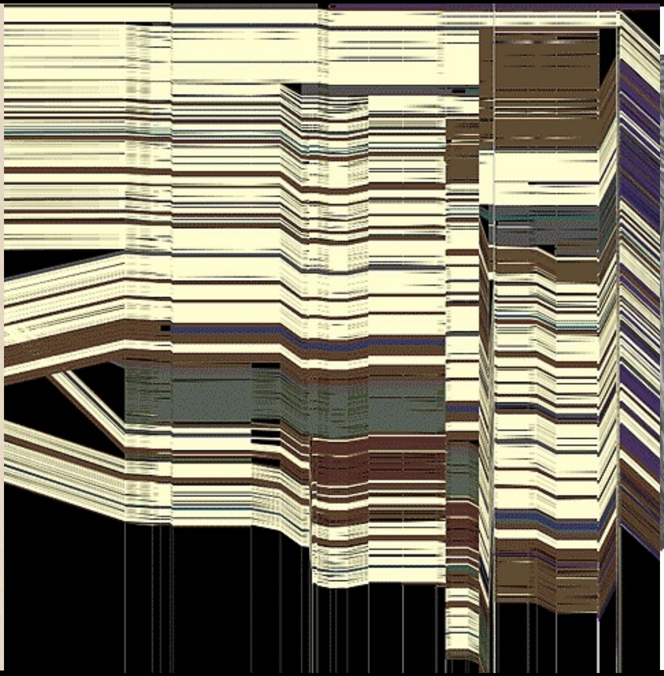
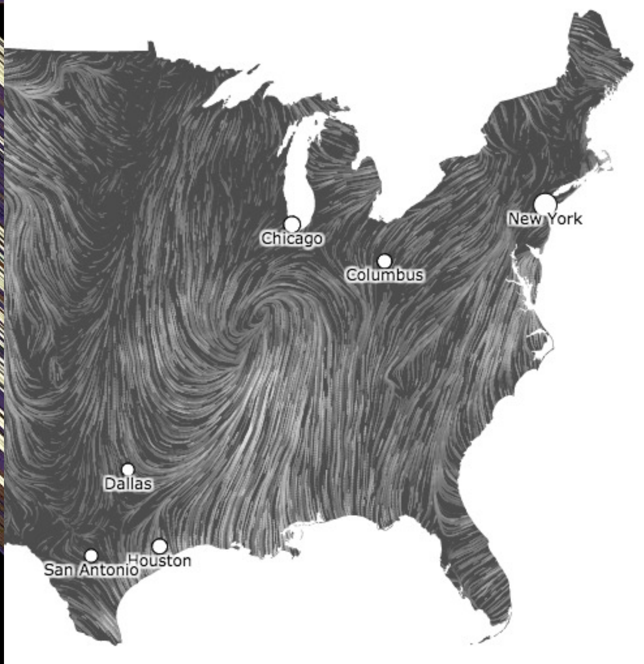
CSE 442 - Data Visualization

Visualization Tools







Leilani Battle University of Washington

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?

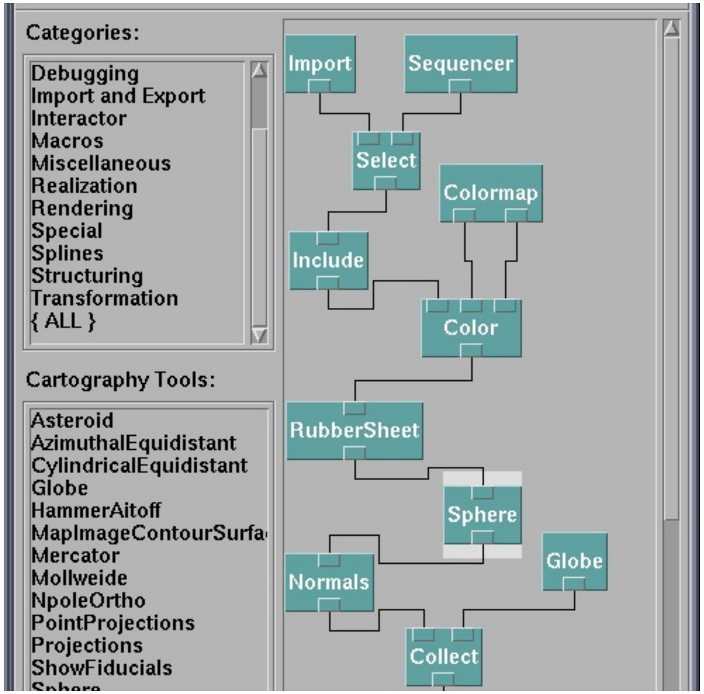


Chart Typology

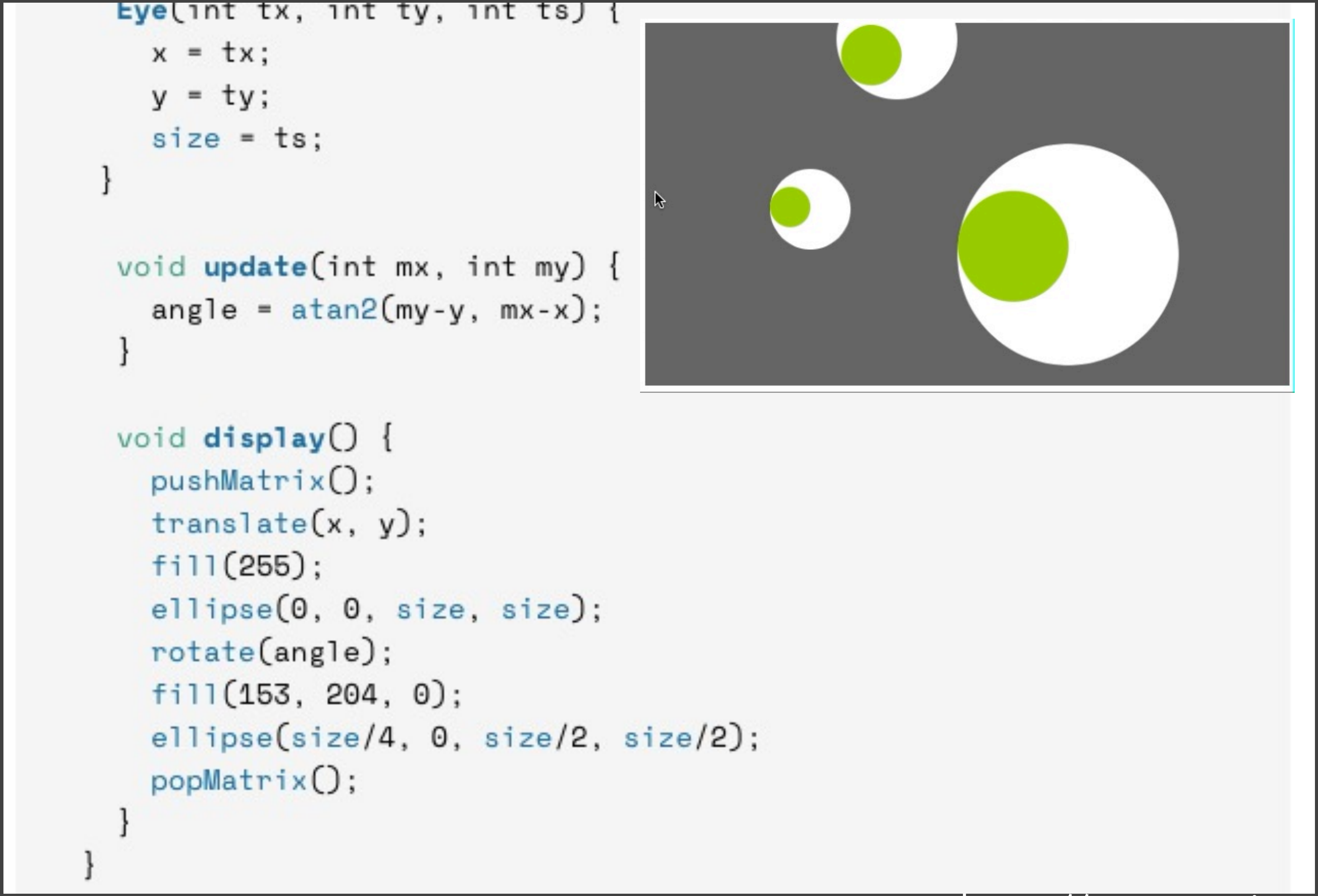
Pick from a stock of templates
Easy-to-use but limited expressiveness
Prohibits novel designs, new data types

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

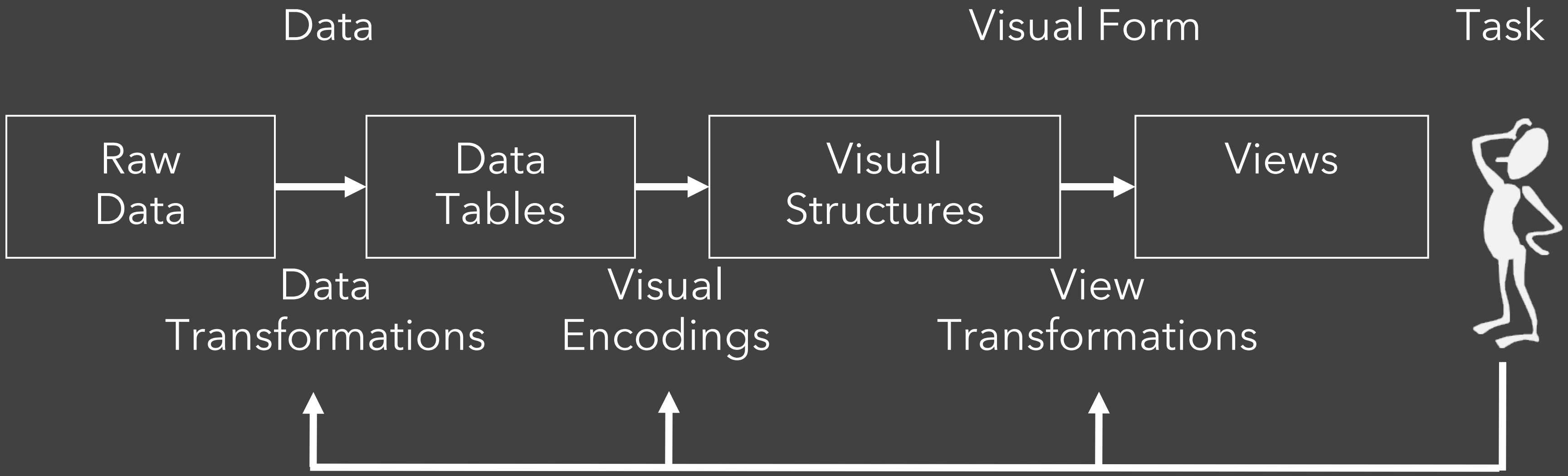
Canvas, OpenGL, Processing

Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Canvas, OpenGL, Processing



Data

Visual Form

Task

Raw Data

Data Tables

Visual Structures

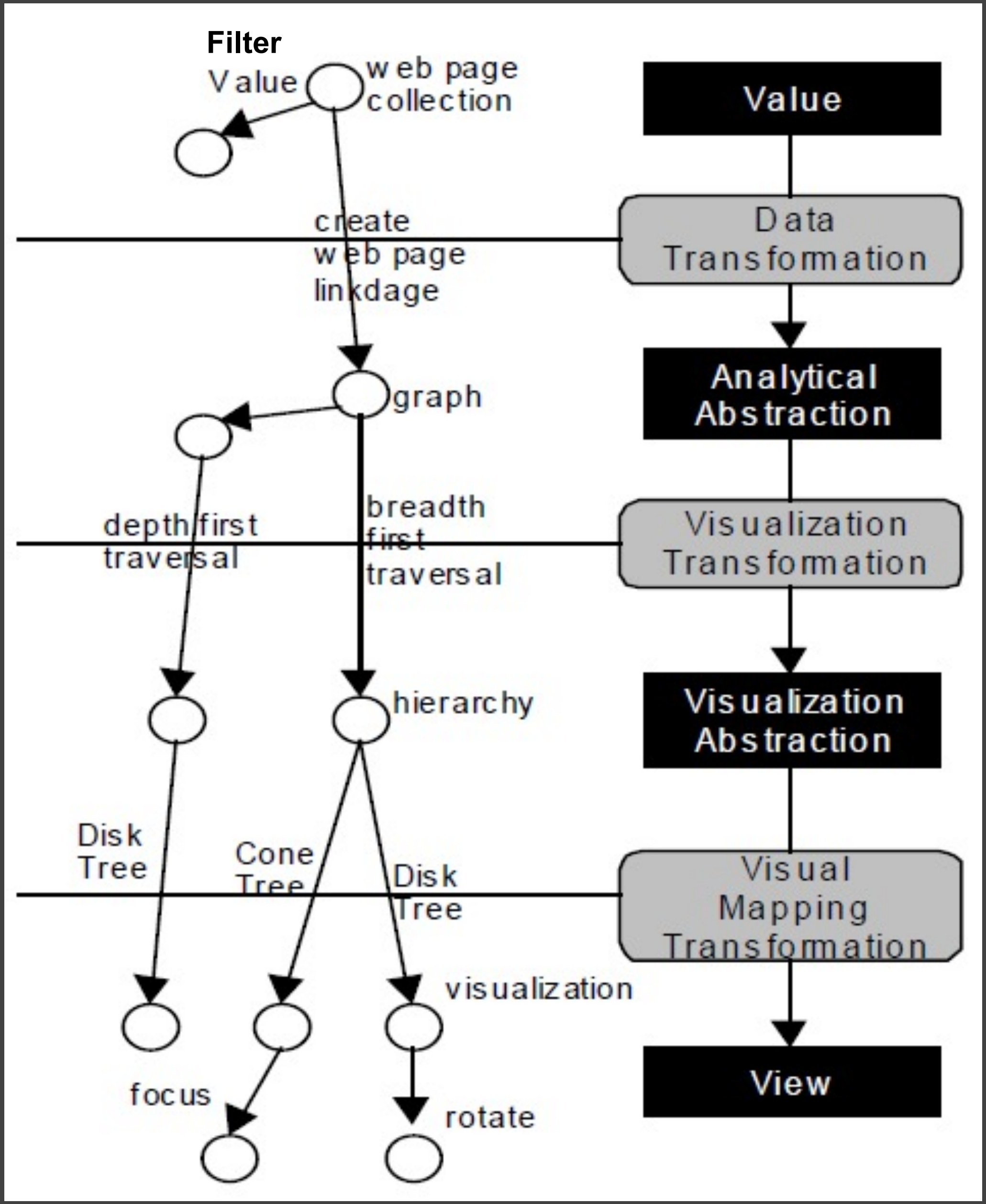
Views

Data Transformations

Visual Encodings

View Transformations

Data State Model
[Chi 98]



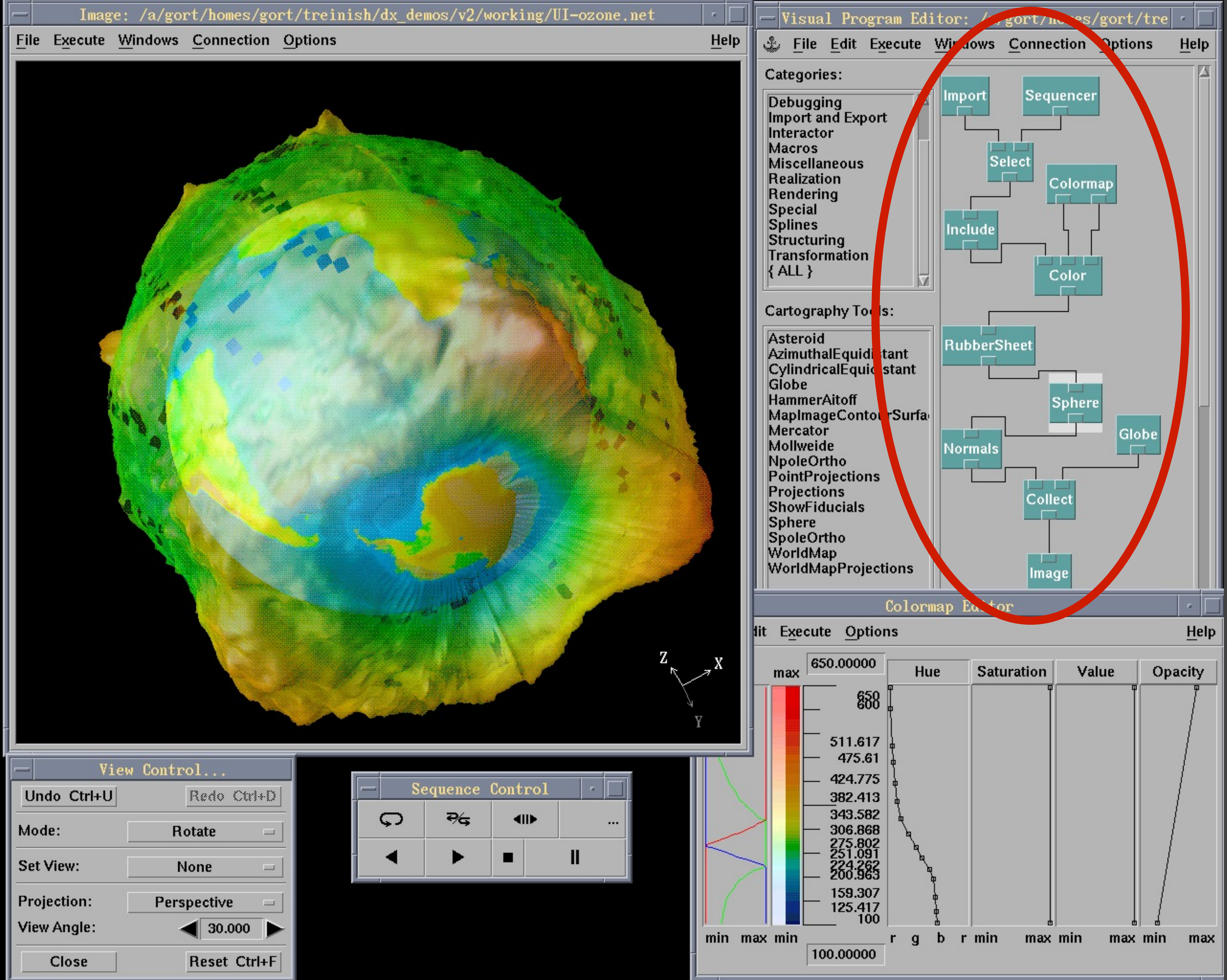


Image: /a/gort/homes/gort/treinish/dx_demos/v2/working/UI-ozone.net

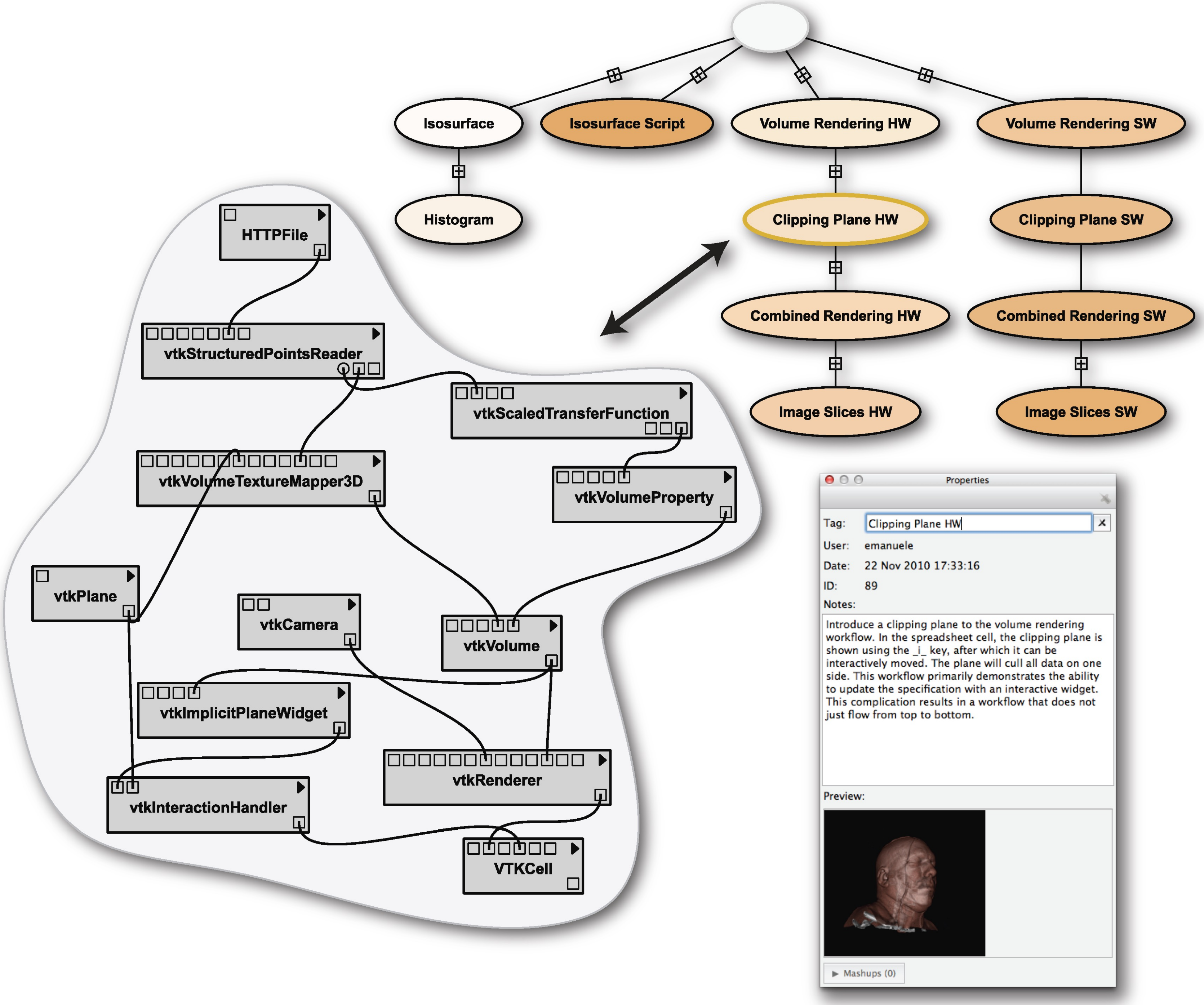
Visual Program Editor: /a/gort/homes/gort/tre

Mode: Rotate

Set View: None

Projection: Perspective

View Angle: 30.000



Tag: Clipping Plane HW

User: emanuele

Date: 22 Nov 2010 17:33:16

ID: 89

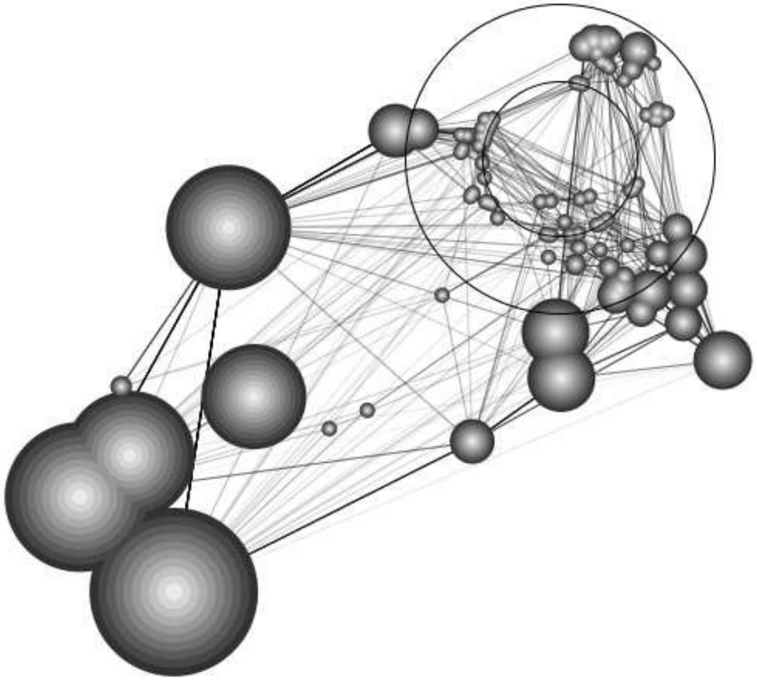
Notes:

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.

