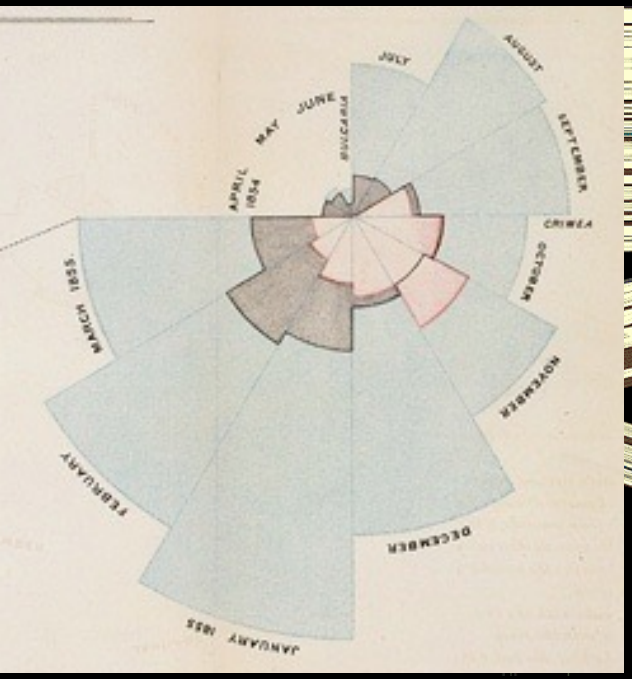
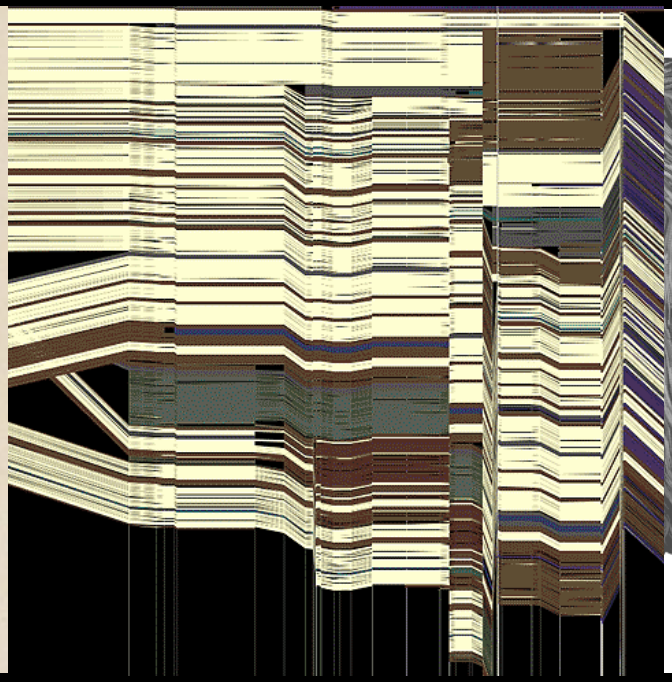
CSE 442 - Data Visualization

Visualization Tools





Jeffrey Heer University of Washington

How do people create visualizations?

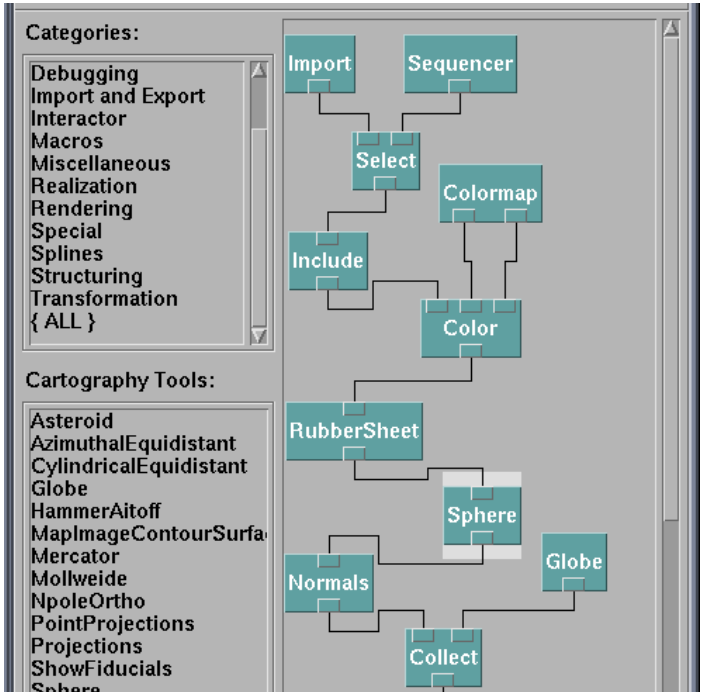


Chart Typology

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

Processing, OpenGL, Java2D

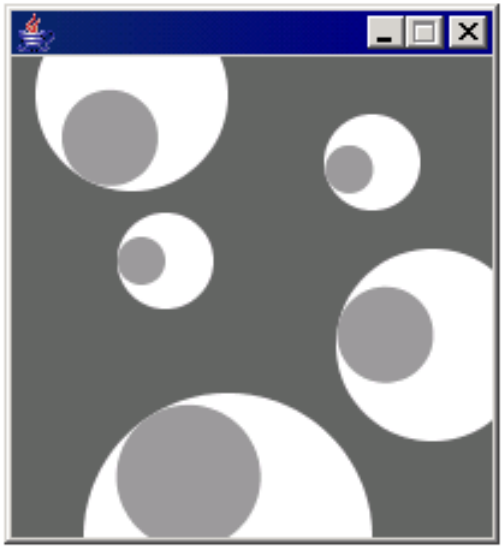
Processing - 0123 Beta

File Edit Sketch Tools Help

Run

sketch_070126a §

```
    ey = y;
    size = s;
  }