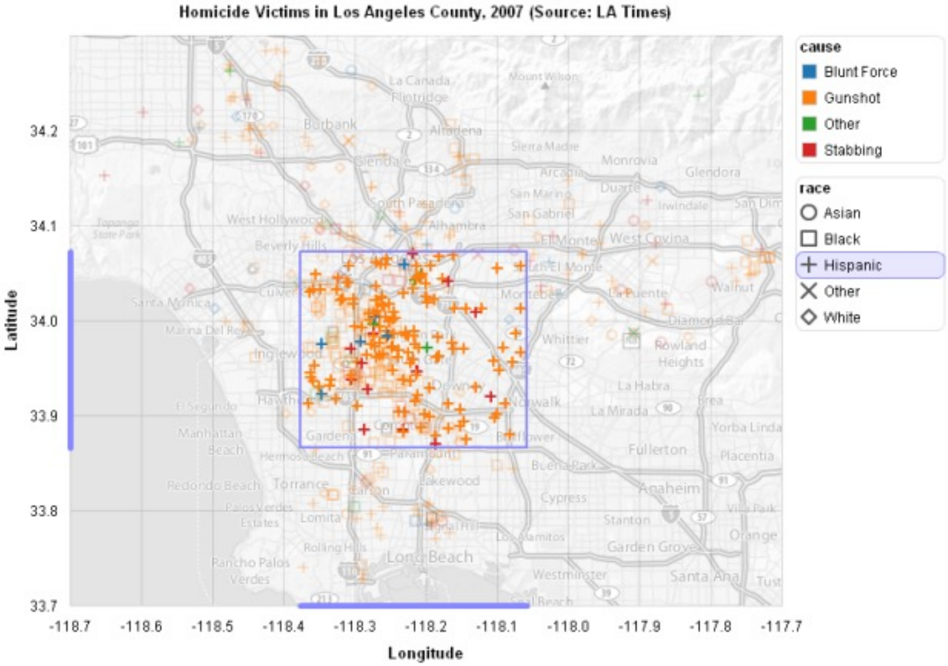
Preview:

Prefuse & Flare

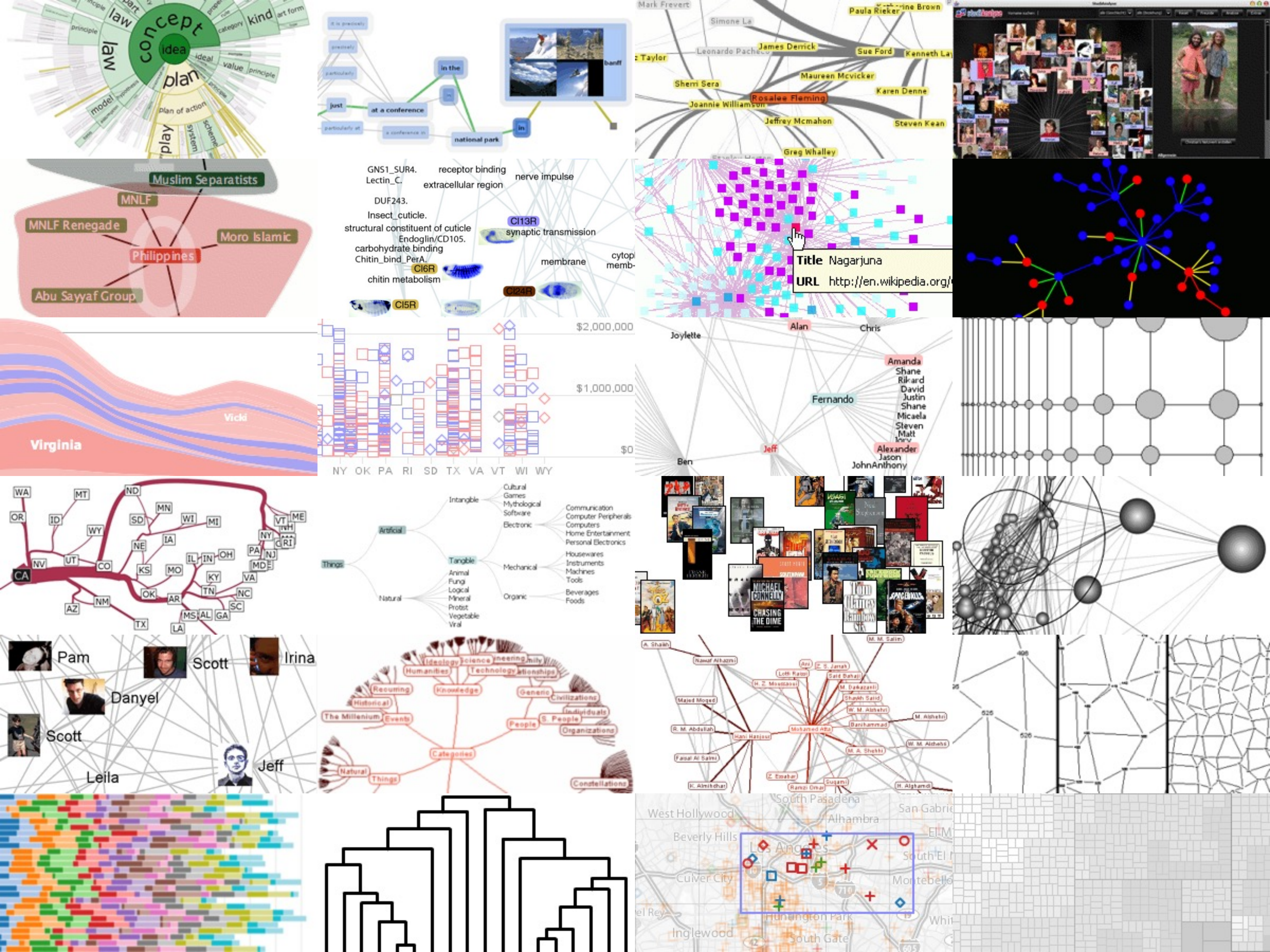
Operator-based toolkits for visualization design
Vis = (Input Data -> Visual Objects) + Operators

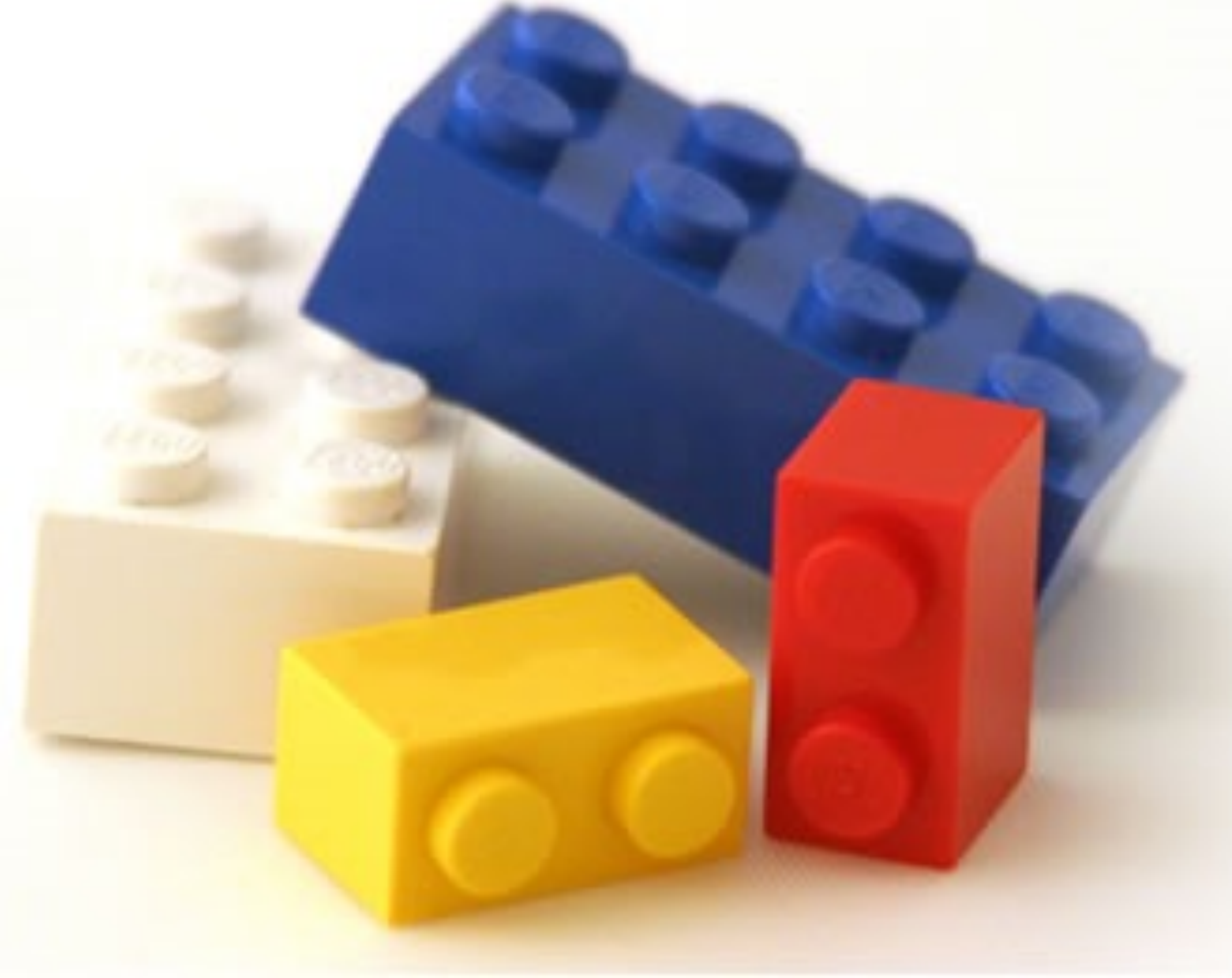


Prefuse (http://prefuse.org)



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


concept
idea
law
plan
kind
category
ideal
value
principle
model
play
system
plan of action
art form
in the
at a conference
just
national park
banff
in
particularly at
Mark Frevert
Simone La
Paula Rieker
Leonardo Pacheco
James Derrick
Sue Ford
Kenneth Lay
Taylor
Maureen Mcvicker
Sherri Sera
Rosalee Fleming
Karen Denne
Joannie Williamson
Jeffrey Mcmahon
Steven Kean
Greg Whalley
Muslim Separatists
MNLF
MNLF Renegade
Moro Islamic
Philippines
Abu Sayyaf Group
GNS1_SUR4.
Lectin_C.
receptor binding
extracellular region
nerve impulse
DUF243.
Insect_cuticle.
CI13R
synaptic transmission
structural constituent of cuticle
Endoglin/CD105.
carbohydrate binding
Chitin_bind_PerA.
CI6R
membrane
chitin metabolism
CI24R
CI5R
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
Shane
Rikard
David
Justin
Shane
Micaela
Steven
Matt
Fernando
Jeff
Alexander
Jason
JohnAnthony
Ben
WA
MT
ND
OR
ID
MN
SD
WI
MI
WY
IA
NE
NV
UT
CO
CA
KS
MO
IL
IN
OH
NY
PA
NJ
MD
VA
KY
TN
NC
OK
AR
AZ
NM
TX
MS
AL
GA
SC
LA
VT
NH
ME
RI
Things
Artificial
Intangible
Cultural
Games
Mythological
Software
Electronic
Communication
Computer Peripherals
Computers
Home Entertainment
Personal Electronics
Tangible
Mechanical
Housewares
Instruments
Machines
Tools
Natural
Animal
Fungi
Logical
Mineral
Protist
Vegetable
Viral
Organic
Beverages
Foods
Pam
Scott
Irina
Danyel
Scott
Jeff
Leila
Ideology
Science
Humanities
Technology
Knowledge
Recurring
Historical
The Millenium
Events
Generic
Civilizations
Individuals
People
S. People
Organizations
Categories
Natural
Things
Constellations
Mohamed Atta
Nawaf Alhazmi
Hani Hanjour
Majed Moqed
Ziad Jarrah
Marwan Al-Shehhi
Salem Alhazmi
Satam Suqami
Waleed M. Alshehri
Wail Alshehri
Abdul Aziz Al-Omari
Mohand Alshehri
Hamza Alghamdi
Ahmed Alghamdi
Ahmed Alnami
Saeed Alghamdi
Ahmed Al Haznawi
West Hollywood
South Pasadena
Alhambra
San Gabriel
Beverly Hills
Los Angeles
El Monte
South El Monte
Culver City
Montebello
Inglewood
Huntington Park
South Gate
Whittier



Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Canvas, OpenGL, Processing

Chart Typologies

Excel, Google Charts

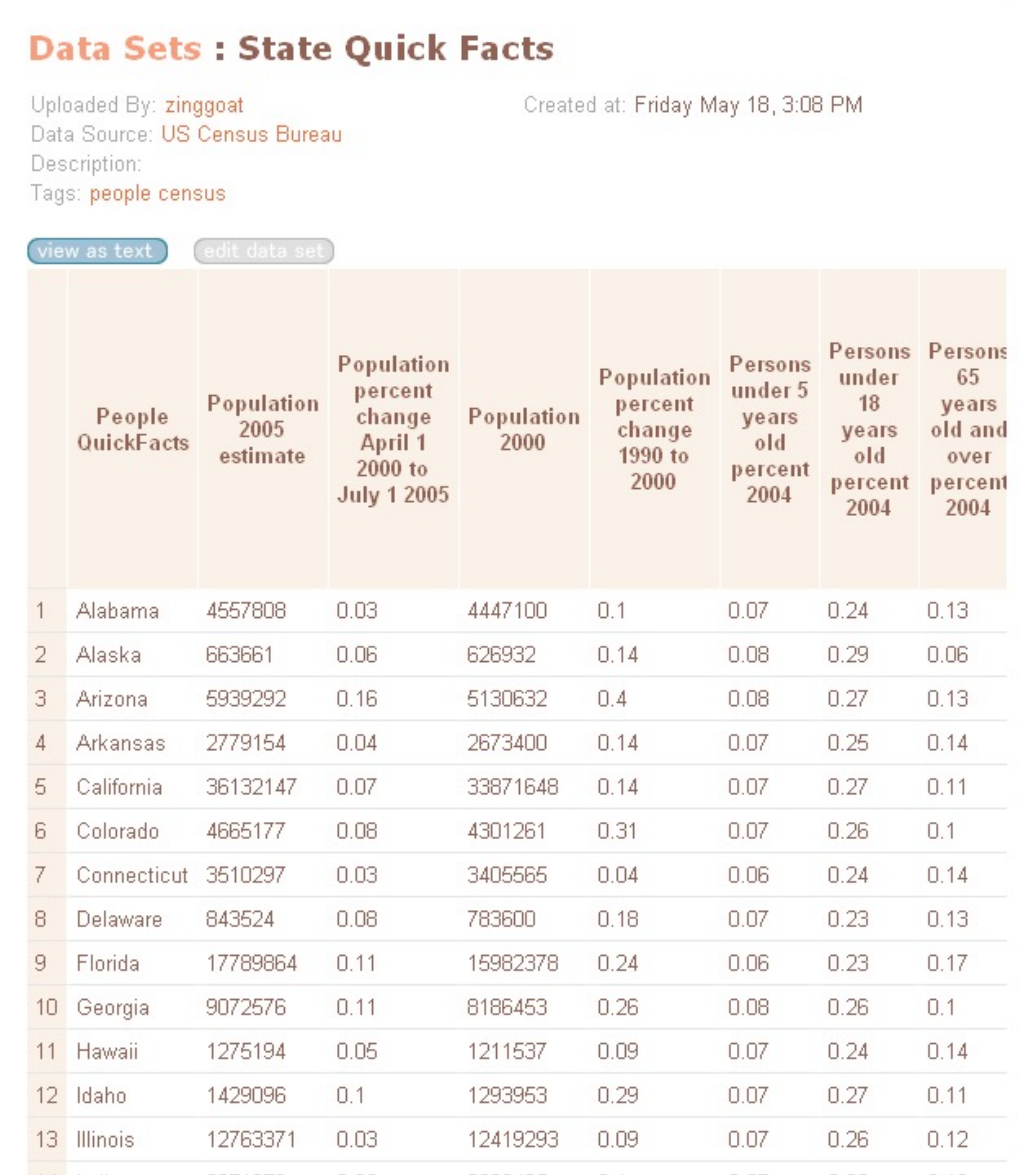
Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Canvas, OpenGL, Processing

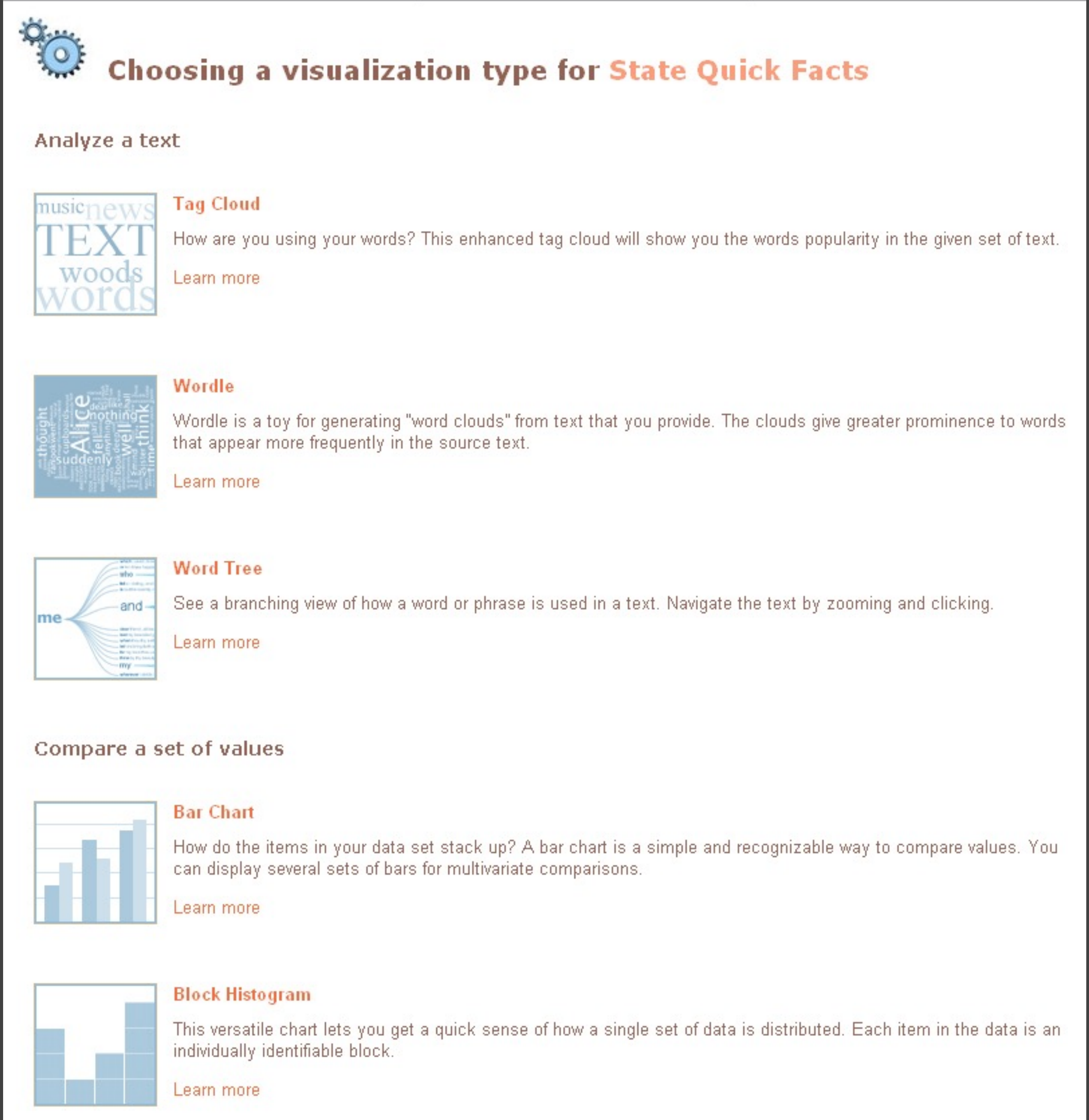
Chart Typologies



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

	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

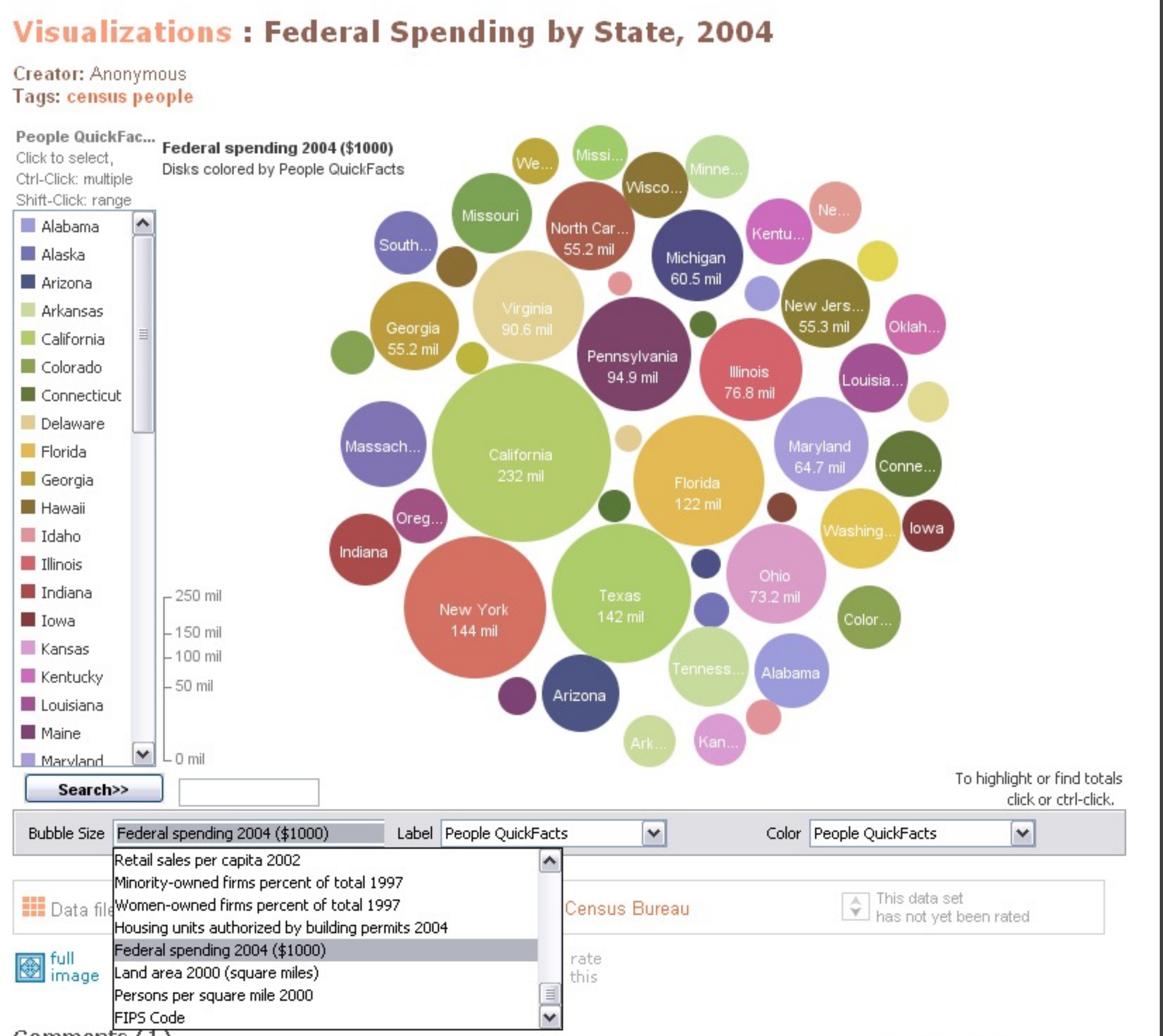
How do the items in your data set stack up? A bar chart is a simple and recognizable way to compare values. You can display several sets of bars for multivariate comparisons.

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

Bubble Size: Federal spending 2004 ($1000)
Label: People QuickFacts
Color: People QuickFacts

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. Her name is
NOUN
Hillary Clinton. Our piano is a Steinway Concert tree
CELEBRITY (FEMALE) NOUN
and it has 88 cups. It also has a soft pedal and a/an
PLURAL NOUN
Smily pedal. When I have a lesson, I sit down on the piano
ADJECTIVE
AlBERTO and play for 16 Minuts. I do scales to
NOUN PERIOD OF TIME
exercise my cats, and then I usually play a minuet by
PLURAL NOUN
Johann Sebastian washington. Teacher says I am a natural
CELEBRITY (LAST NAME)
Haunted House and have a good musical leg. Perhaps
NOUN PART OF THE BODY
when I get better I will become a concert vet and give
PROFESSION
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

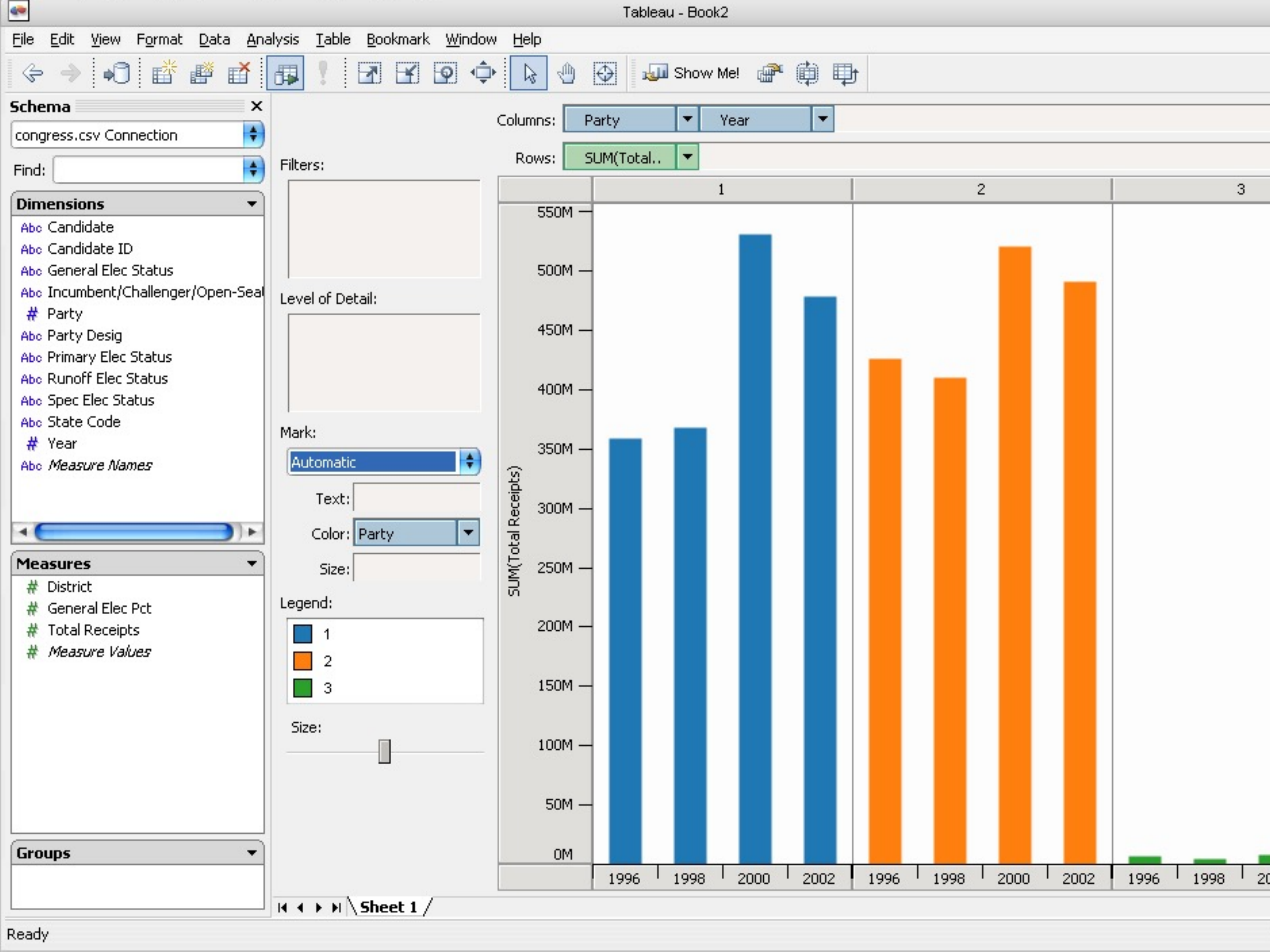
Canvas, OpenGL, Processing

Chart Typologies
Excel, Many Eyes, Google Charts

Visual Analysis Grammars
VizQL, ggplot2

Component Architectures
Prefuse, Flare, Improvise, VTK

Graphics APIs
Canvas, OpenGL, Processing



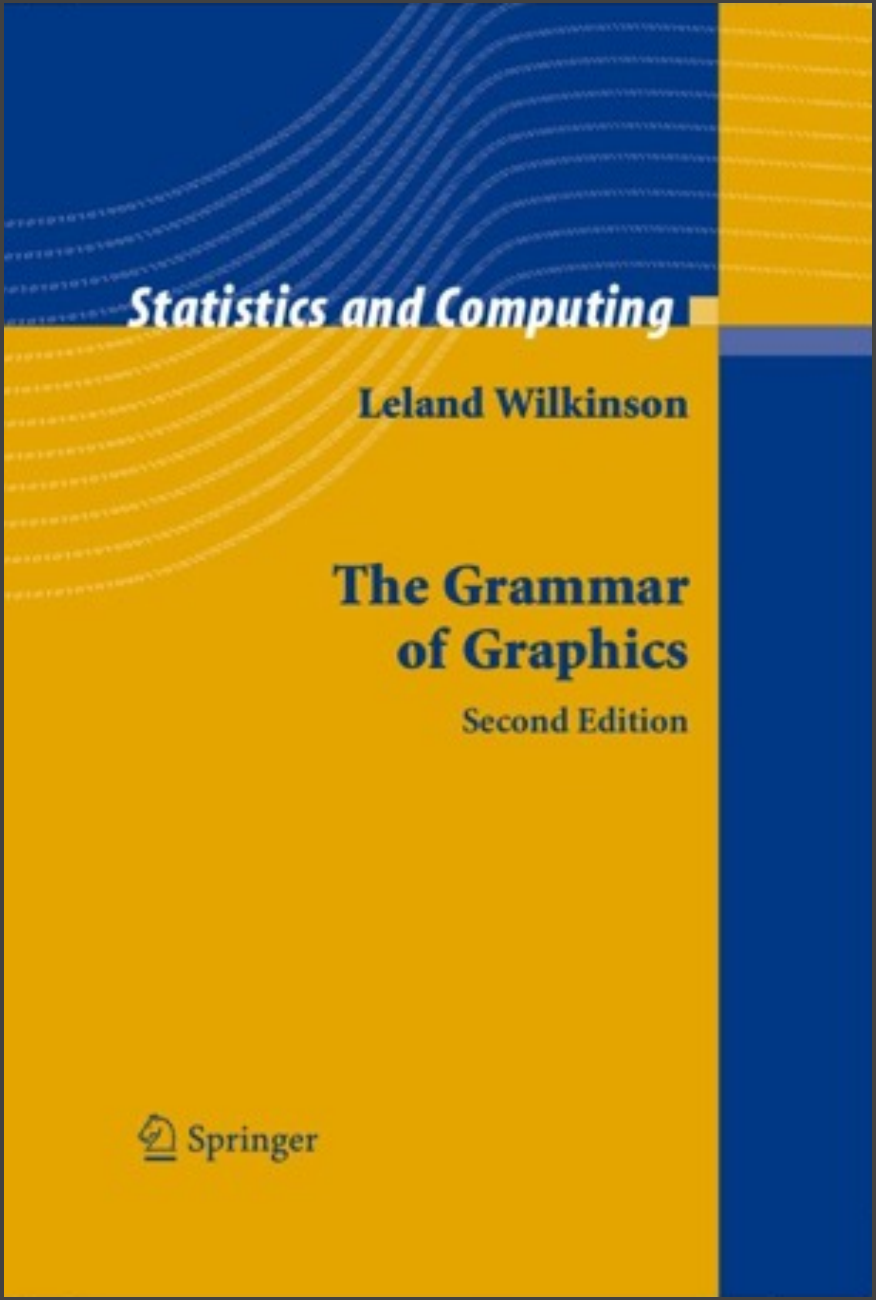
congress.csv Connection

Columns: Party Year

Rows: SUM(Total..

Mark: Automatic

Color: Party



Statistics and Computing

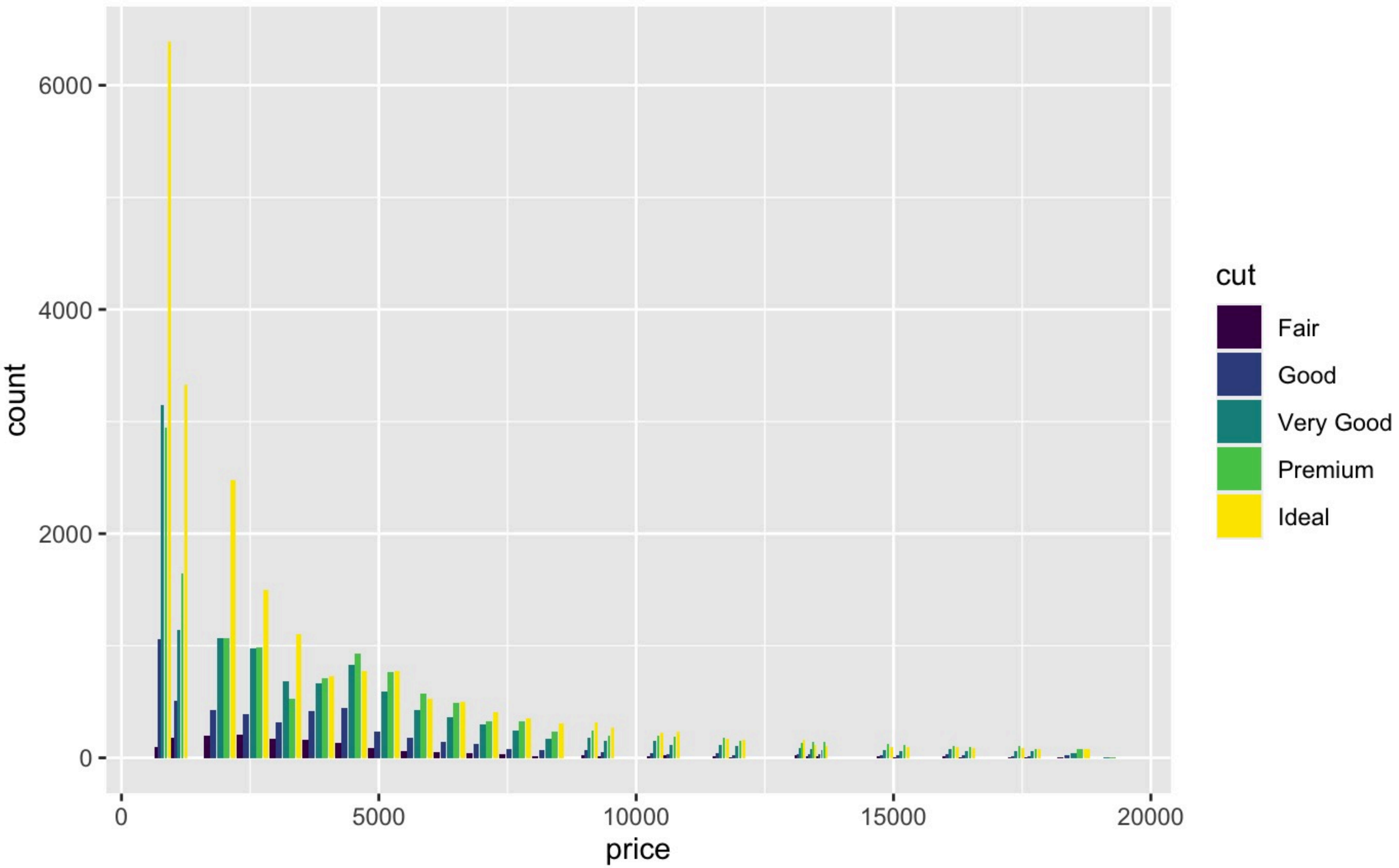
Leland Wilkinson

The Grammar of Graphics

Second Edition

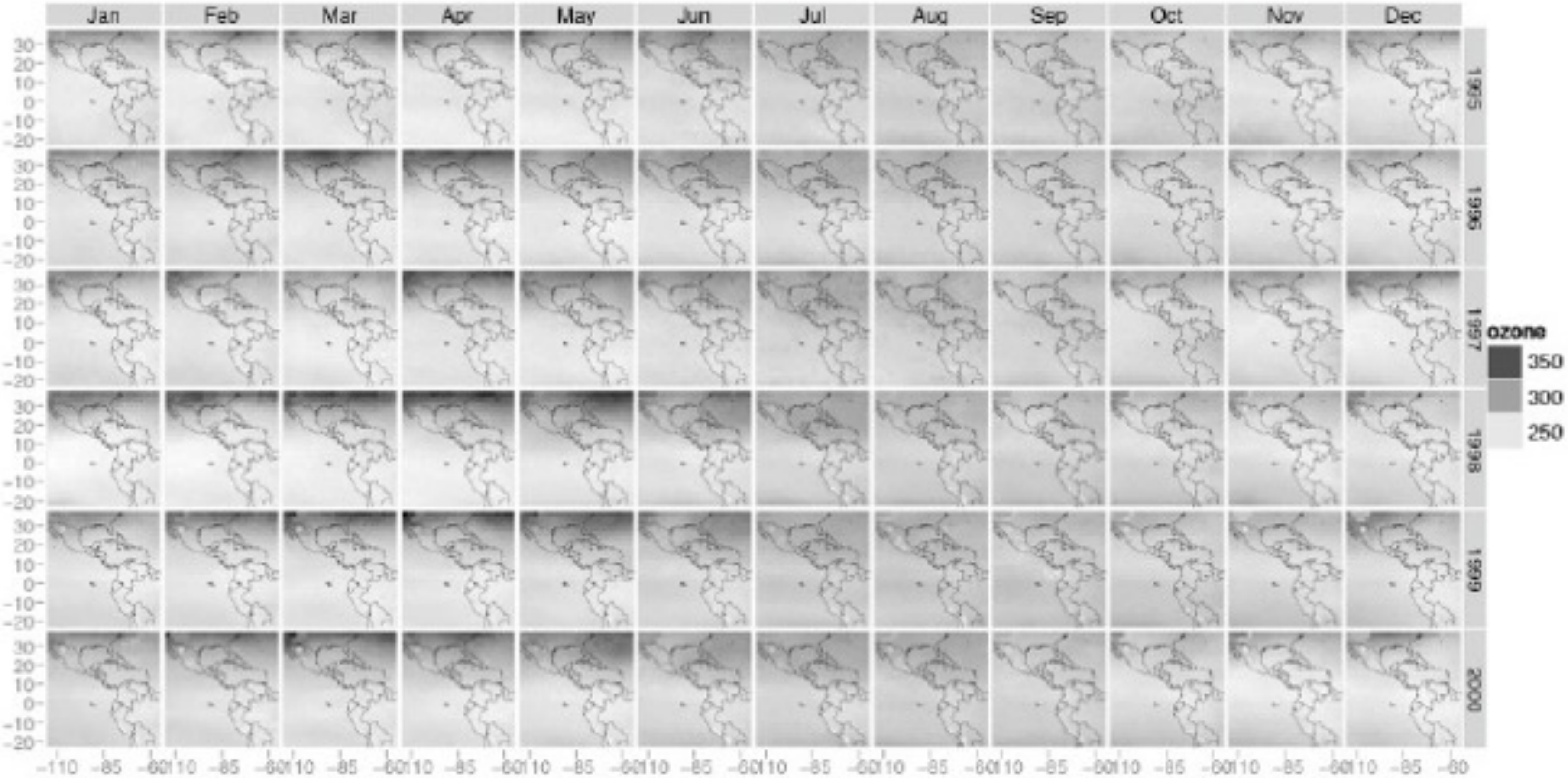
Springer

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

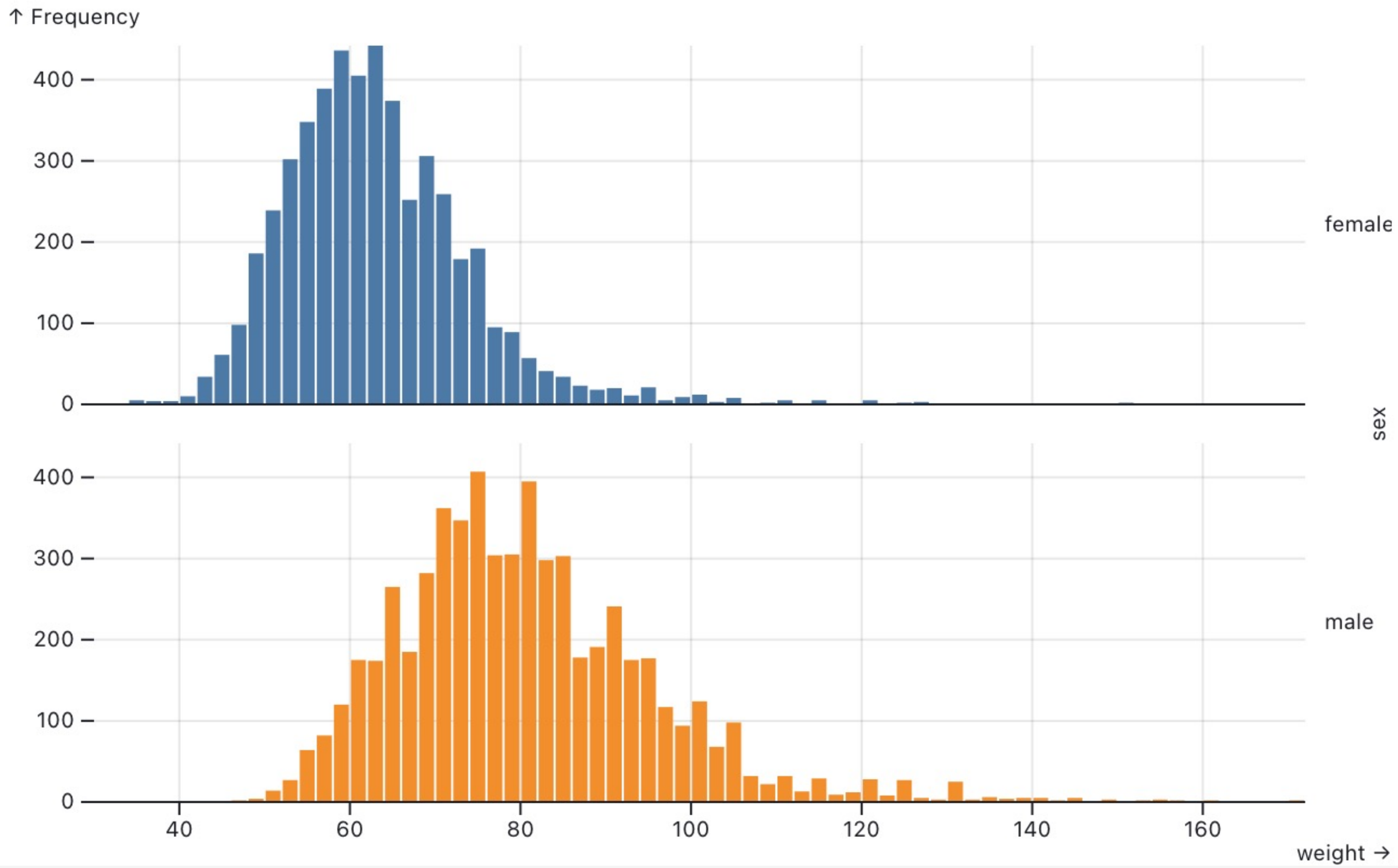


```
ggplot(diamonds, aes(x=price, fill=cut))
+ geom_bar(position="dodge")
```