  void update(int mx, int my) {
    angle = atan2(my-ey, mx-ex);
  }

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



59

http://processing.org

US Air Traffic, Aaron Koblin

Graphics APIs

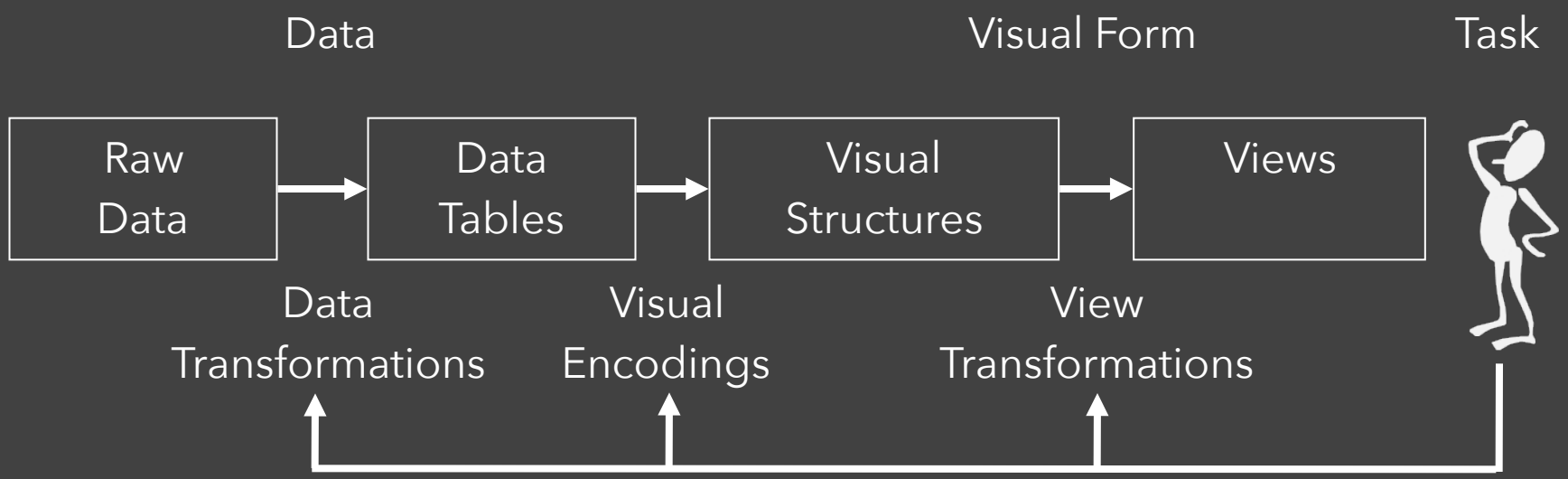
Processing, OpenGL, Java2D

Component Architectures

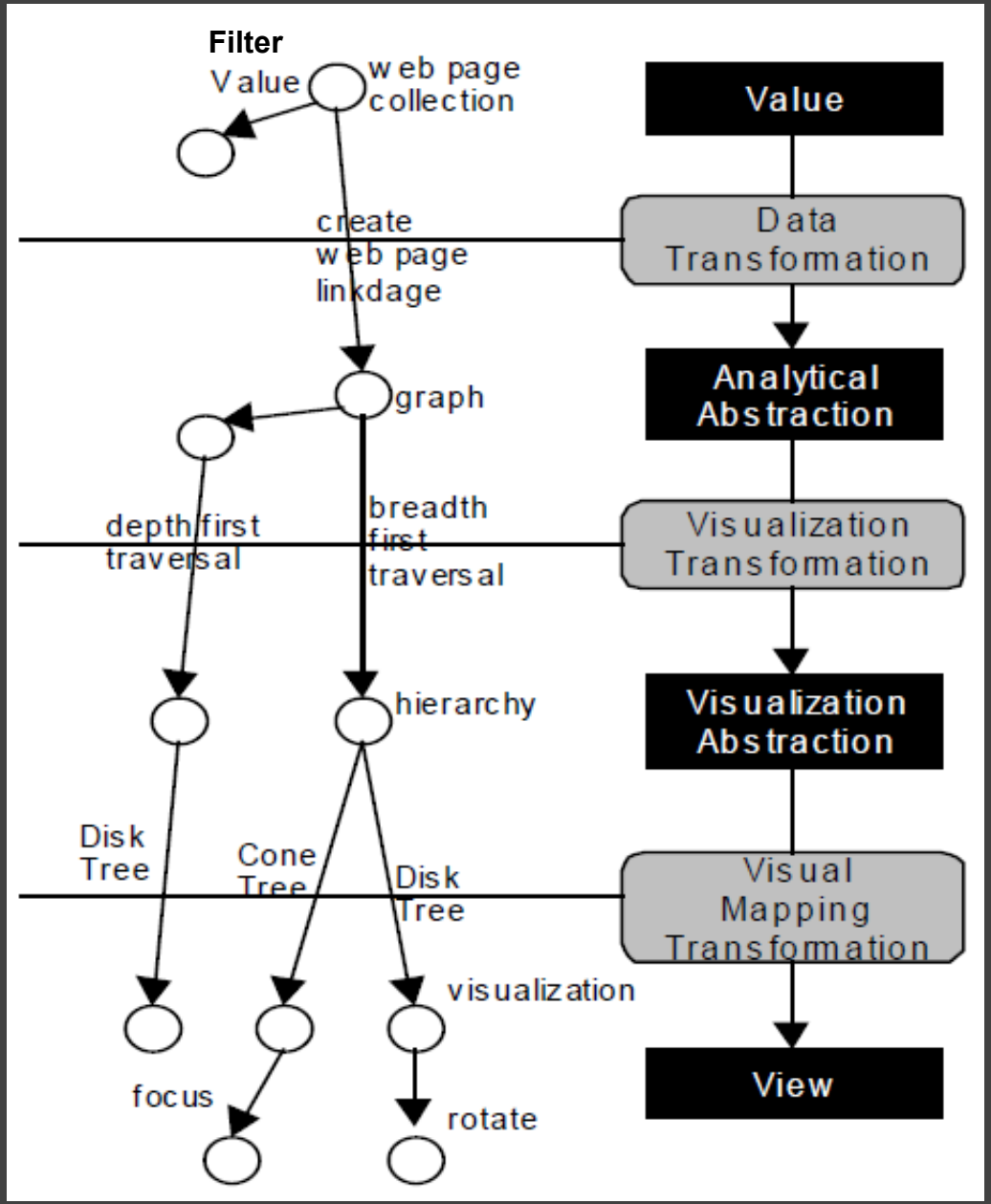
Prefuse, Flare, Improvise, VTK

Graphics APIs

Processing, OpenGL, Java2D


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

Data State Model
[Chi 98]



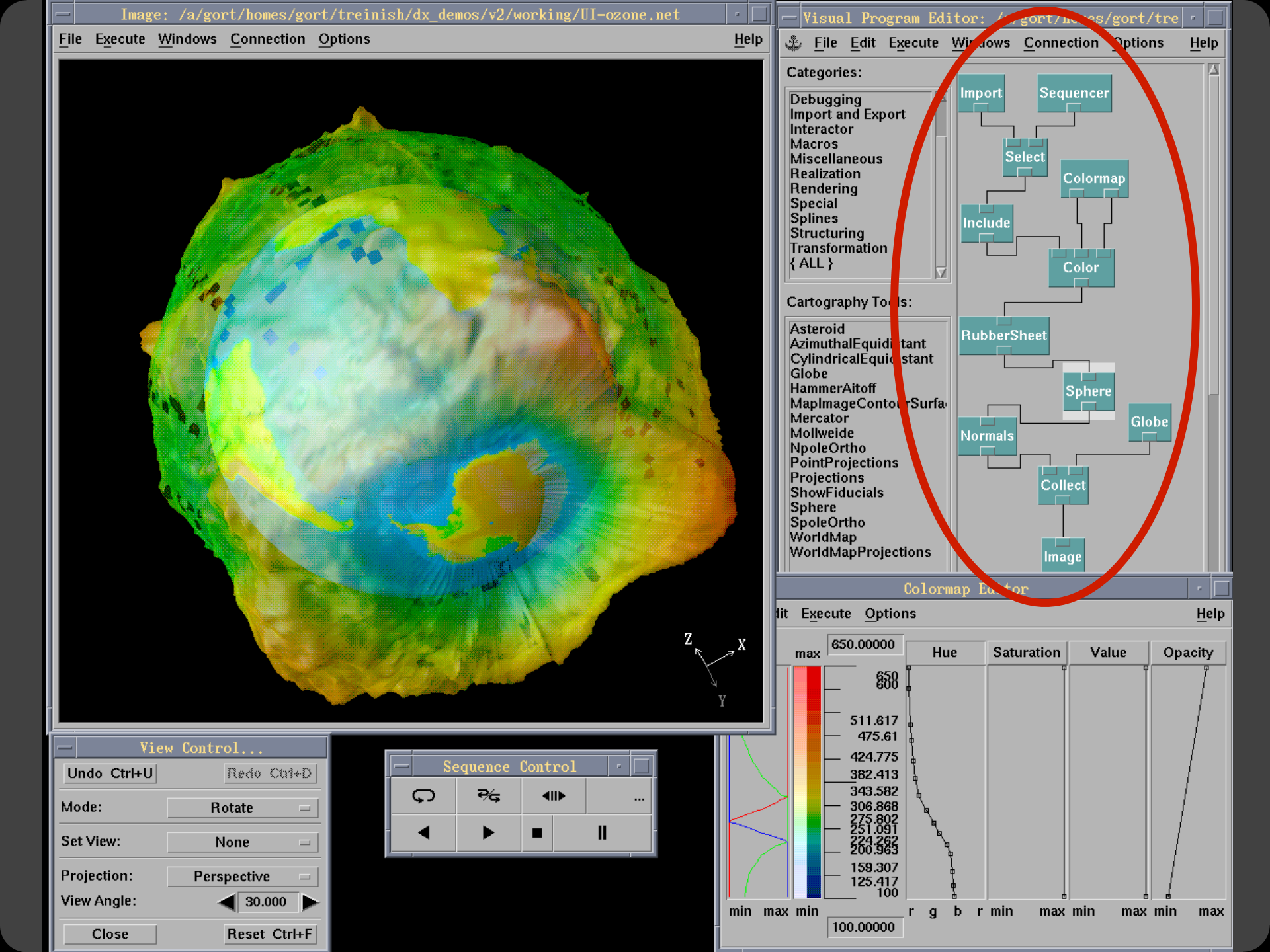
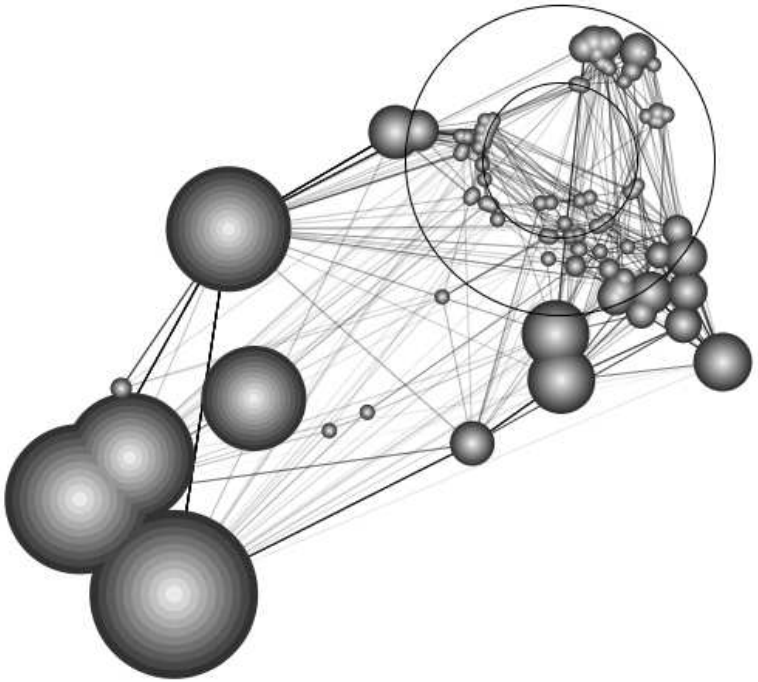


Image: /a/gort/homes/gort/treinish/dx_demos/v2/working/UI-ozone.net
File Execute Windows Connection Options Help
Visual Program Editor:
File Edit Execute Windows Connection Options Help
Categories:
Debugging
Import and Export
Interactor
Macros
Miscellaneous
Realization
Rendering
Special
Splines
Structuring
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

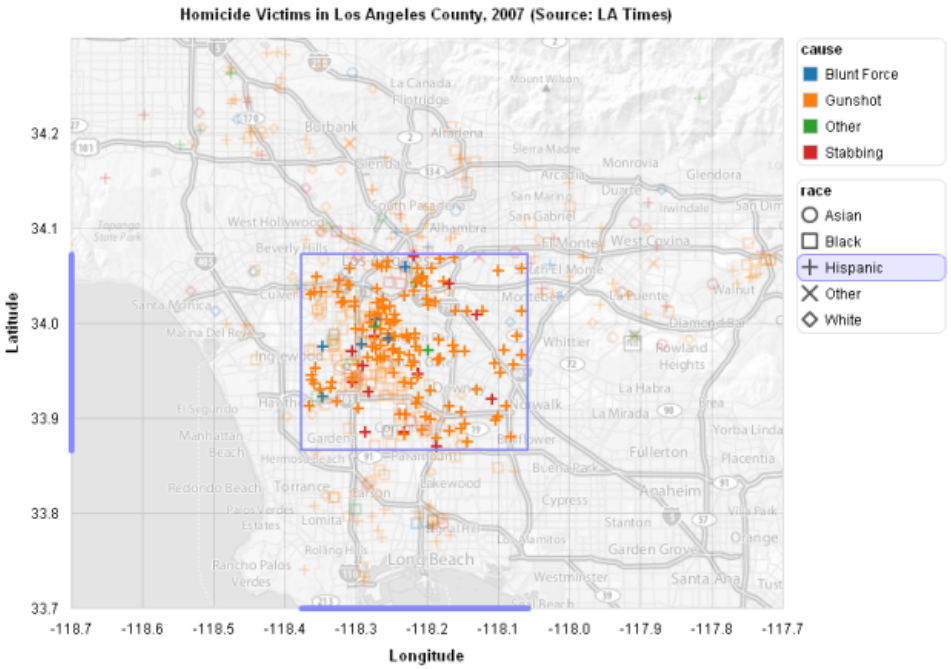
Prefuse & Flare

Operator-based toolkits for visualization design