ggplot2



```
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

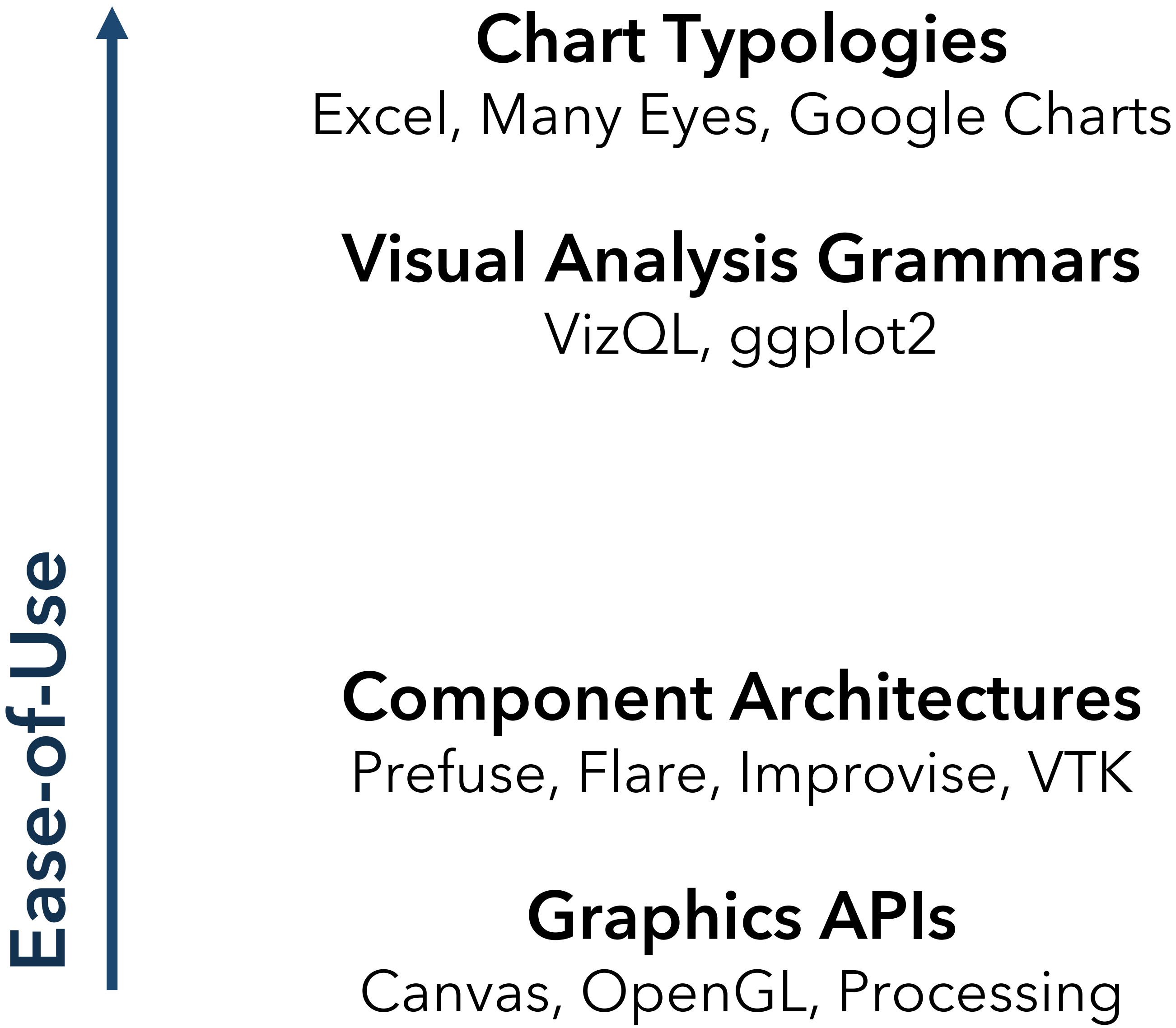


Chart Typologies
Excel, Many Eyes, Google Charts

Visual Analysis Grammars
VizQL, ggplot2

Component Architectures
Prefuse, Flare, Improvise, VTK

Graphics APIs
Canvas, OpenGL, Processing

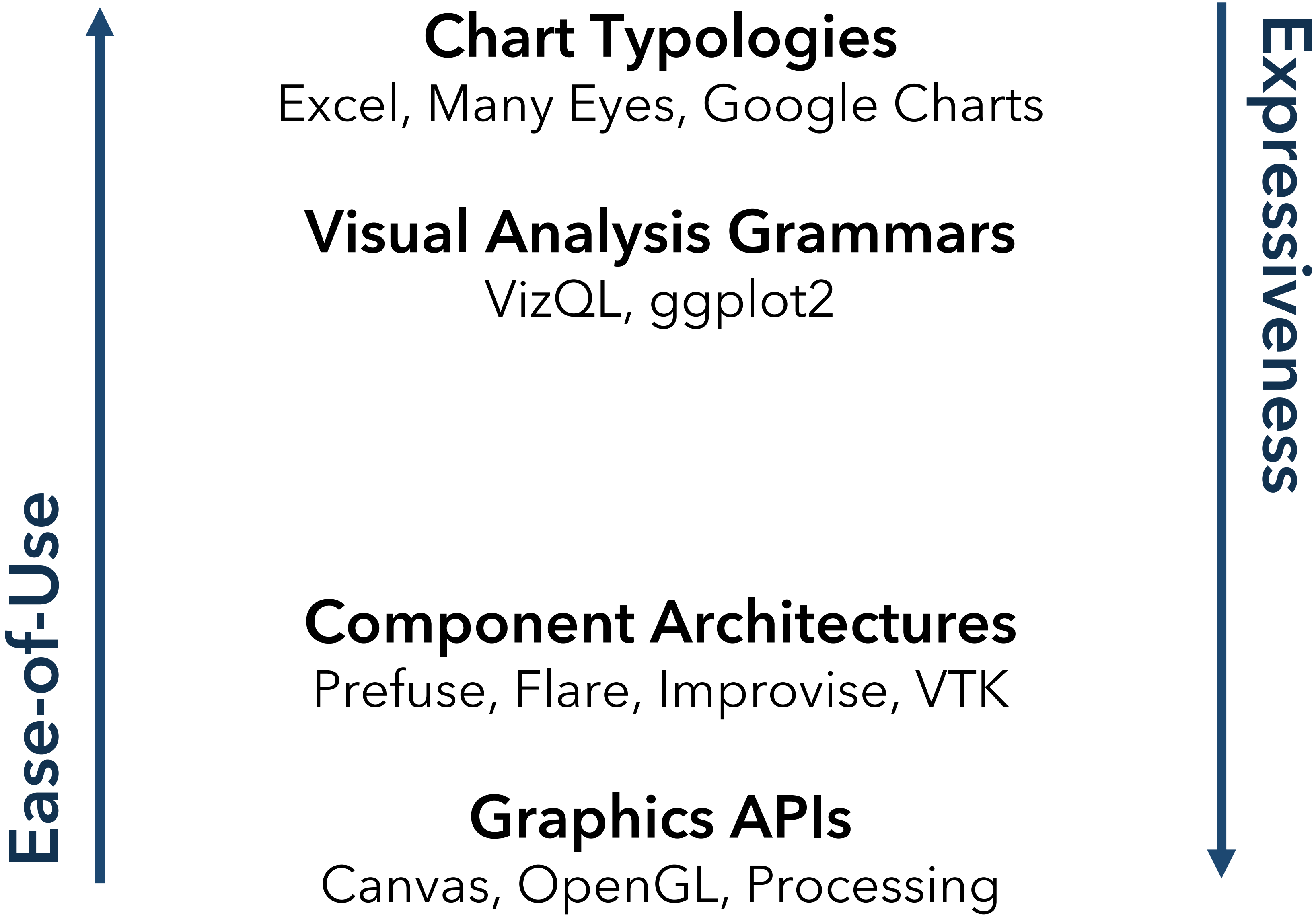


Chart Typologies
Excel, Many Eyes, Google Charts

Visual Analysis Grammars
VizQL, ggplot2

Component Architectures
Prefuse, Flare, Improvise, VTK

Graphics APIs
Canvas, OpenGL, Processing

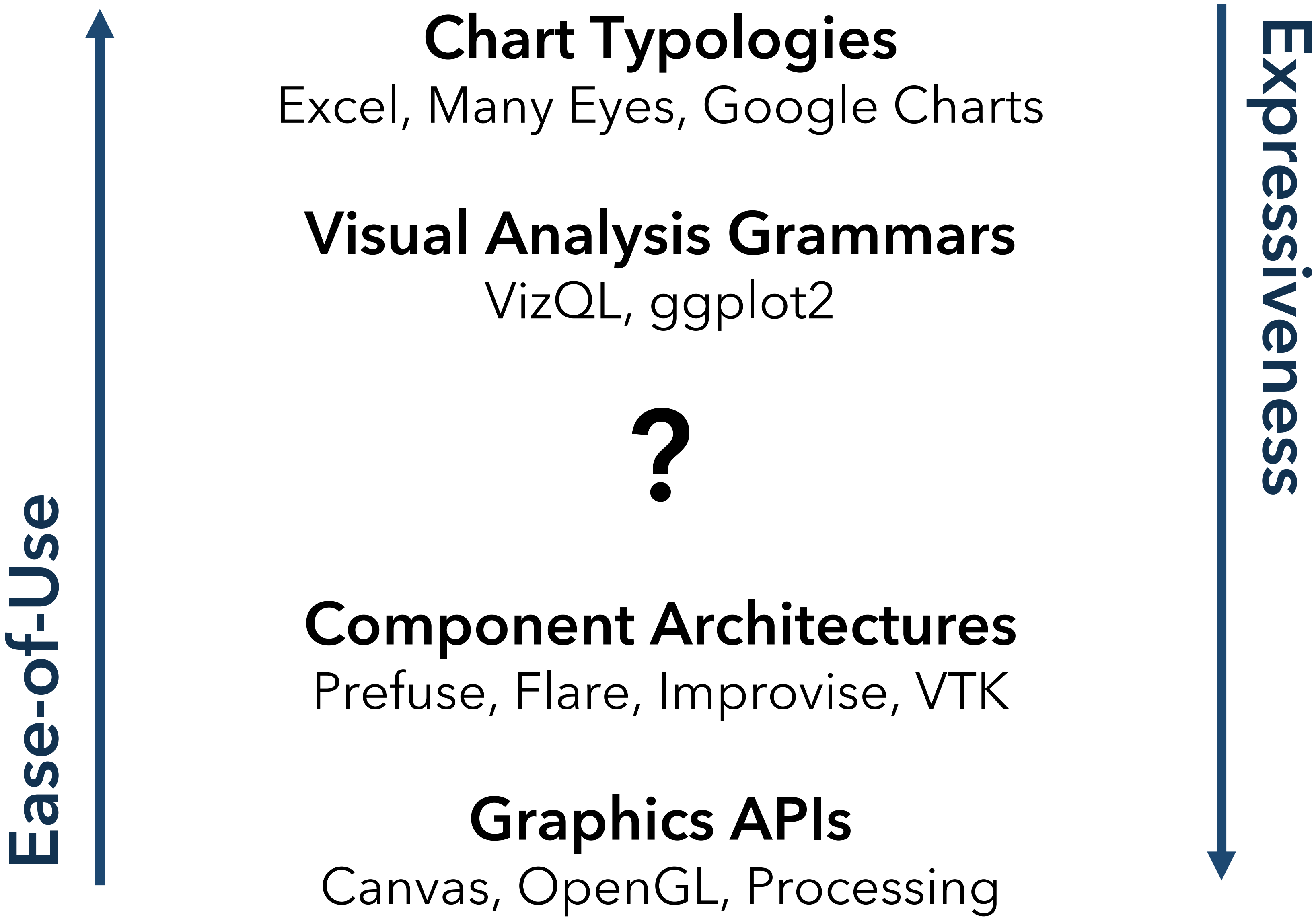


Chart Typologies
Excel, Many Eyes, Google Charts

Visual Analysis Grammars
VizQL, ggplot2

?

Component Architectures
Prefuse, Flare, Improvise, VTK

Graphics APIs
Canvas, OpenGL, Processing

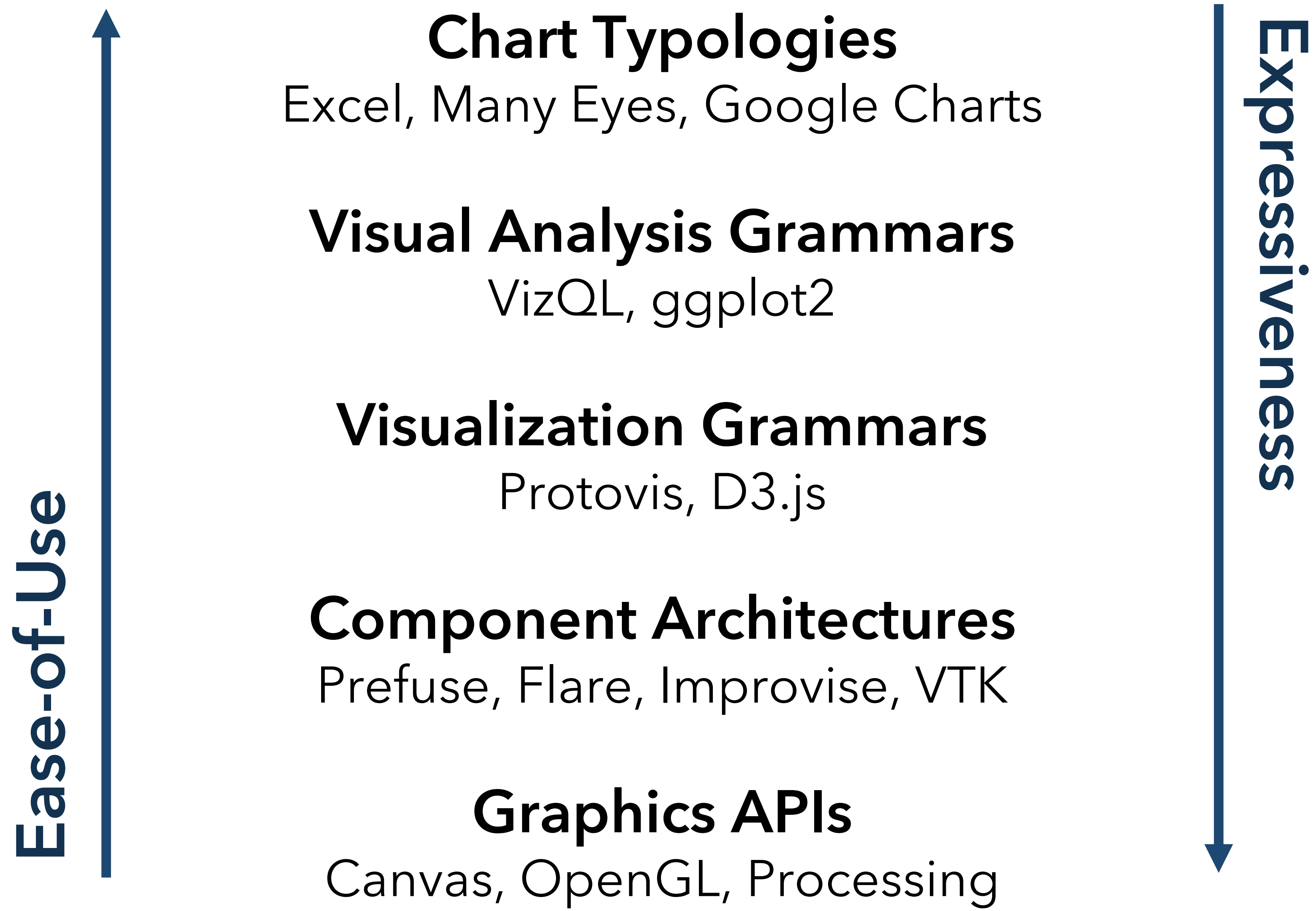


Chart Typologies
Excel, Many Eyes, Google Charts

Visual Analysis Grammars
VizQL, ggplot2

Visualization Grammars
Protovis, D3.js

Component Architectures
Prefuse, Flare, Improvise, VTK

Graphics APIs
Canvas, OpenGL, Processing

Protovis & D3

Today's first task is not to invent wholly new [*graphical*] techniques, though these are needed. Rather we need most vitally to recognize and reorganize the **essential of old techniques**, to **make easy their assembly in new ways**, and to **modify their external appearances to fit the new opportunities**.

J. W. Tukey, M. B. Wilk
Data Analysis & Statistics, 1965

Visualization Grammar

Visualization Grammar

Data	Input data to visualize

Visualization Grammar

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

Visualization Grammar

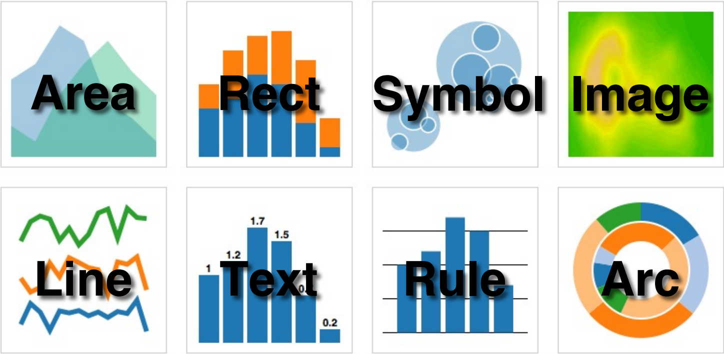
Data	Input data to visualize
Transforms	Group, aggregate, stats, layout
Scales	Map data values to visual values

Visualization Grammar

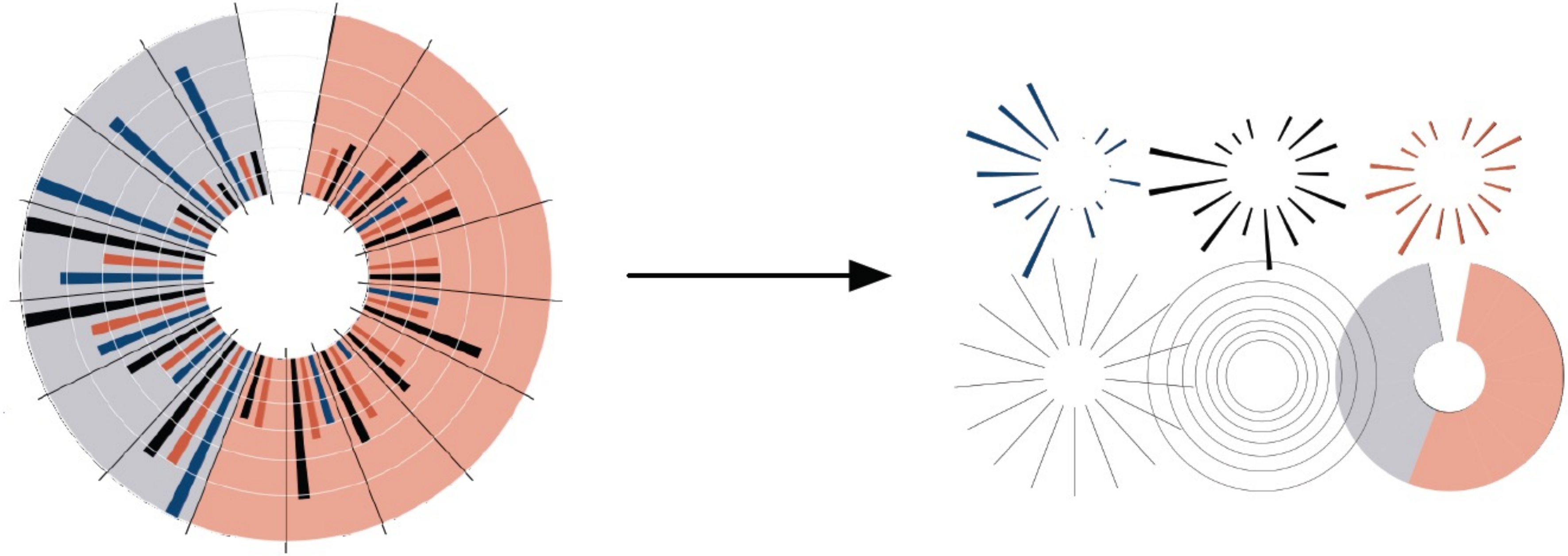
Data	Input data to visualize
Transforms	Group, aggregate, stats, layout
Scales	Map data values to visual values
Guides	Axes & legends visualize scales

Visualization Grammar

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**

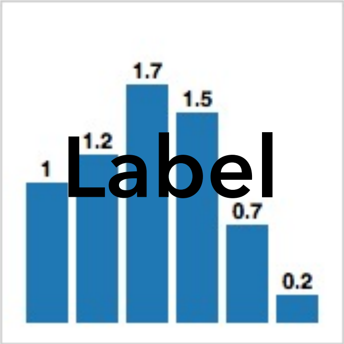
Area

Bar

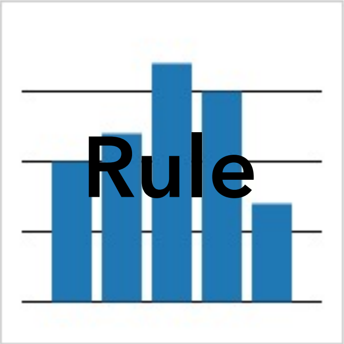
Dot

Image

Line



Label



Rule

Wedge

MARKS: Protovis graphical primitives

MARK

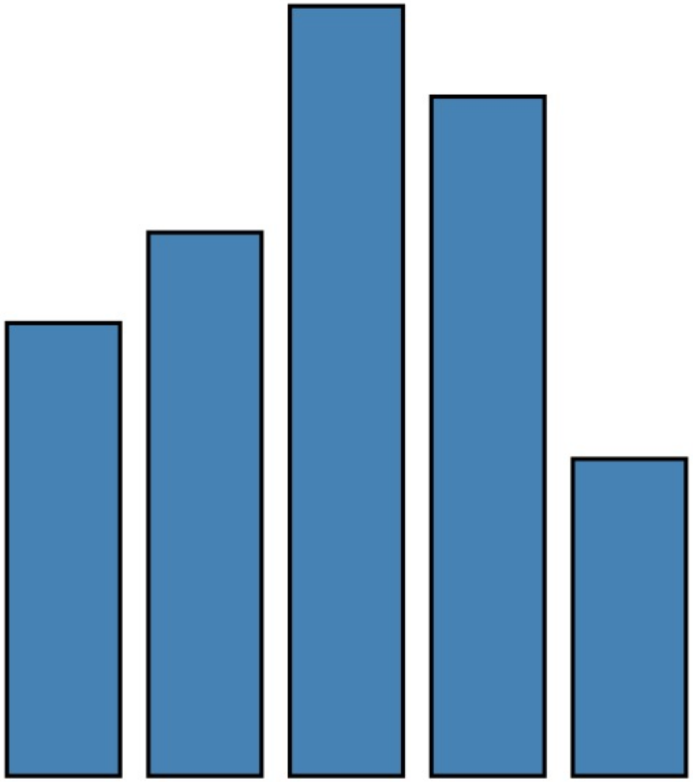
$\lambda : \boldsymbol{D} \longrightarrow \boldsymbol{R}$

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

RECT

$$\lambda : \boldsymbol{D} \longrightarrow \boldsymbol{R}$$

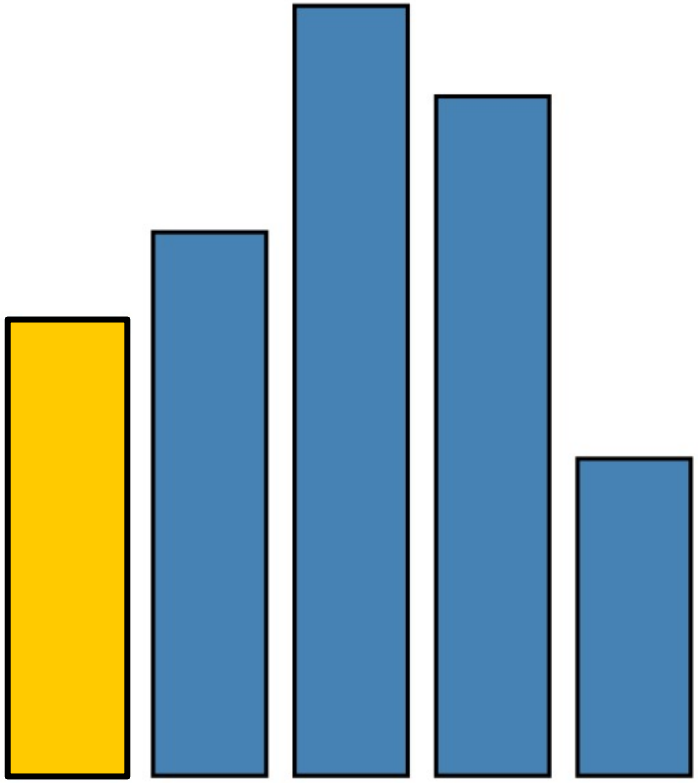
data	1	1.2	1.7	1.5	0.7
visible	true				
left	λ: index * 25				
bottom	0				
width	20				
height	λ: datum * 80				
fillStyle	blue				
strokeStyle	black				
lineWidth	1.5				
...	...				



RECT

$$\lambda : \boldsymbol{D} \longrightarrow \boldsymbol{R}$$

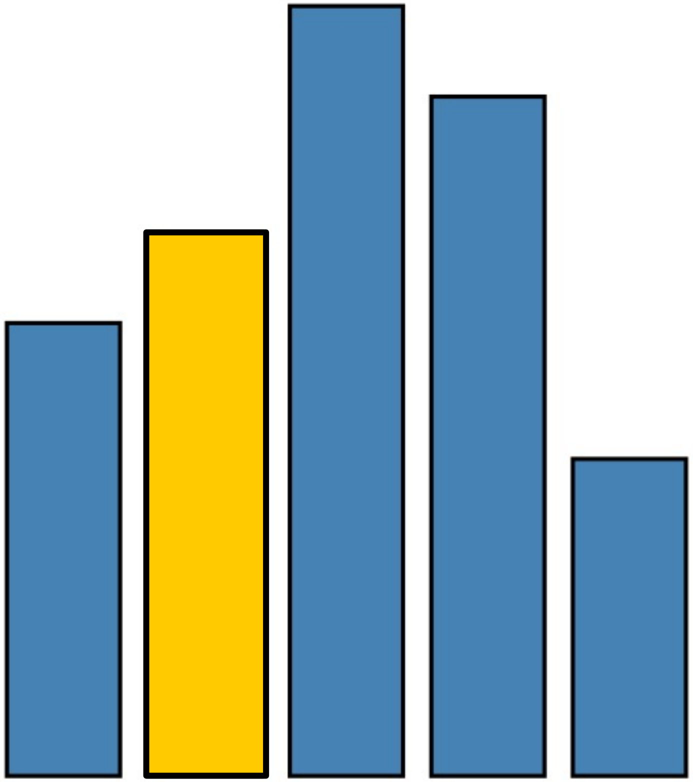
data	1	1.2	1.7	1.5	0.7
visible	true				
left	0 * 25				
bottom	0				
width	20				
height	1 * 80				
fillStyle	blue				
strokeStyle	black				
lineWidth	1.5				
…	…				



RECT

$$\lambda : \boldsymbol{D} \longrightarrow \boldsymbol{R}$$

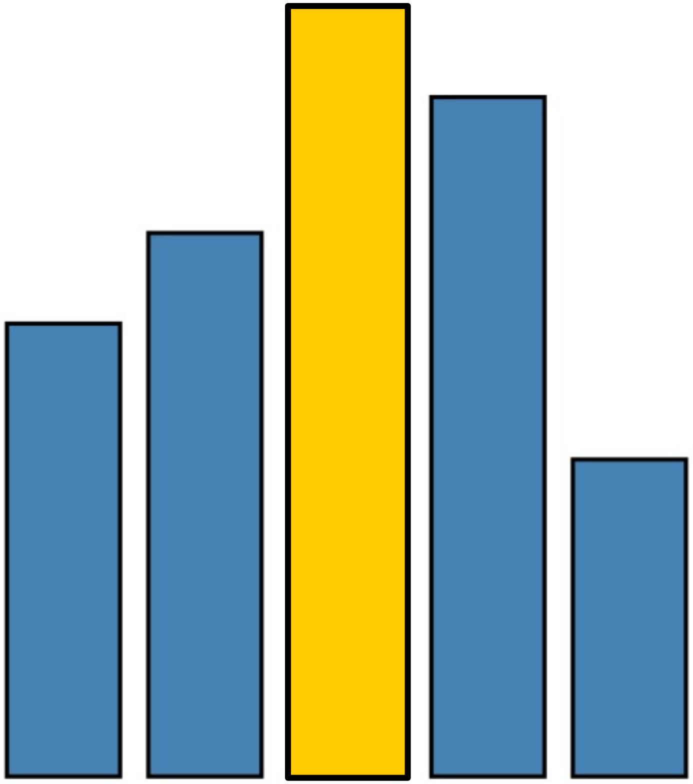
data	1	1.2	1.7	1.5	0.7
visible	true				
left	1 * 25				
bottom	0				
width	20				
height	1.2 * 80				
fillStyle	blue				
strokeStyle	black				
lineWidth	1.5				
…	…				



RECT

$$\lambda : \boldsymbol{D} \longrightarrow \boldsymbol{R}$$

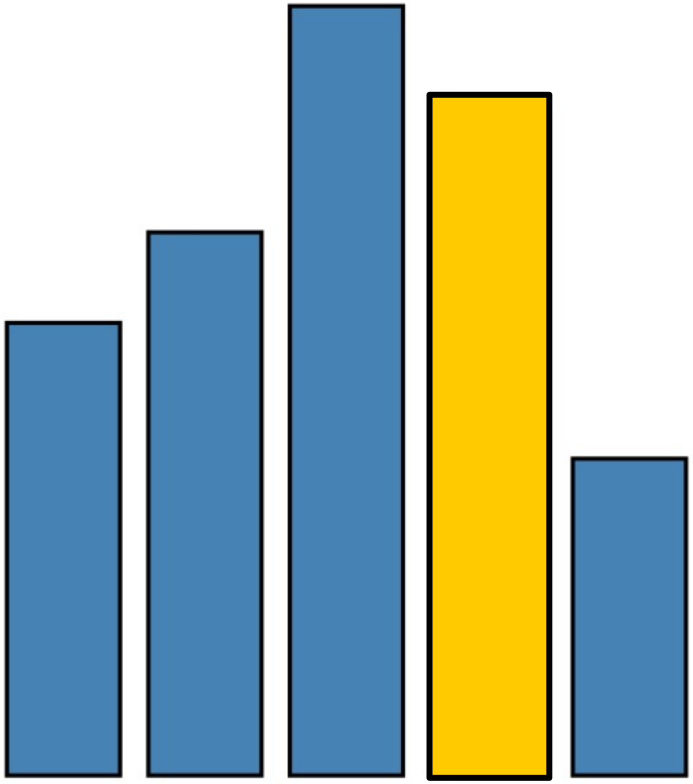
data	1	1.2	1.7	1.5	0.7
visible	true				
left	2 * 25				
bottom	0				
width	20				
height	1.7 * 80				
fillStyle	blue				
strokeStyle	black				
lineWidth	1.5				
…	…				



RECT

$$\lambda : \boldsymbol{D} \longrightarrow \boldsymbol{R}$$

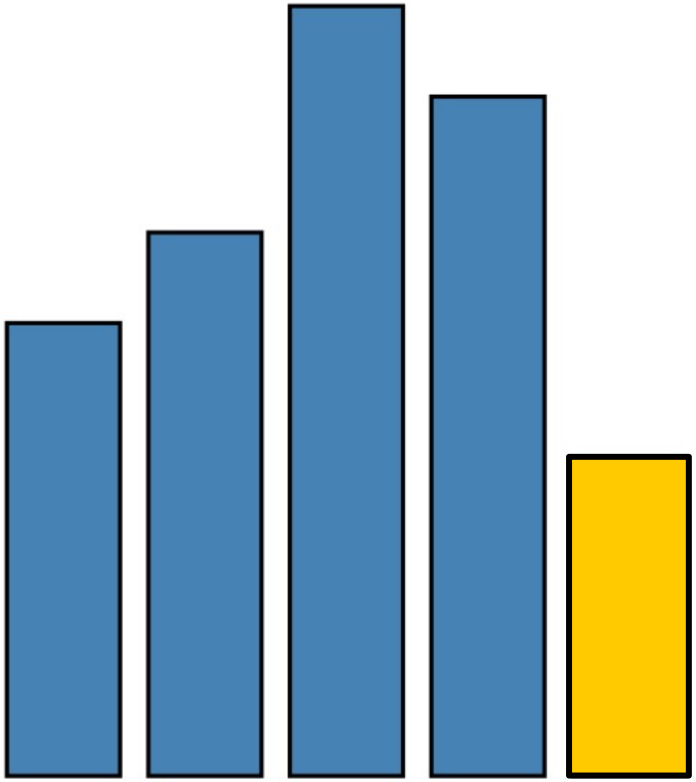
data	1	1.2	1.7	1.5	0.7
visible	true				
left	3 * 25				
bottom	0				
width	20				
height	1.5 * 80				
fillStyle	blue				
strokeStyle	black				
lineWidth	1.5				
...	...				



RECT

$$\lambda : \boldsymbol{D} \longrightarrow \boldsymbol{R}$$

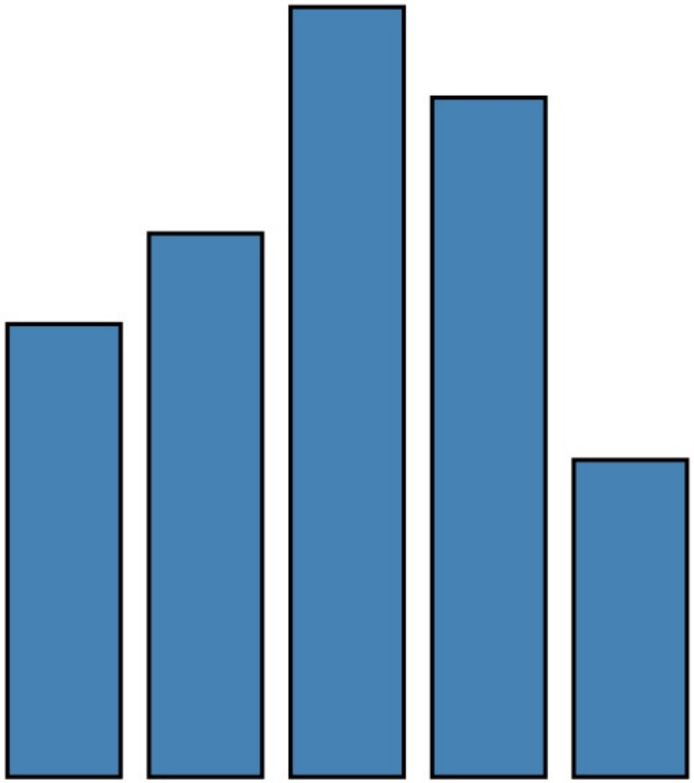
data	1	1.2	1.7	1.5	0.7
visible	true				
left	4 * 25				
bottom	0				
width	20				
height	0.7 * 80				
fillStyle	blue				
strokeStyle	black				
lineWidth	1.5				
...	...				



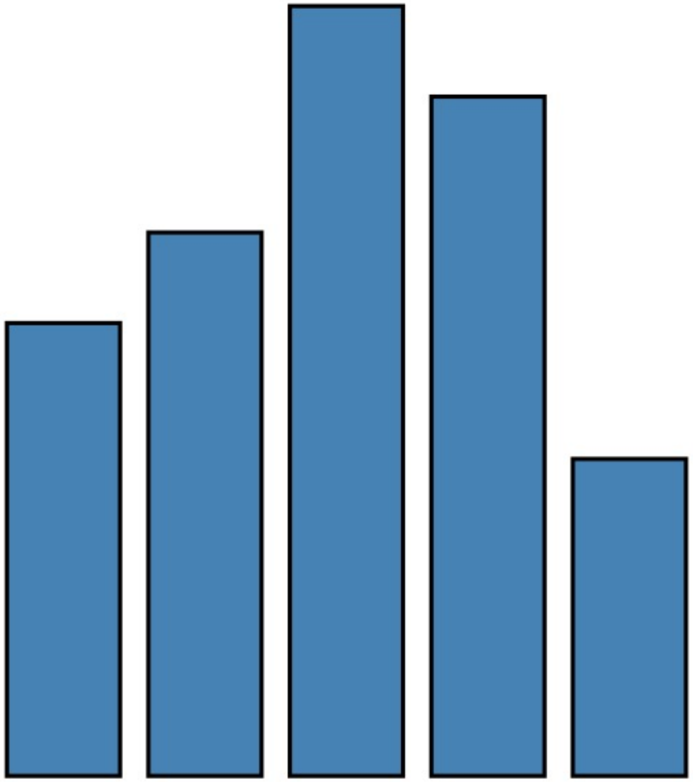
RECT

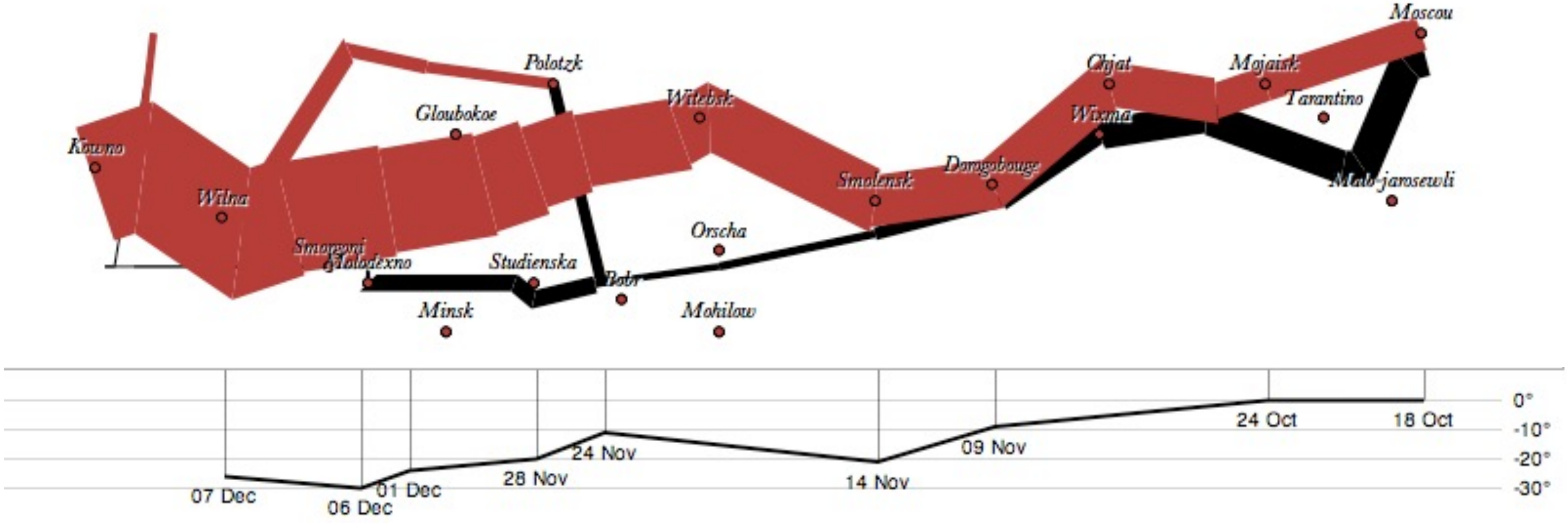
$$\lambda : \boldsymbol{D} \longrightarrow \boldsymbol{R}$$

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();
```





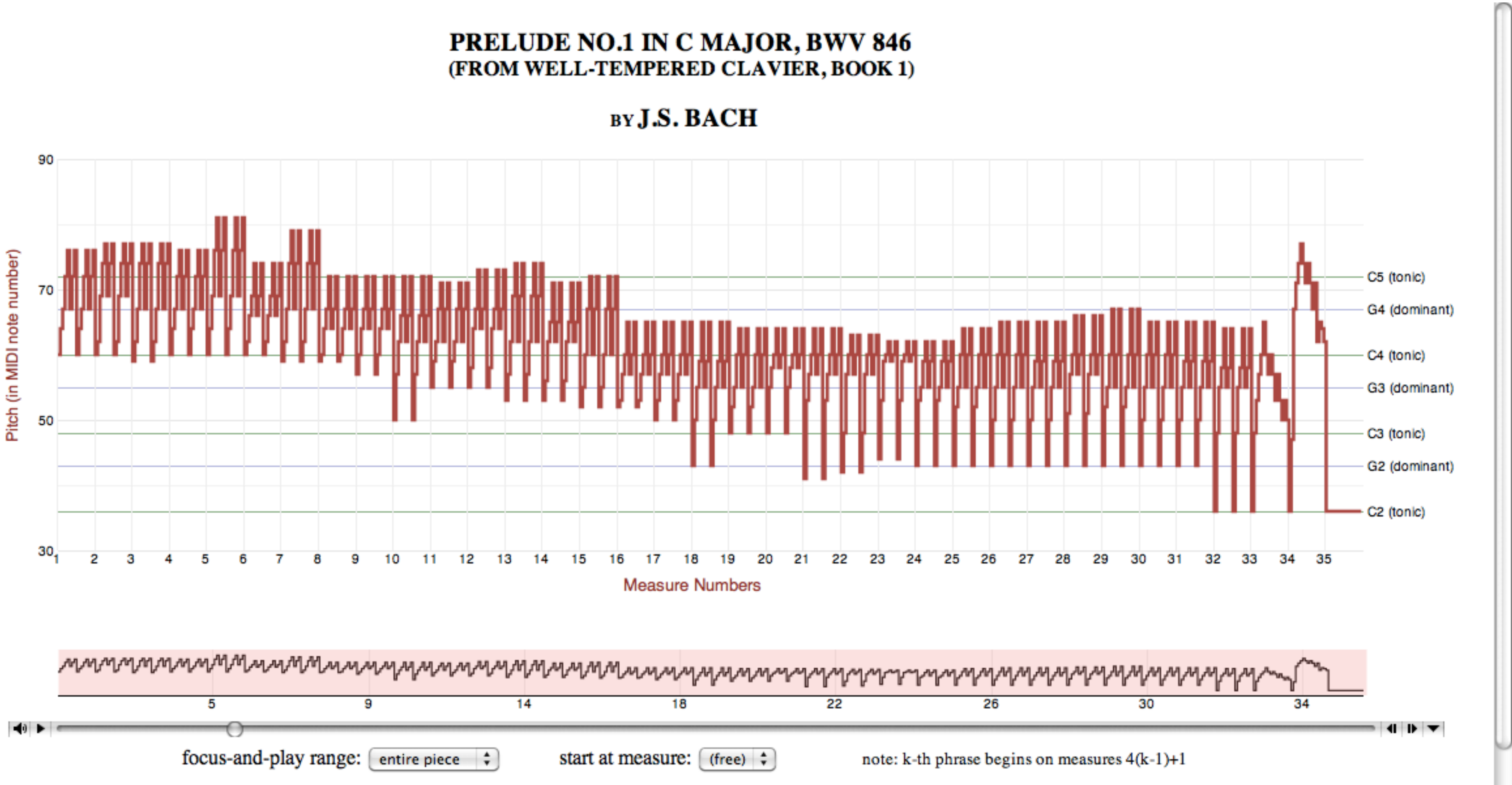
```
var army = pv.nest(napoleon.army, "dir", "group“);
var vis = new pv.Panel();

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");
```



PRELUDE NO.1 IN C MAJOR, BWV 846
(FROM WELL-TEMPERED CLAVIER, BOOK 1)