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



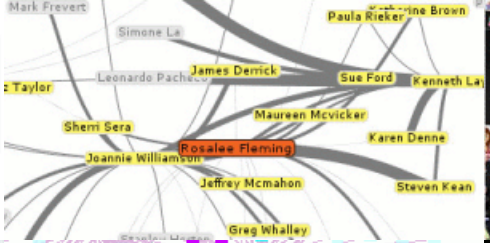
Prefuse (http://prefuse.org)

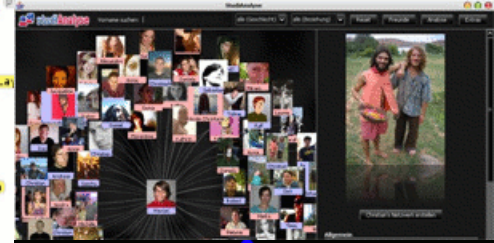


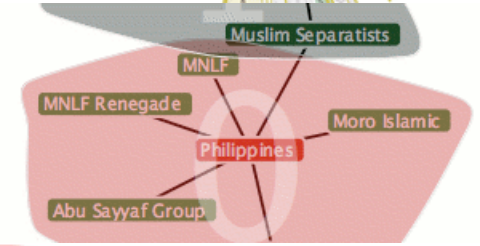
Flare (http://flare.prefuse.org)

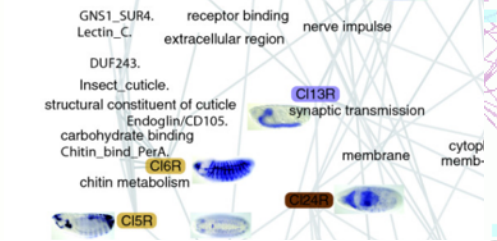

concept
idea
law
plan
kind
principle
model
play
plan of action
ideal
value

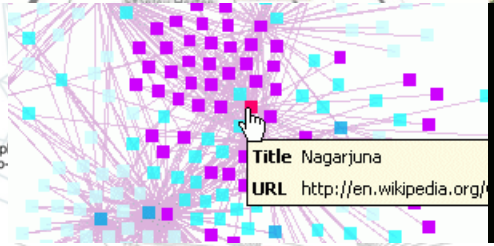

in the
at a conference
just
national park
in
banff

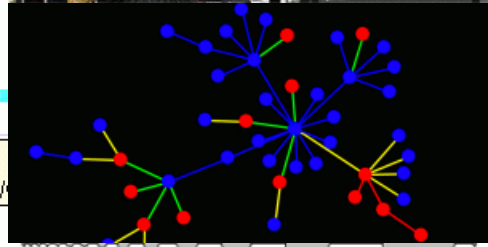

Mark Frevert
Paula Rieker
Simone La
James Derrick
Sue Ford
Kenneth La
Leonardo Pacheco
Sherri Sera
Maureen Mcvicker
Rosalee Fleming
Karen Denne
Joannie Williamson
Jeffrey Mcmahon
Steven Kean
Greg Whalley

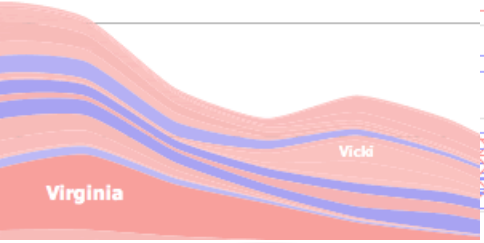



Muslim Separatists
MNLF
MNLF Renegade
Moro Islamic
Philippines
Abu Sayyaf Group

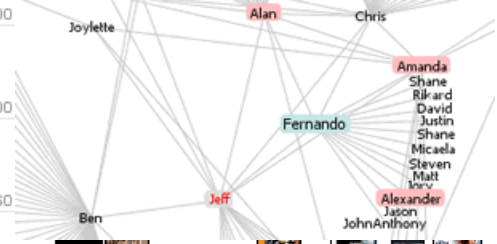

GNS1_SUR4.
Lectin_C.
receptor binding
extracellular region
nerve impulse
DUF243.
Insect_cuticle.
structural constituent of cuticle
Endoglin/CD105.
carbohydrate binding
Chitin_bind_PerA.