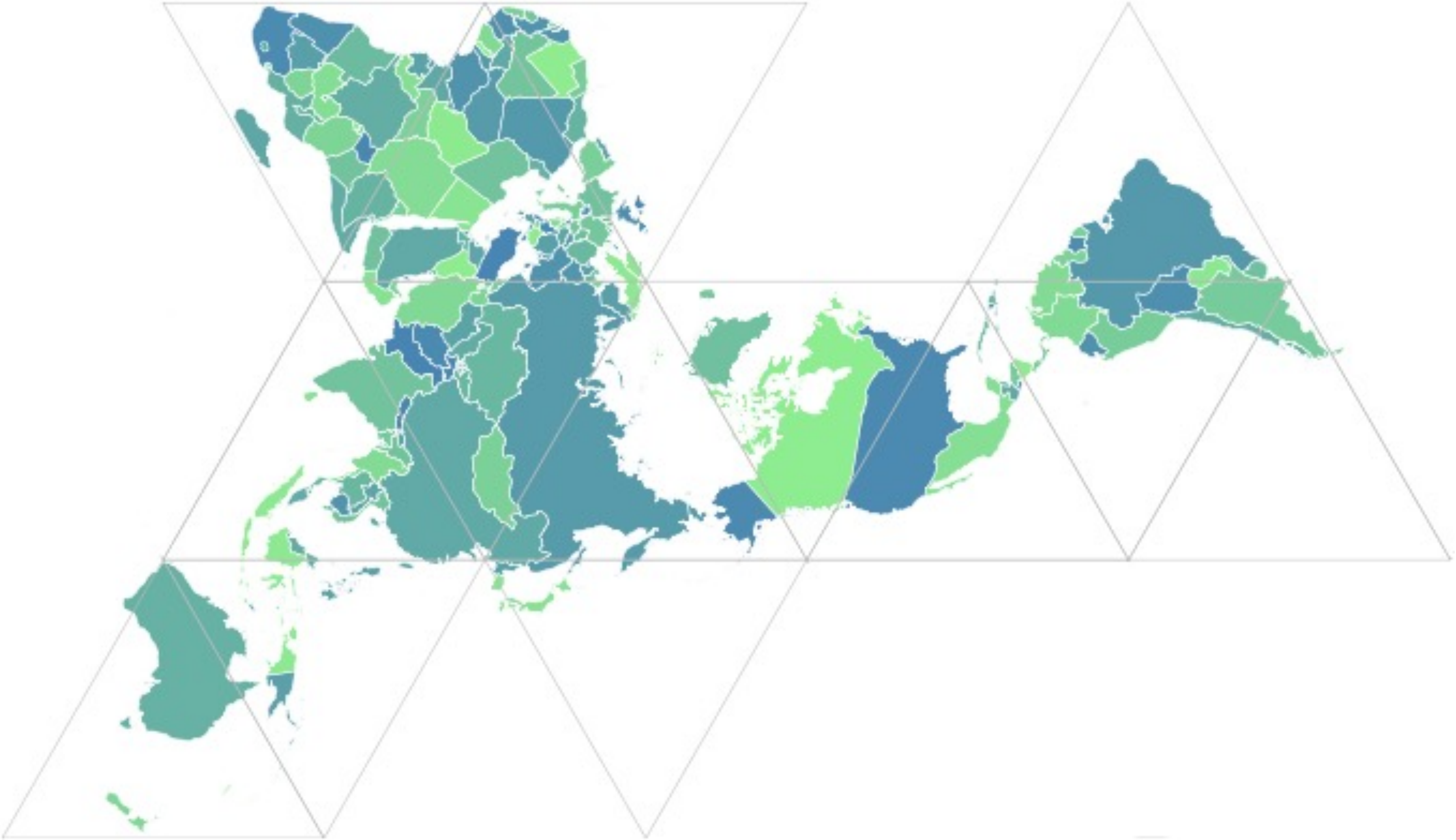
BY J.S. BACH

focus-and-play range: entire piece

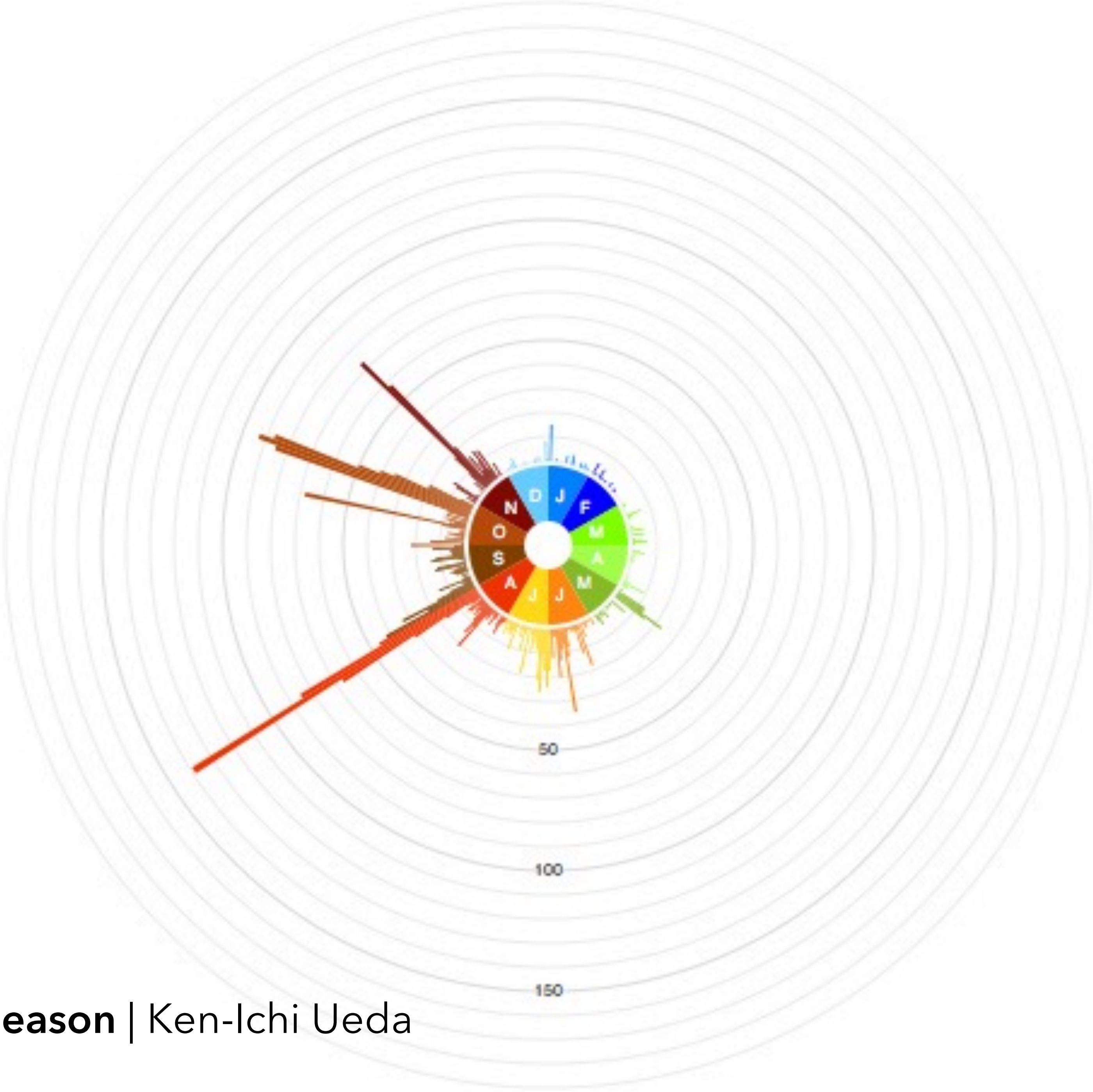
start at measure: (free)

note: k-th phrase begins on measures 4(k-1)+1

Bach's Prelude #1 in C Major | Jieun Oh

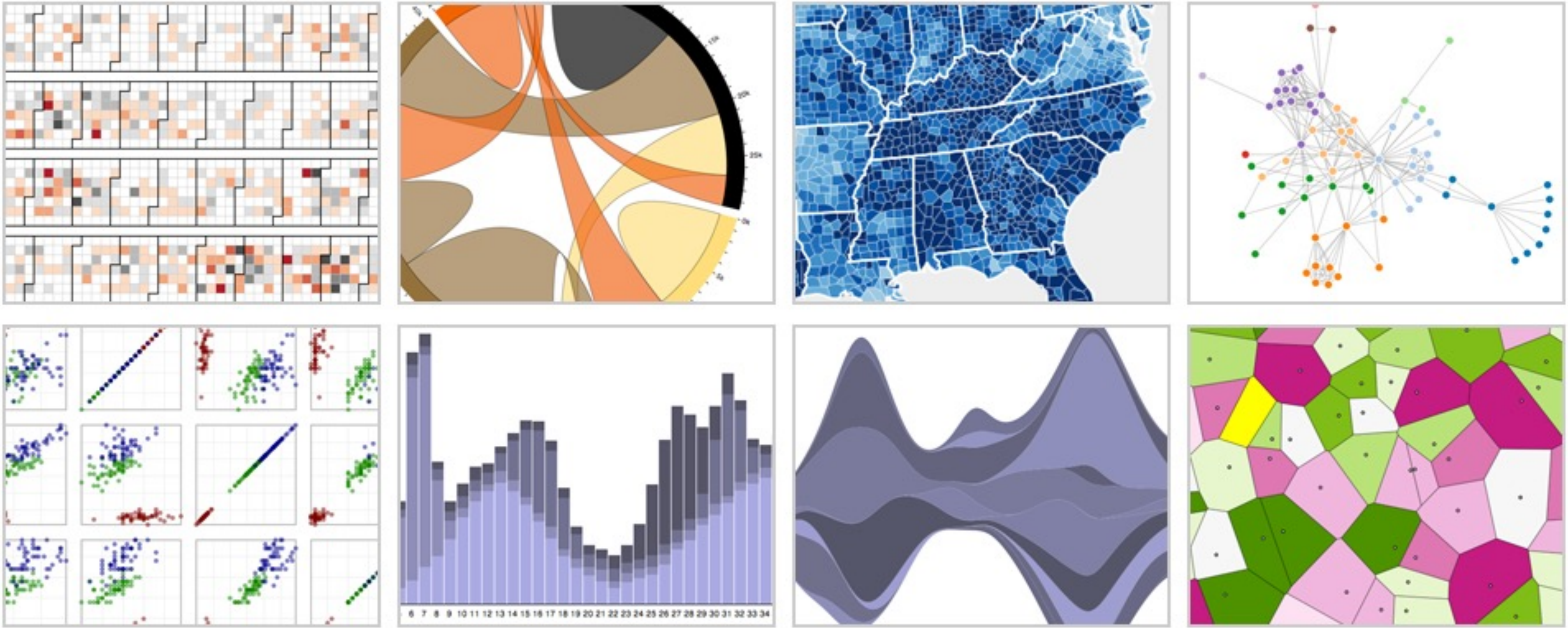


Dymaxion Maps | Vadim Ogievetsky



FlickrSeason | Ken-Ichi Ueda

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

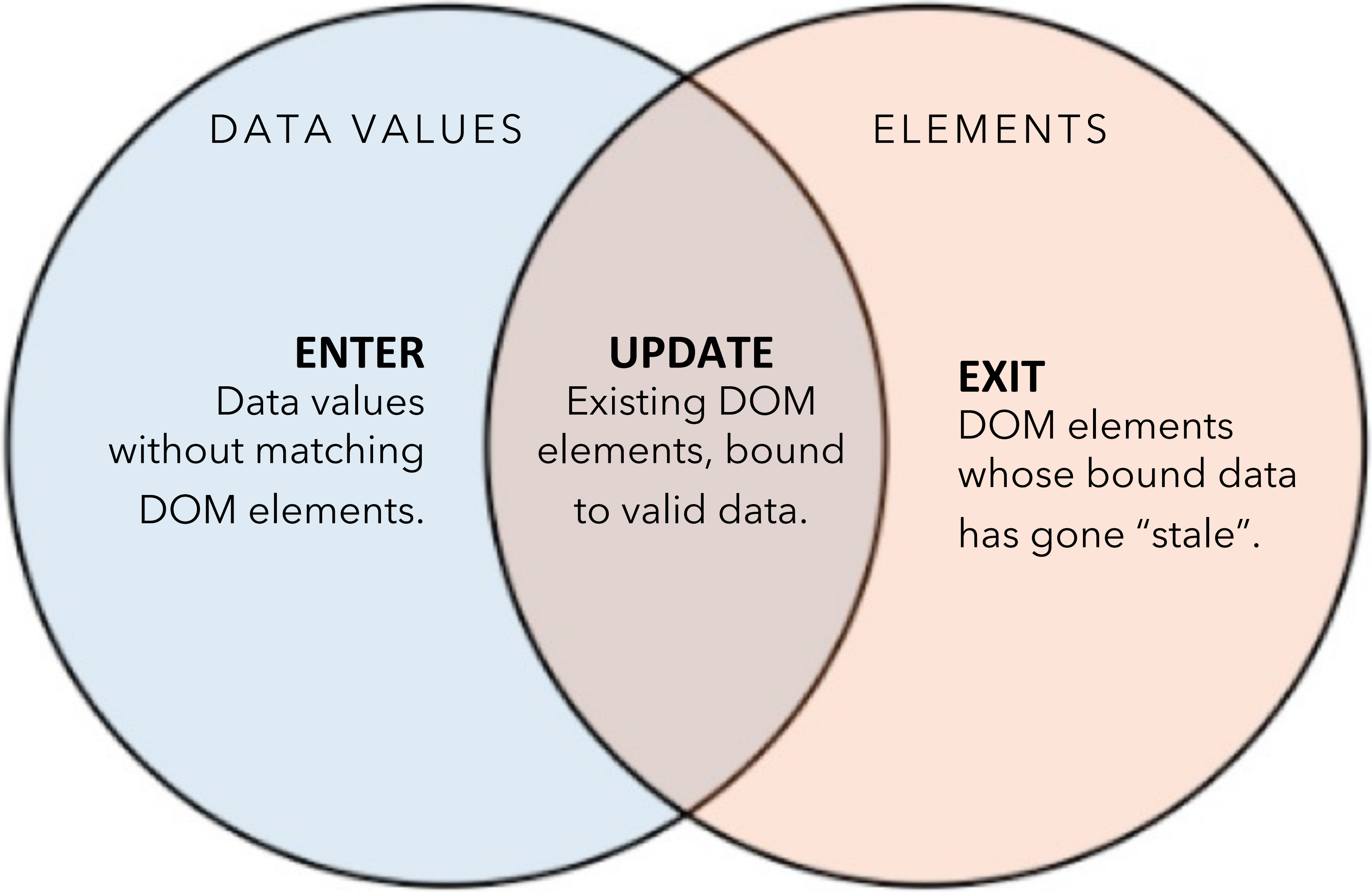
Selections can ***bind* data and DOM elements**.

```
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



DATA VALUES

ELEMENTS

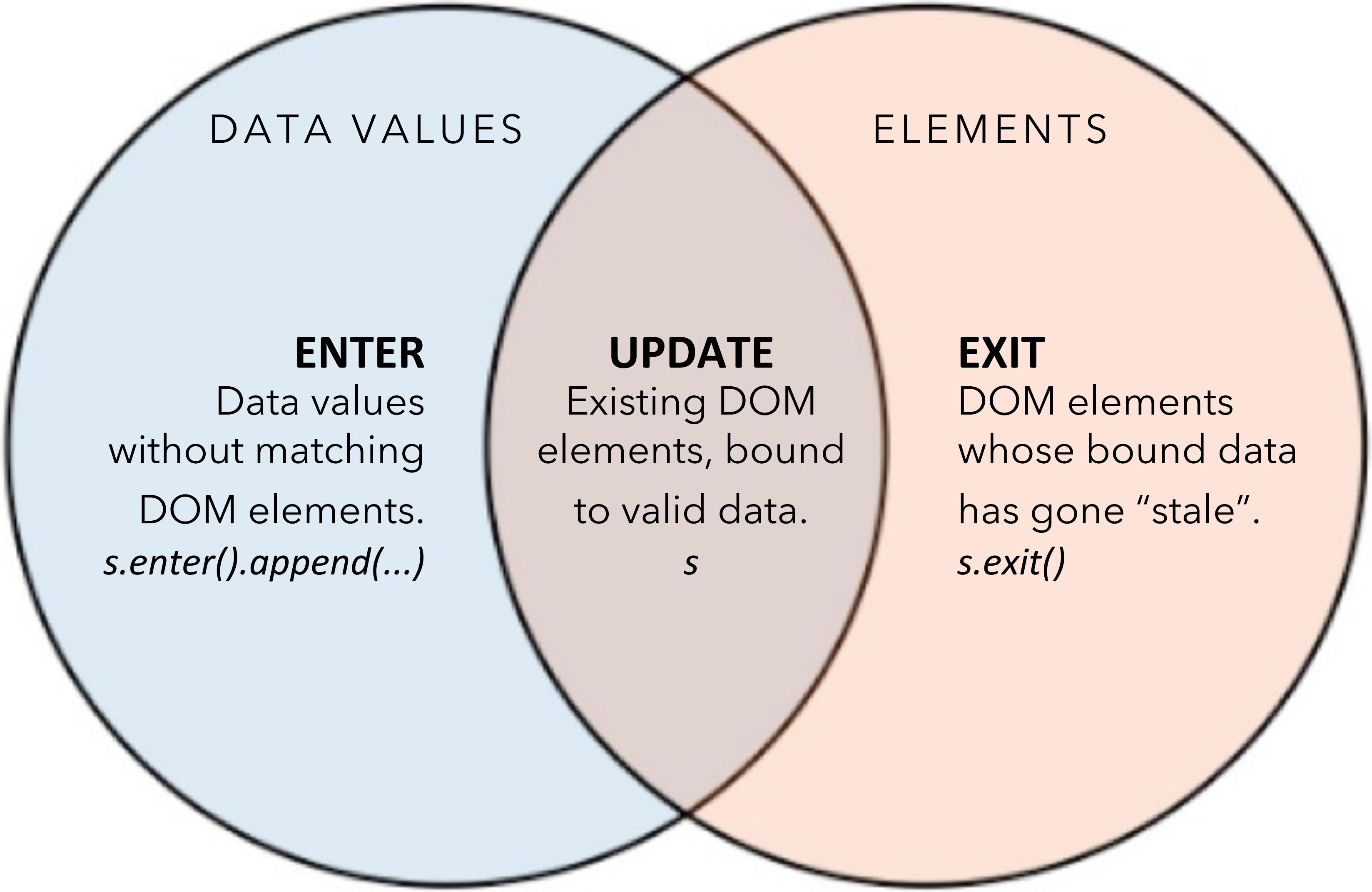
ENTER
Data values without matching DOM elements.

UPDATE
Existing DOM elements, bound to valid data.

EXIT
DOM elements whose bound data has gone “stale”.

The Data Join

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



DATA VALUES

ELEMENTS

ENTER
Data values
without matching
DOM elements.
s.enter().append(...)

UPDATE
Existing DOM
elements, bound
to valid data.
s

EXIT
DOM elements
whose bound data
has gone "stale".
s.exit()

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!

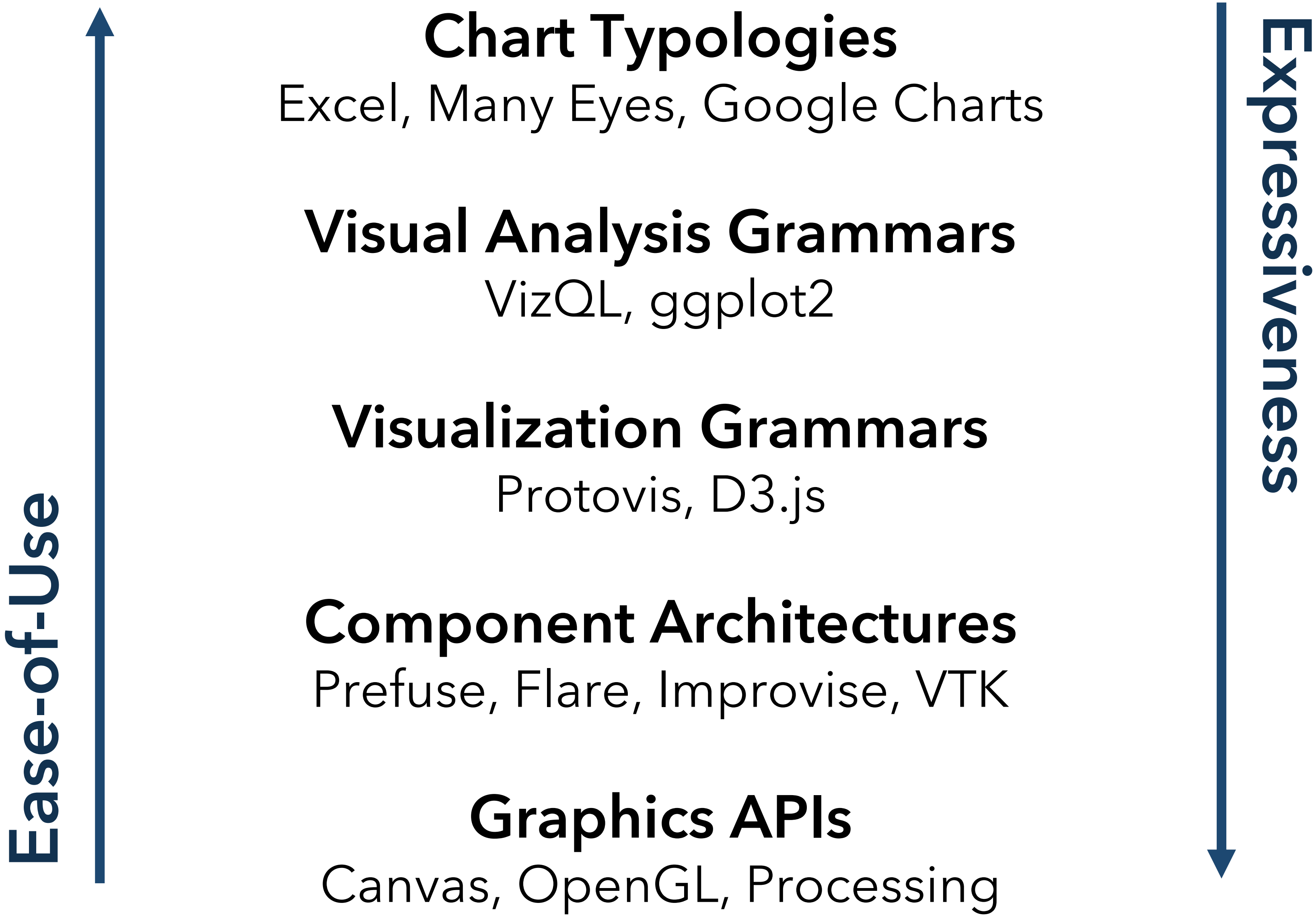


Chart Typologies
Excel, Many Eyes, Google Charts

Visual Analysis Grammars
VizQL, ggplot2

Visualization Grammars
Protovis, D3.js

Component Architectures
Prefuse, Flare, Improvise, VTK

Graphics APIs
Canvas, OpenGL, Processing

Administrivia

A2: Deceptive Visualization

Design **two** static visualizations for a dataset:

1. An *earnest* visualization that faithfully conveys the data
2. A *deceptive* visualization that tries to mislead viewers

Your two visualizations may address different questions.

Try to design a deceptive visualization that appears to be earnest: *can you trick your classmates and course staff?*

You are free to choose your own dataset, but we have also provided some preselected datasets for you.

Submit two images and a brief write-up on Gradescope.

Due by **Wed 1/25 11:59pm**.

A2 Peer Reviews

On Friday 1/27 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 2/1 11:59pm**.

Assignment Regrades

Students can request a regrade through Gradescope but must justify the reasons for the regrade.

Timeline: within 72 hours after the grade is released

We will review the entire assignment in more detail, which could result in a higher OR lower grade.

Tutorial on Thursday

D3.js Deep Dive: Thursday 1/26 during lecture, Led by Tukey and Yu

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 2/3 4:30-6pm in G20, Led by Aakash and Wei Jun

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

Declarative Languages

Visualization Grammars

Protovis, D3.js

Component Architectures

Prefuse, Flare, Improvise, VTK

Programming Toolkits

Graphics APIs

Canvas, OpenGL, Processing

Chart Typologies

Excel, Many Eyes, Google Charts

Charting Tools

Visual Analysis Grammars

VizQL, ggplot2

Declarative Languages

Visualization Grammars

Protovis, D3.js

Component Architectures

Prefuse, Flare, Improvise, VTK

Programming Toolkits

Graphics APIs

Canvas, OpenGL, Processing

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))
```

ODAY'S PAPER | VIDEO | MOST POPULAR | TIMES TOPICS | MOST RECENT

Subscribe to The Times | Welcome, jmheer

The New York Times

Tuesday, October 26, 2010 Last Update: 3:50 PM ET

Search

ING DIRECT

Subscribe to Hom

2010 Midterm Elections

Tea Party Vow to Deter Voter Fraud Is Called Scare Tactic

By IAN URBINA 2:19 PM ET

Voting rights group say that Tea Party members' plan to question voters' eligibility at the polls is intended to suppress minority and poor voters.

Post a Comment | Read (355)

Glaxo Pays $750 Million Fine for Tainted Products

By GARDINER HARRIS and DUFF

Joshua Kristal for The New York Times

Painting at 99, With No Compromises

By ROBIN FINN

An exhibition celebrating Will Barnet's centennial year traces his evolution as a modern American artist.

OPINION »

OP-ED CONTRIBUTOR

Humans to Asteroids: Watch Out!

How to keep near-Earth objects from hitting us.

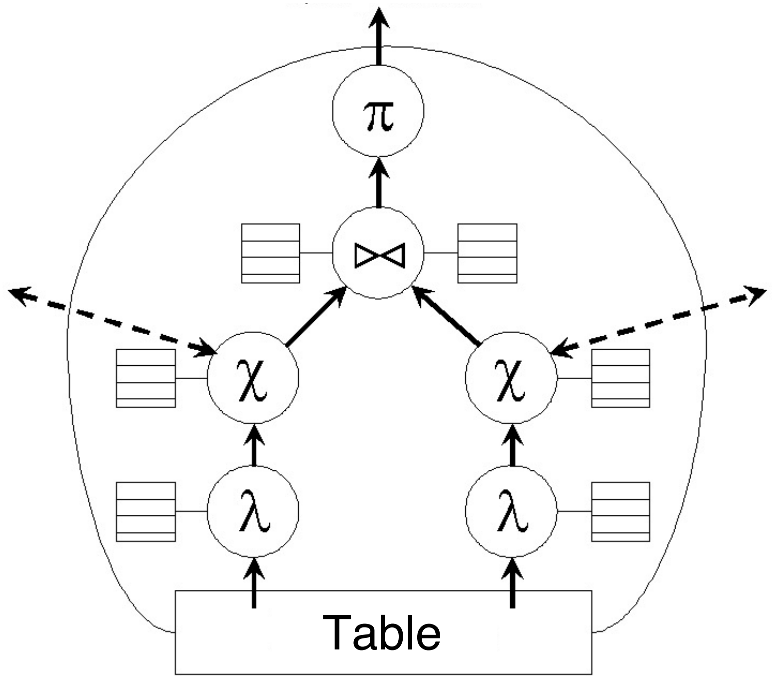
- Brooks: No Second Thoughts | Comments (200)
- Herbert: The Corrosion of America
- Cohen: Turkey Steps Out
- Editorial: Mortgage Mess
- Bloggingheads: Jon Stewart's Power