chitin metabolism
membrane
synaptic transmission
Cl13R
Cl6R
Cl24R
Cl5R

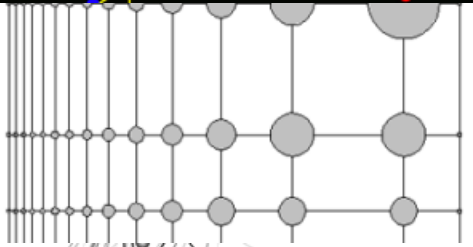

Title Nagarjuna
URL http://en.wikipedia.org/

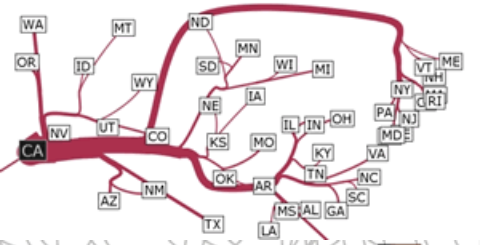


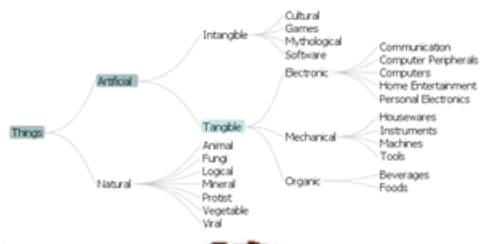

Virginia
Vicki


$2,000,000
$1,000,000
$0
NY OK PA RI SD TX VA VT WI WY

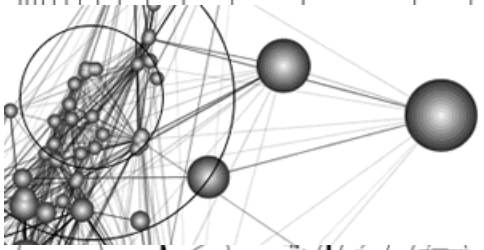

Joylette
Alan
Chris
Amanda
Shane
Rikard
David
Justin
Shane
Micaela
Steven
Matt
Fernando
Jeff
Alexander
Jason
JohnAnthony
Ben

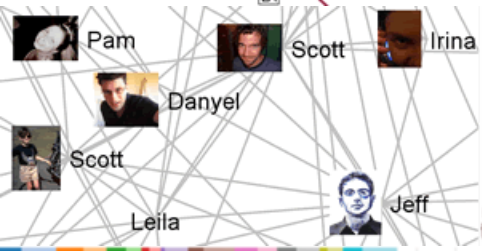



WA
MT
ND
MN
OR
ID
SD
WI
MI
VT
NH
ME
WY
NY
RI
NE
IA
PA
NJ
IL
IN
OH
NV
UT
CO
KS
MO
MD
DE
CA
KY
VA
TN
NC
OK
AR
NM
SC
AZ
MS
AL
GA
TX
LA


Things
Artificial
Natural
Intangible
Tangible
Electronic
Mechanical
Organic
Cultural
Games
Mythological
Software
Communication
Computer Peripherals
Computers
Home Entertainment
Personal Electronics
Housewares
Instruments
Machines
Tools
Animal
Fungi
Logical
Mineral
Protist
Vegetable
Viral
Beverages
Foods

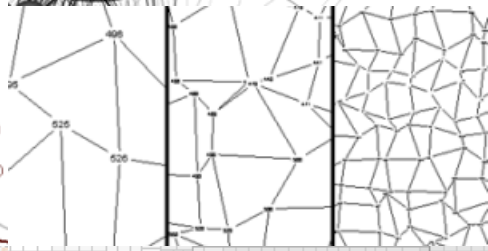

M. M. Salim

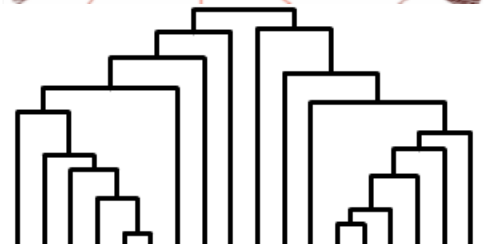


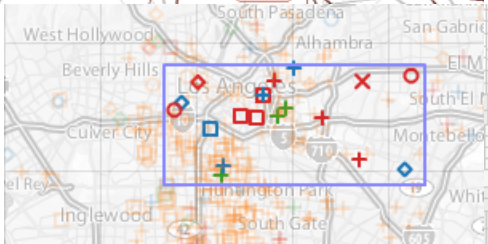

Pam
Scott
Irina
Danyel
Scott
Jeff
Leila

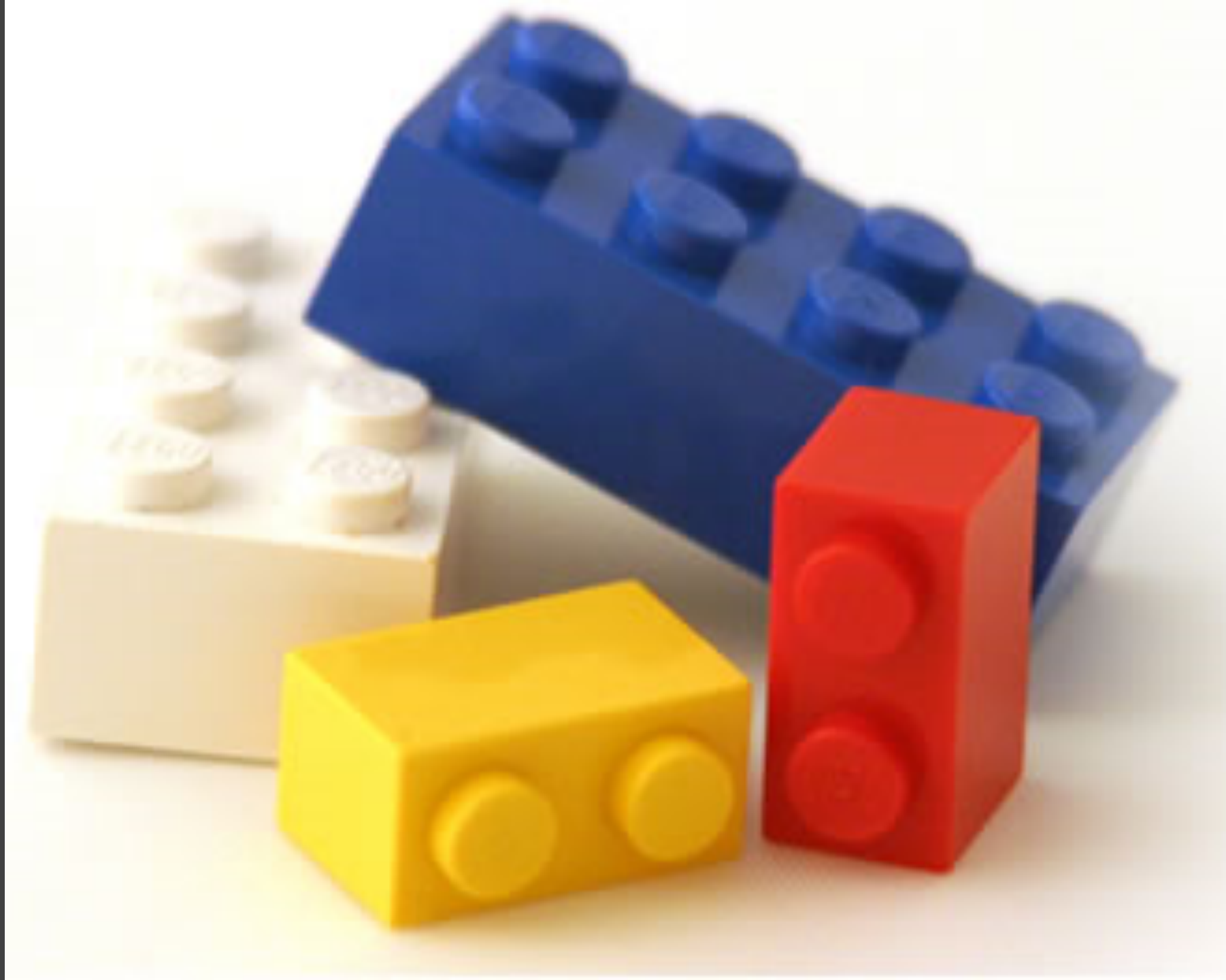

Ideology
Science
Humanities
Technology
Knowledge
Recurring
Historical
The Millenium
Events
Generic
Civilizations
Individuals
People
S. People
Organizations
Categories
Natural
Things
Constellations

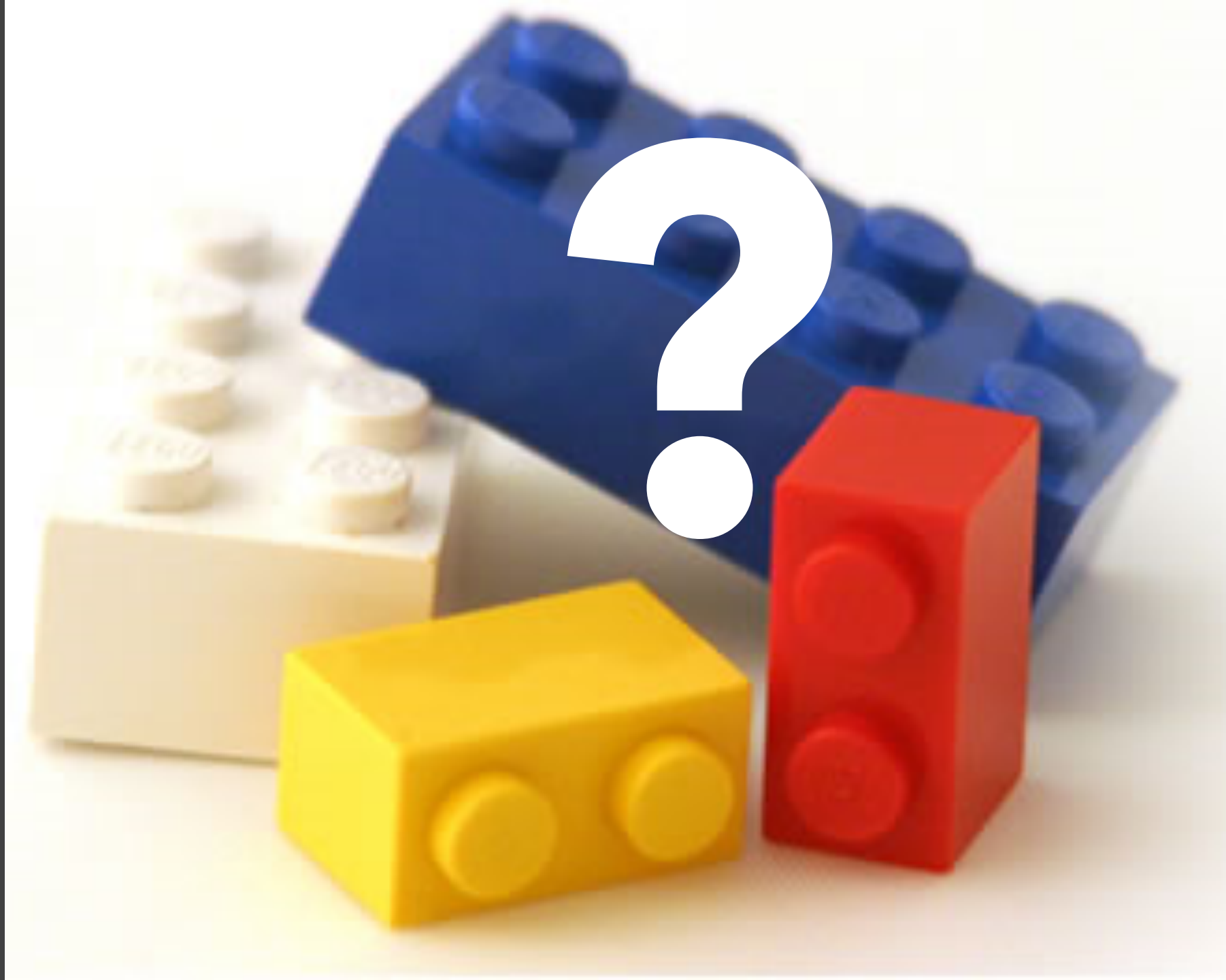

A. Shaikh
Nawaf Alhazmi
Ani
Lotfi Karoui
H. Z. Moustaoui
Majed Moqed
R. M. Abdallah
Hani Hanjour
Mohamed Atta
Faisal Al Salmi
Z. Essabar
Ramzi Omar
Sugami
H. Alghamdi
M. Alshehri
W. M. Alshehri
M. A. Shehri


496
526
526




West Hollywood
Beverly Hills
Culver City
Inglewood
South Pasadena
Alhambra
San Gabriel
Montebello
Huntington Park
South Gate





Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Processing, OpenGL, Java2D

Chart Typologies

Excel, Many Eyes, Google Charts

Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Processing, OpenGL, Java2D

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.