MARKETS » At 3:56 PM ET

S.&P. 500 | Dow | Nasdaq

```
<!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/[illegible]/wto/wt_capi.
     charset="utf-8"></script>
     <span id="wto_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

Declarative
Languages

Visualization Grammars
Protovis, D3.js

Programming
Toolkits

Component Architectures
Prefuse, Flare, Improvise, VTK

Graphics APIs
Processing, OpenGL, Java2D

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***

Declarative
Languages

Visualization Grammars

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

Component Architectures

Prefuse, Flare, Improvise, VTK

Programming
Toolkits

Graphics APIs

Processing, OpenGL, Java2D

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***

Declarative
Languages

Visualization Grammars

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

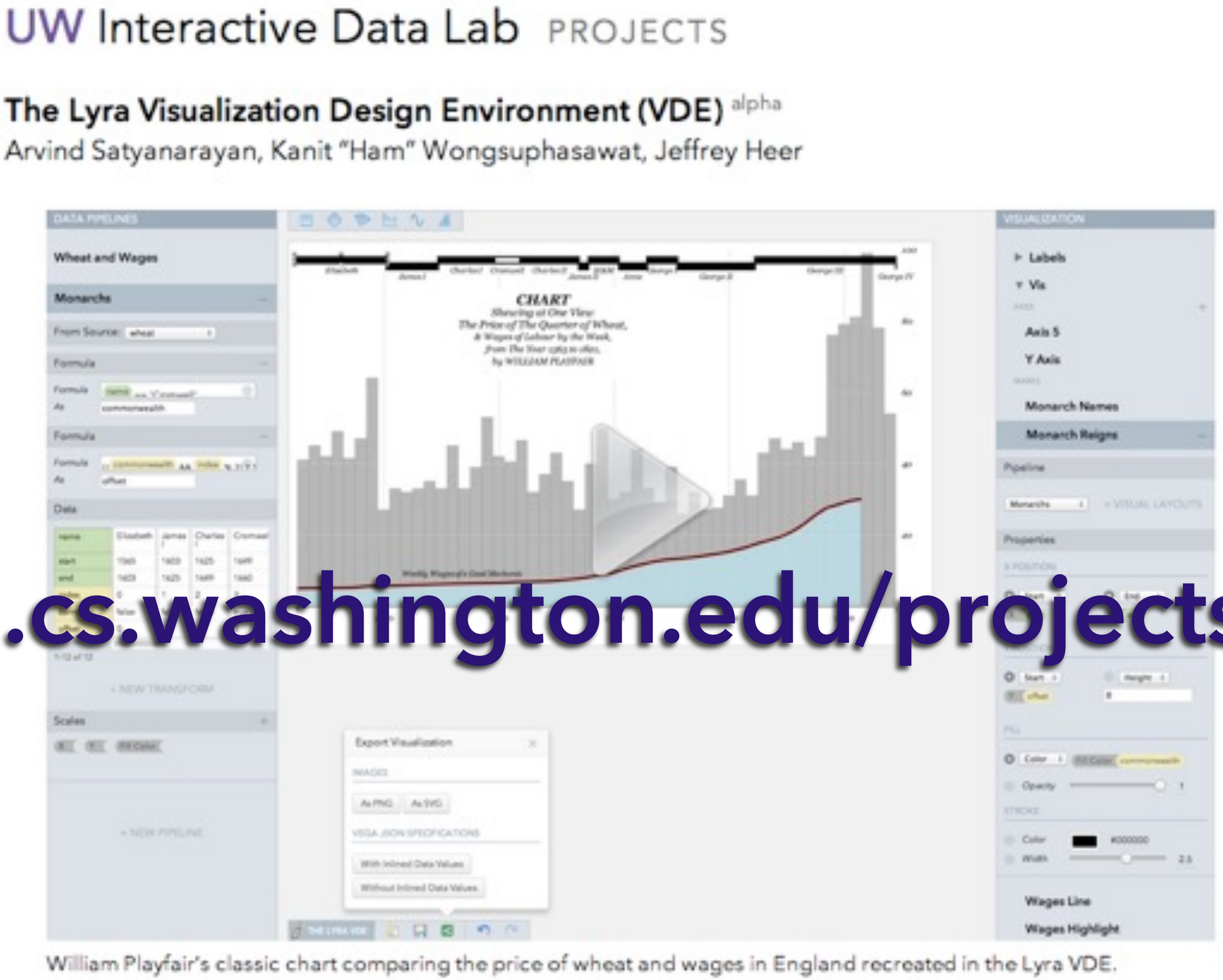
Component Architectures

Prefuse, Flare, Improvise, VTK

Programming
Toolkits

Graphics APIs

Processing, OpenGL, Java2D



UW Interactive Data Lab PROJECTS

The Lyra Visualization Design Environment (VDE) alpha

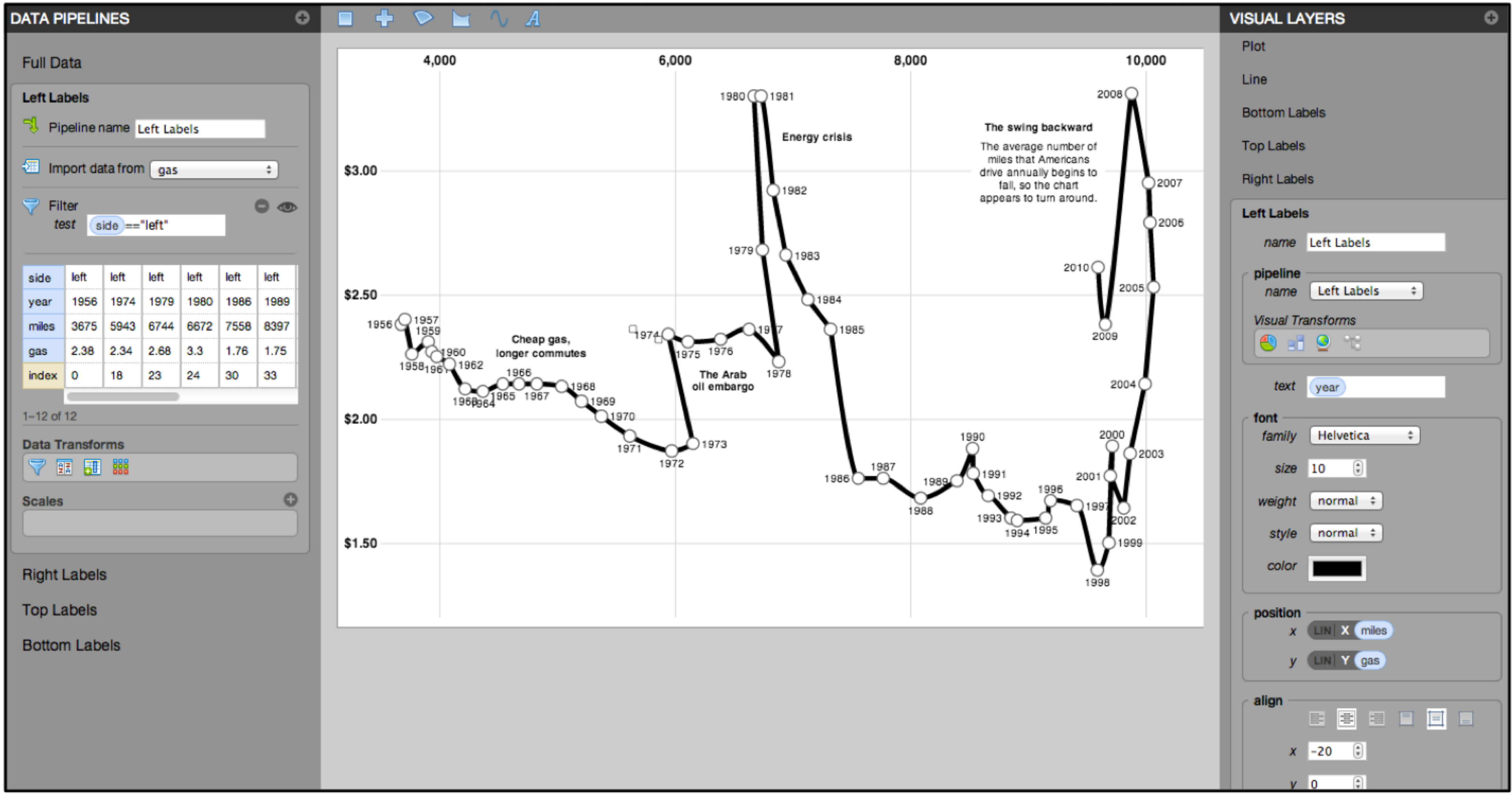
Arvind Satyanarayan, Kanit "Ham" Wongsuphasawat, Jeffrey Heer

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



DATA PIPELINES

Full Data

Left Labels

Pipeline name: Left Labels

Import data from: gas

Filter

test: side=="left"

side	left	left	left	left	left	left
year	1956	1974	1979	1980	1986	1989
miles	3675	5943	6744	6672	7558	8397
gas	2.38	2.34	2.68	3.3	1.76	1.75
index	0	18	23	24	30	33

1–12 of 12

Data Transforms

Scales

Right Labels

Top Labels

Bottom Labels

VISUAL LAYERS

Plot

Line

Bottom Labels

Top Labels

Right Labels

Left Labels

name: Left Labels

pipeline name: Left Labels

Visual Transforms

text: year

font family: Helvetica

size: 10

weight: normal

style: normal

color: ■

position x: LIN X miles

y: LIN Y gas

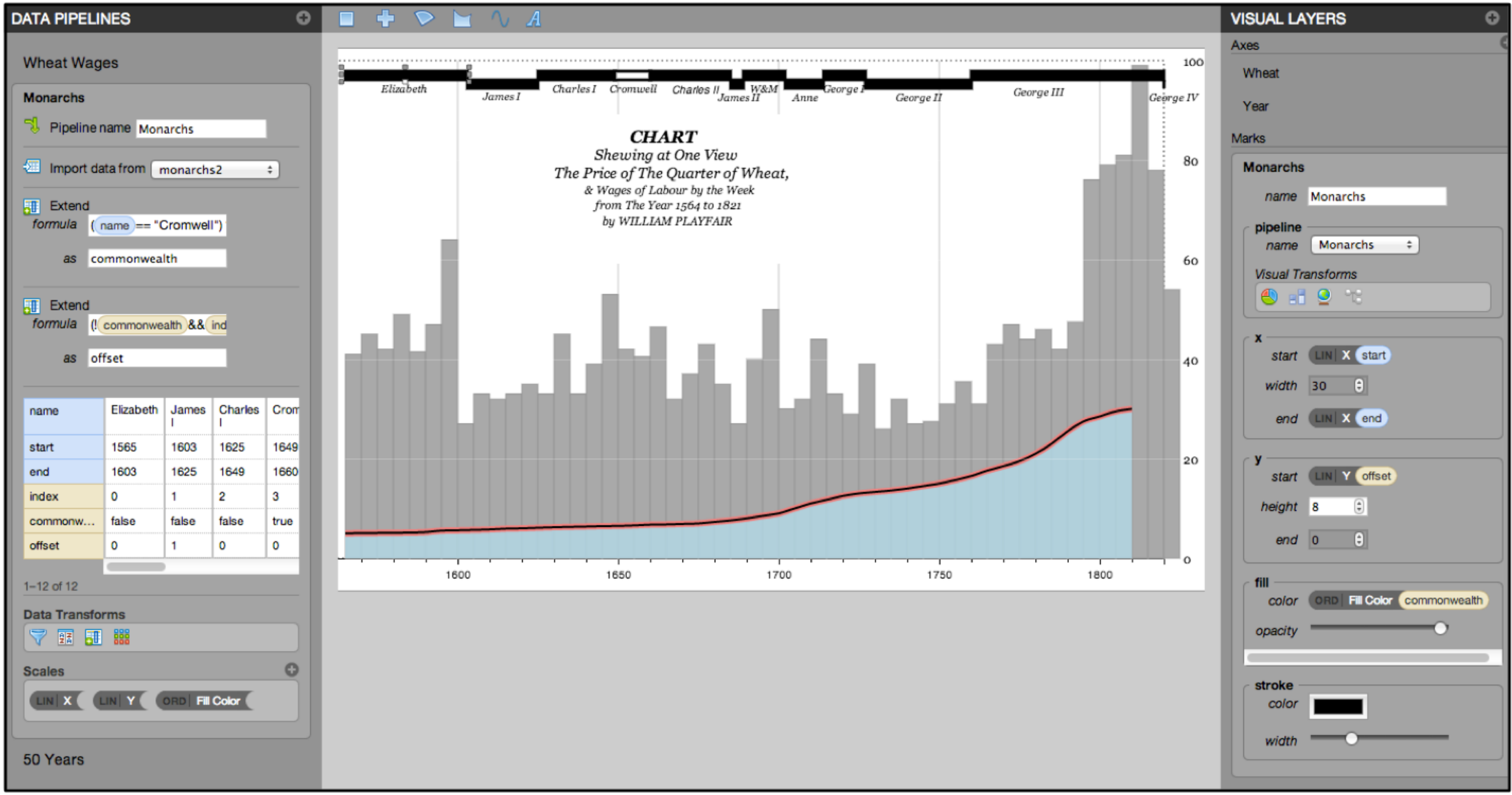
align

x: -20

y: 0

Driving Shifts into Reverse by Hannah Fairfield, NYTimes

Lyra A Visualization Design Environment



Pipeline name Monarchs

Import data from monarchs2

Extend
formula (name == "Cromwell")
as commonwealth

Extend
formula (!commonwealth && ind
as offset

name	Elizabeth	James I	Charles I	Crom
start	1565	1603	1625	1649
end	1603	1625	1649	1660
index	0	1	2	3
commonw...	false	false	false	true
offset	0	1	0	0

1–12 of 12

name Monarchs

pipeline
name Monarchs

x
start LIN X start
width 30
end LIN X end

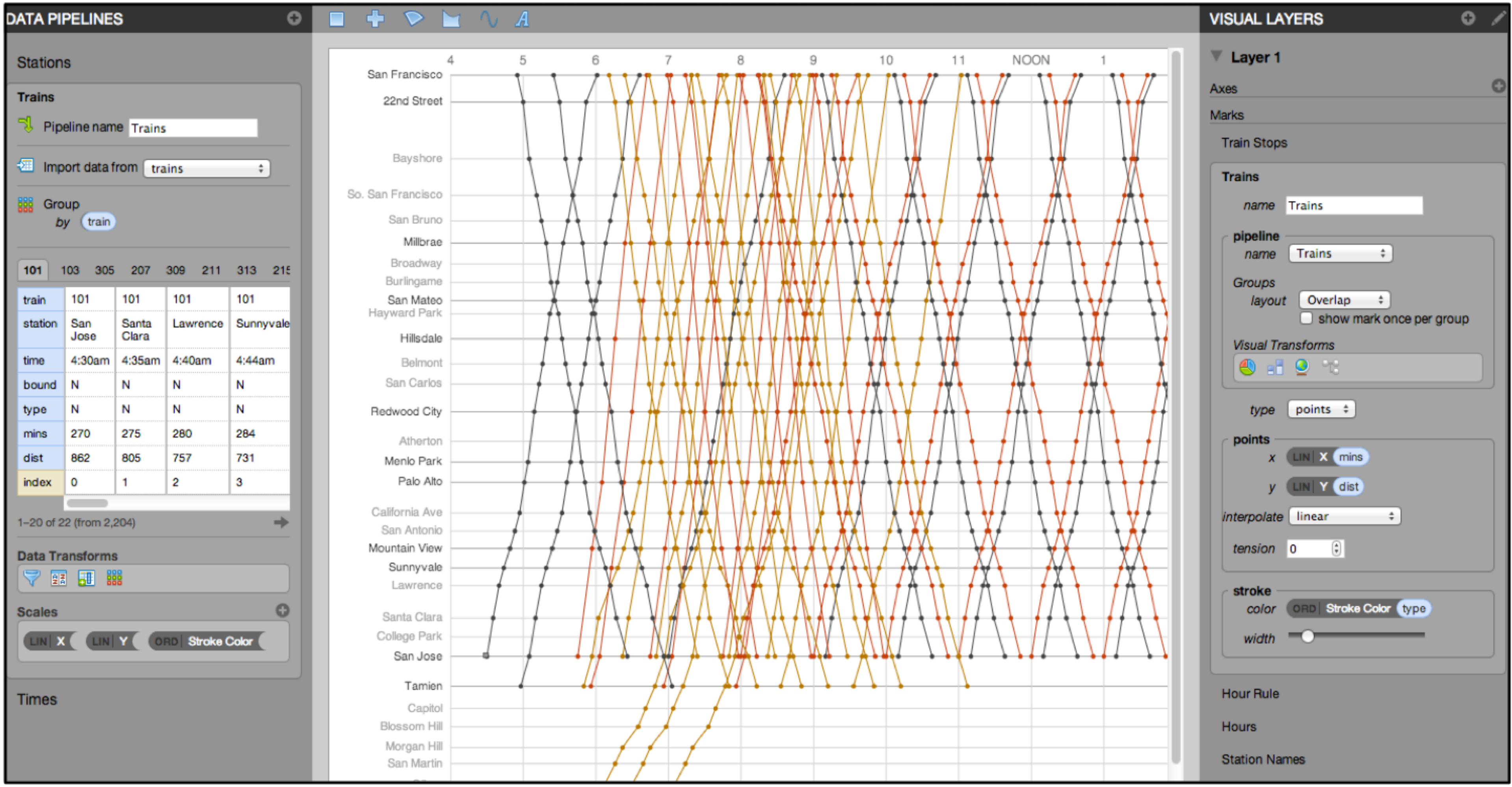
y
start LIN Y offset
height 8
end 0

fill
color ORD Fill Color commonwealth
opacity

stroke
color
width

by William Playfair

Lyra A Visualization Design Environment



DATA PIPELINES

Stations

Trains

Pipeline name: Trains

Import data from: trains

Group by: train

101 | 103 | 305 | 207 | 309 | 211 | 313 | 215

train	101	101	101	101
station	San Jose	Santa Clara	Lawrence	Sunnyvale
time	4:30am	4:35am	4:40am	4:44am
bound	N	N	N	N
type	N	N	N	N
mins	270	275	280	284
dist	862	805	757	731
index	0	1	2	3

1–20 of 22 (from 2,204)

Data Transforms

Scales: LIN X, LIN Y, ORD Stroke Color

Times

VISUAL LAYERS

Layer 1

Axes

Marks

Train Stops

Trains

name: Trains

pipeline

name: Trains

Groups

layout: Overlap

☐ show mark once per group

Visual Transforms

type: points

points

x: LIN X mins

y: LIN Y dist

interpolate: linear

tension: 0

stroke

color: ORD Stroke Color type

width

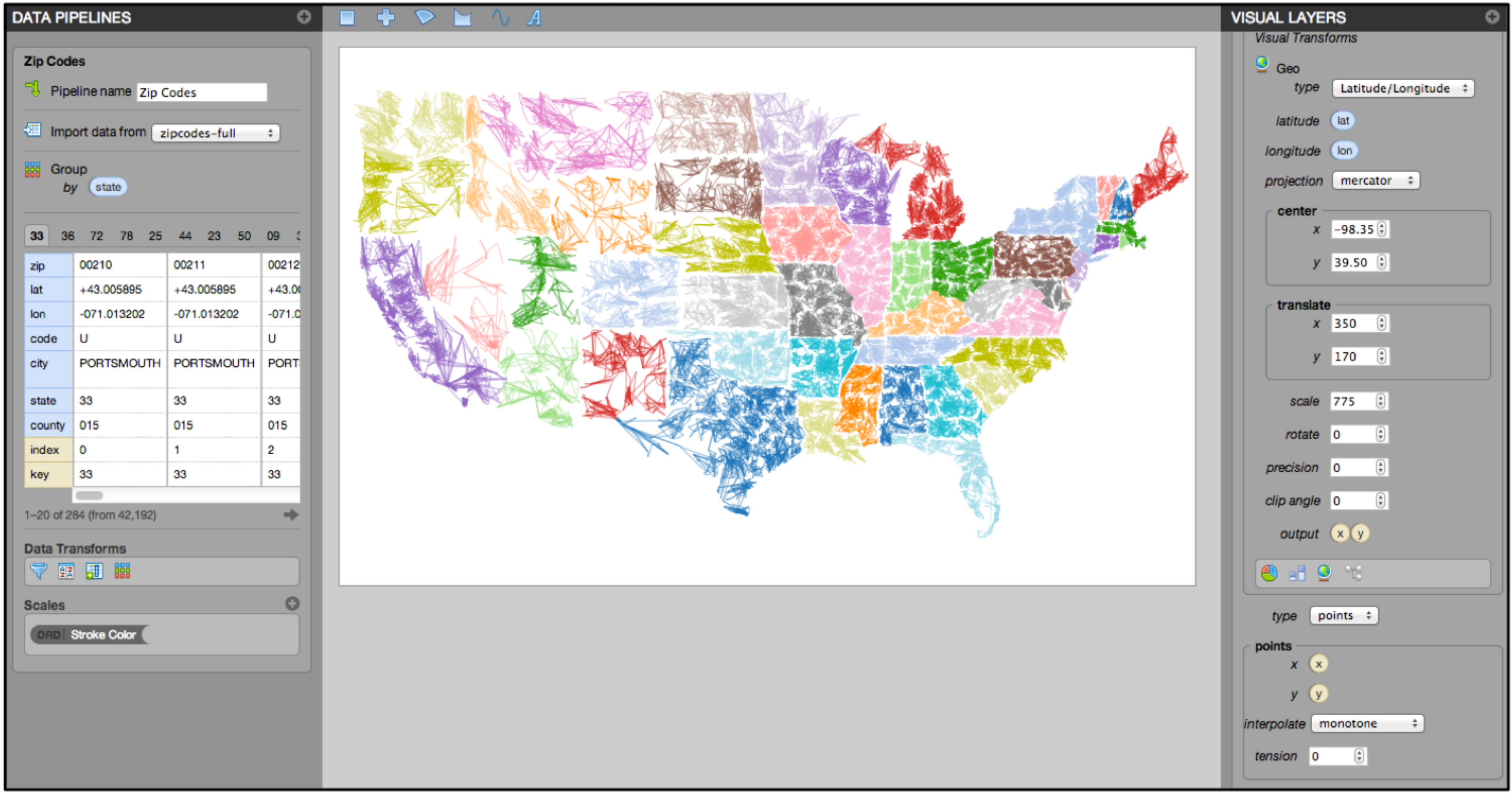
Hour Rule

Hours

Station Names

based on the **Railway Timetable** by E. J. Marey

Lyra A Visualization Design Environment



Zip Codes

Pipeline name: Zip Codes

Import data from: zipcodes-full

Group by: state

33 36 72 78 25 44 23 50 09

zip	00210	00211	00212
lat	+43.005895	+43.005895	+43.0
lon	-071.013202	-071.013202	-071.0
code	U	U	U
city	PORTSMOUTH	PORTSMOUTH	PORT
state	33	33	33
county	015	015	015
index	0	1	2
key	33	33	33

1–20 of 284 (from 42,192)

Data Transforms

Scales

ORD Stroke Color

Visual Transforms

Geo

type: Latitude/Longitude

latitude: lat

longitude: lon

projection: mercator

center x: -98.35

center y: 39.50

translate x: 350

translate y: 170

scale: 775

rotate: 0

precision: 0

clip angle: 0

output: x y

type: points

points x: x

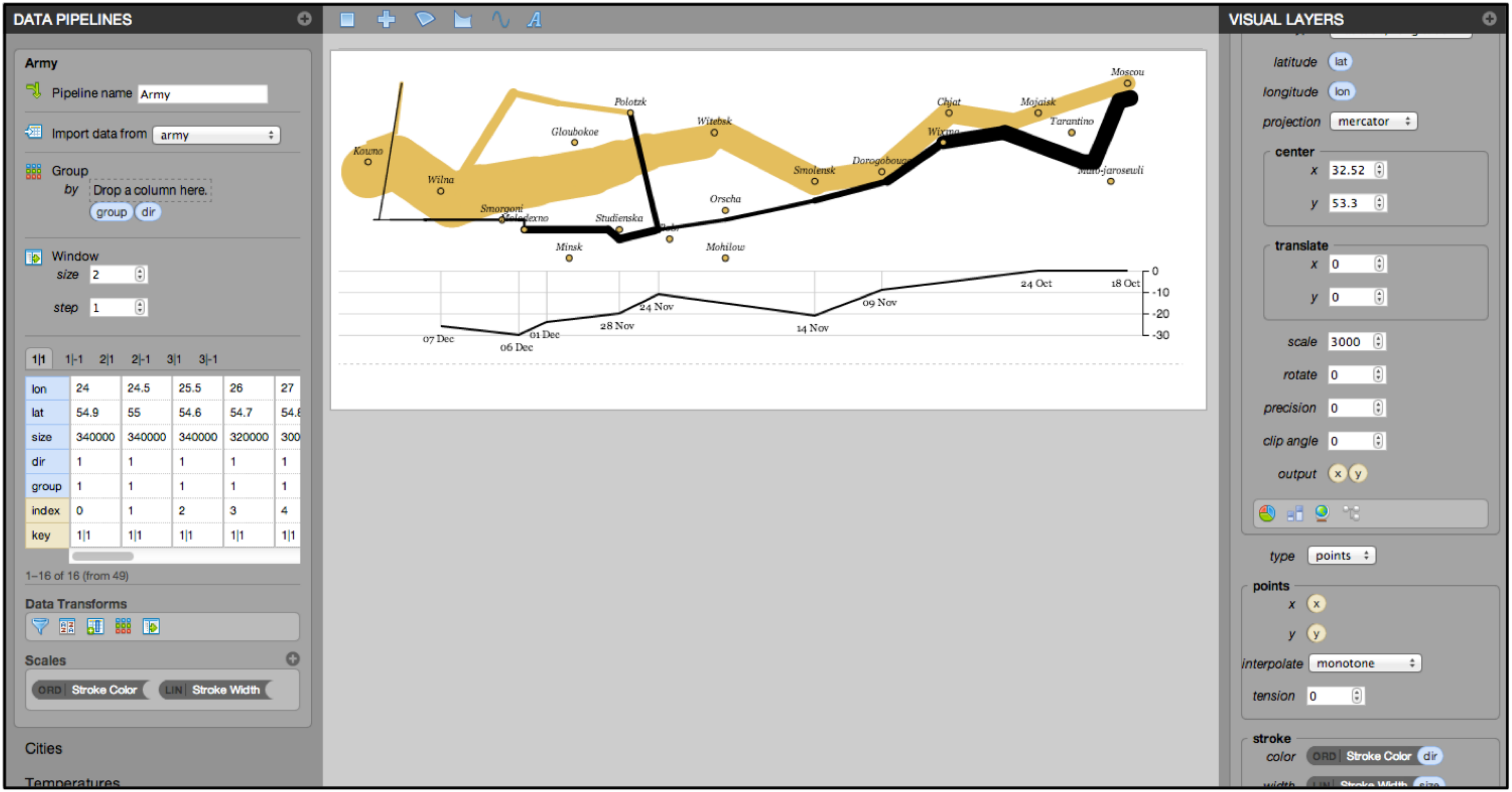
points y: y

interpolate: monotone

tension: 0

ZipScribble by Robert Kosara

Lyra A Visualization Design Environment



DATA PIPELINES

Army

Pipeline name: Army

Import data from: army

Group by: Drop a column here.

group dir

Window

size: 2

step: 1

1|1 1|-1 2|1 2|-1 3|1 3|-1

lon	24	24.5	25.5	26	27
lat	54.9	55	54.6	54.7	54.
size	340000	340000	340000	320000	300
dir	1	1	1	1	1
group	1	1	1	1	1
index	0	1	2	3	4
key	1|1	1|1	1|1	1|1	1|1

1–16 of 16 (from 49)

Data Transforms

Scales

ORD Stroke Color

LIN Stroke Width

Cities

Temperatures

VISUAL LAYERS

latitude: lat

longitude: lon

projection: mercator

center

x: 32.52

y: 53.3

translate

x: 0

y: 0

scale: 3000

rotate: 0

precision: 0

clip angle: 0

output: x y

type: points

points

x: x

y: y

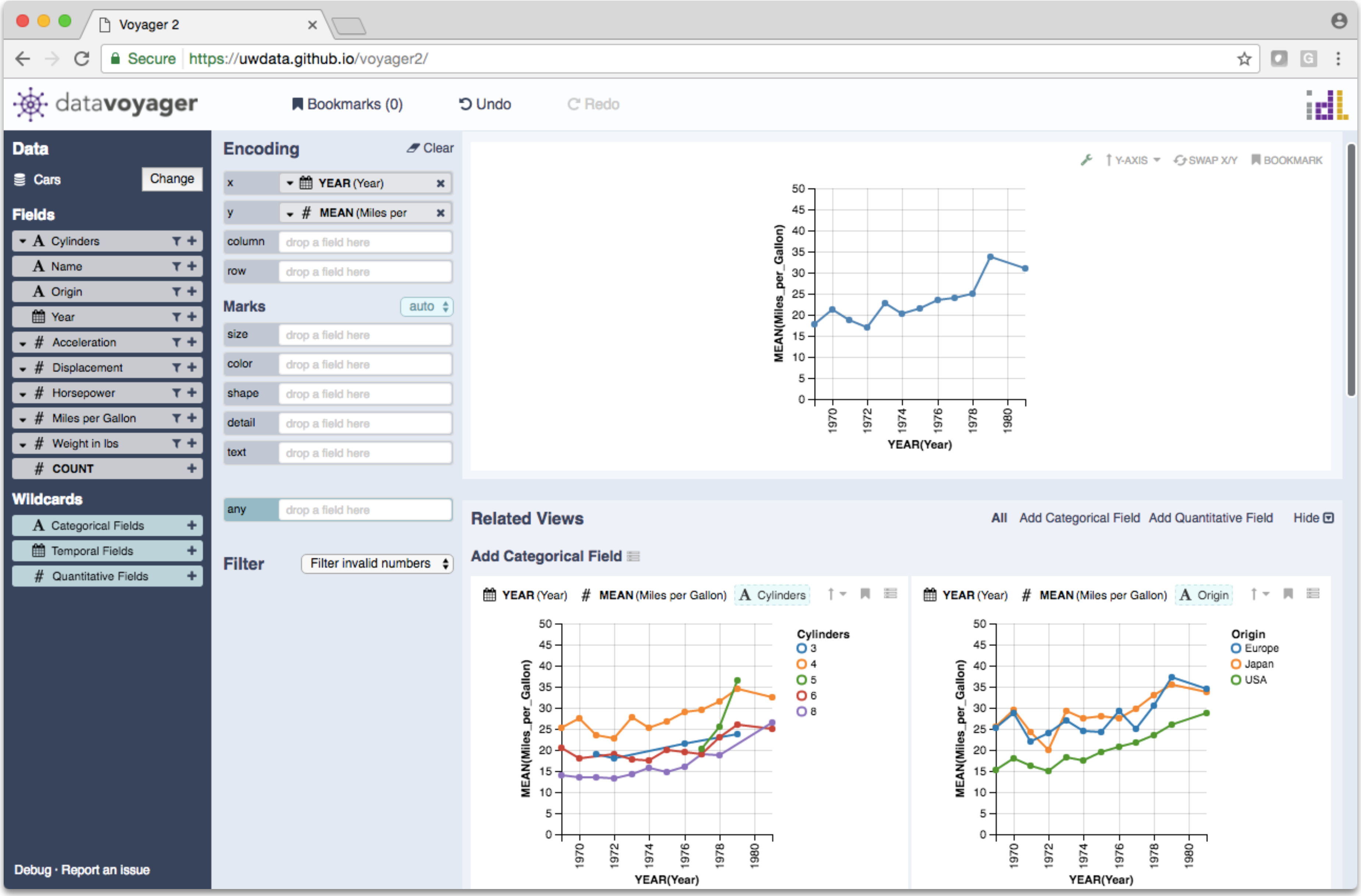
interpolate: monotone

tension: 0

stroke

color: ORD Stroke Color dir

Napoleon's March by Charles Minard

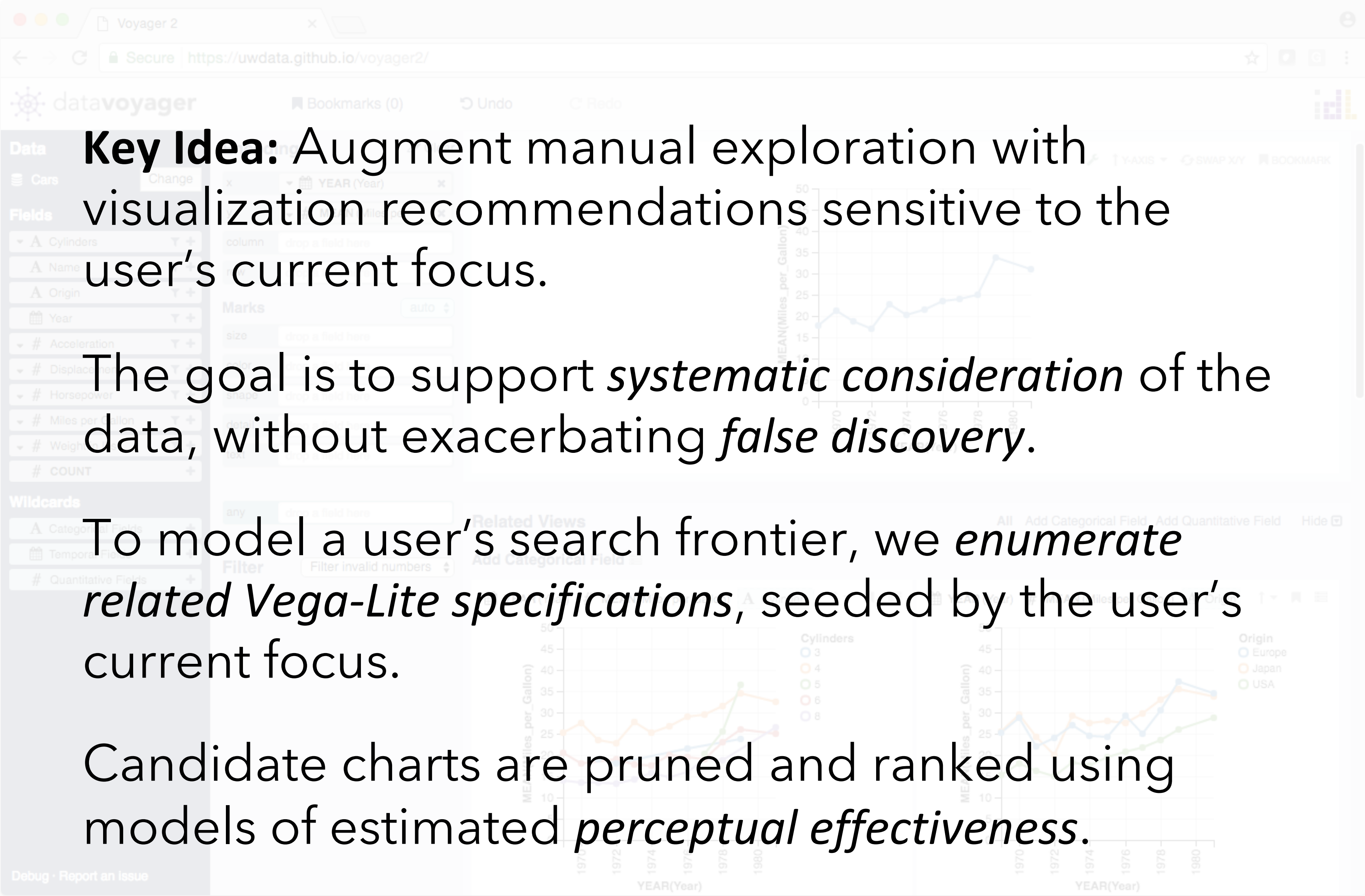


x: YEAR (Year)
y: MEAN (Miles per
column: drop a field here
row: drop a field here
Marks: auto
size: drop a field here
color: drop a field here
shape: drop a field here
detail: drop a field here
text: drop a field here
any: drop a field here
Filter: Filter invalid numbers

YEAR (Year) # MEAN (Miles per Gallon) A Cylinders

YEAR (Year) # MEAN (Miles per Gallon) A Origin

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



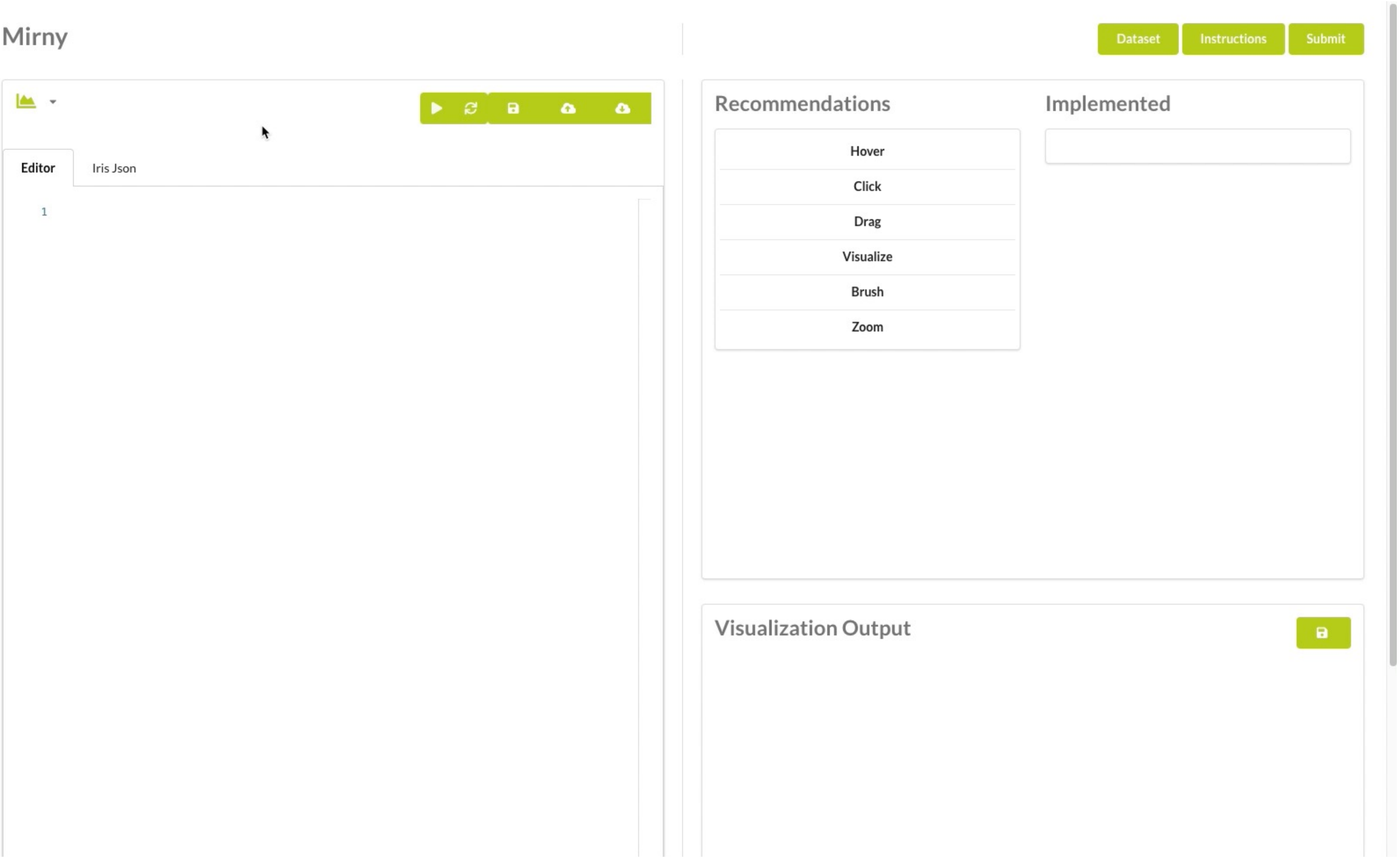
Key Idea: Augment manual exploration with visualization recommendations sensitive to the user's current focus.

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. (to appear in) *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.