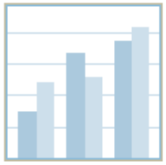
Learn more

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

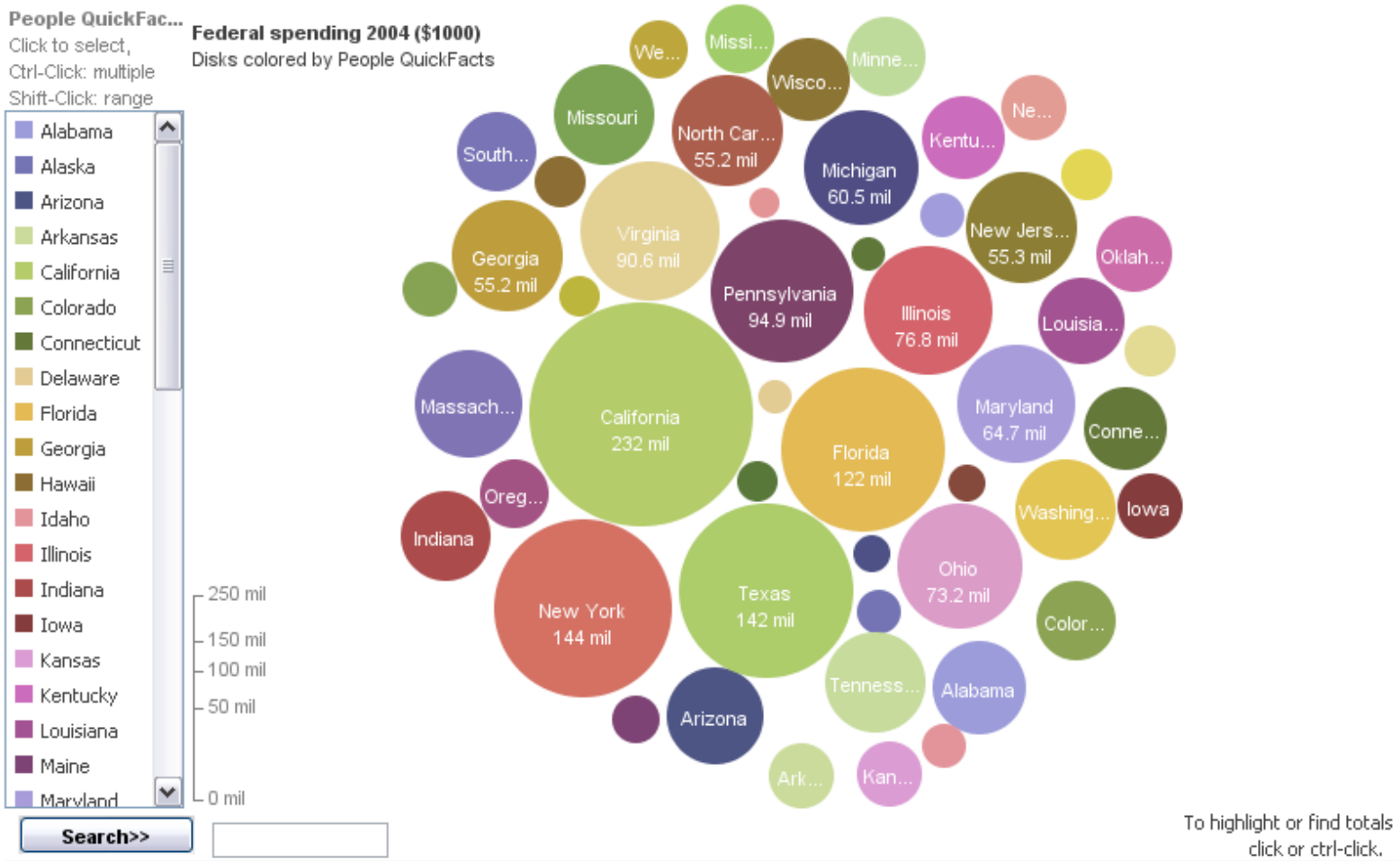
Block Histogram

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 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

Processing, OpenGL, Java2D

Chart Typologies

Excel, Many Eyes, Google Charts

Visual Analysis Grammars

VizQL, ggplot2

Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Processing, OpenGL, Java2D

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:

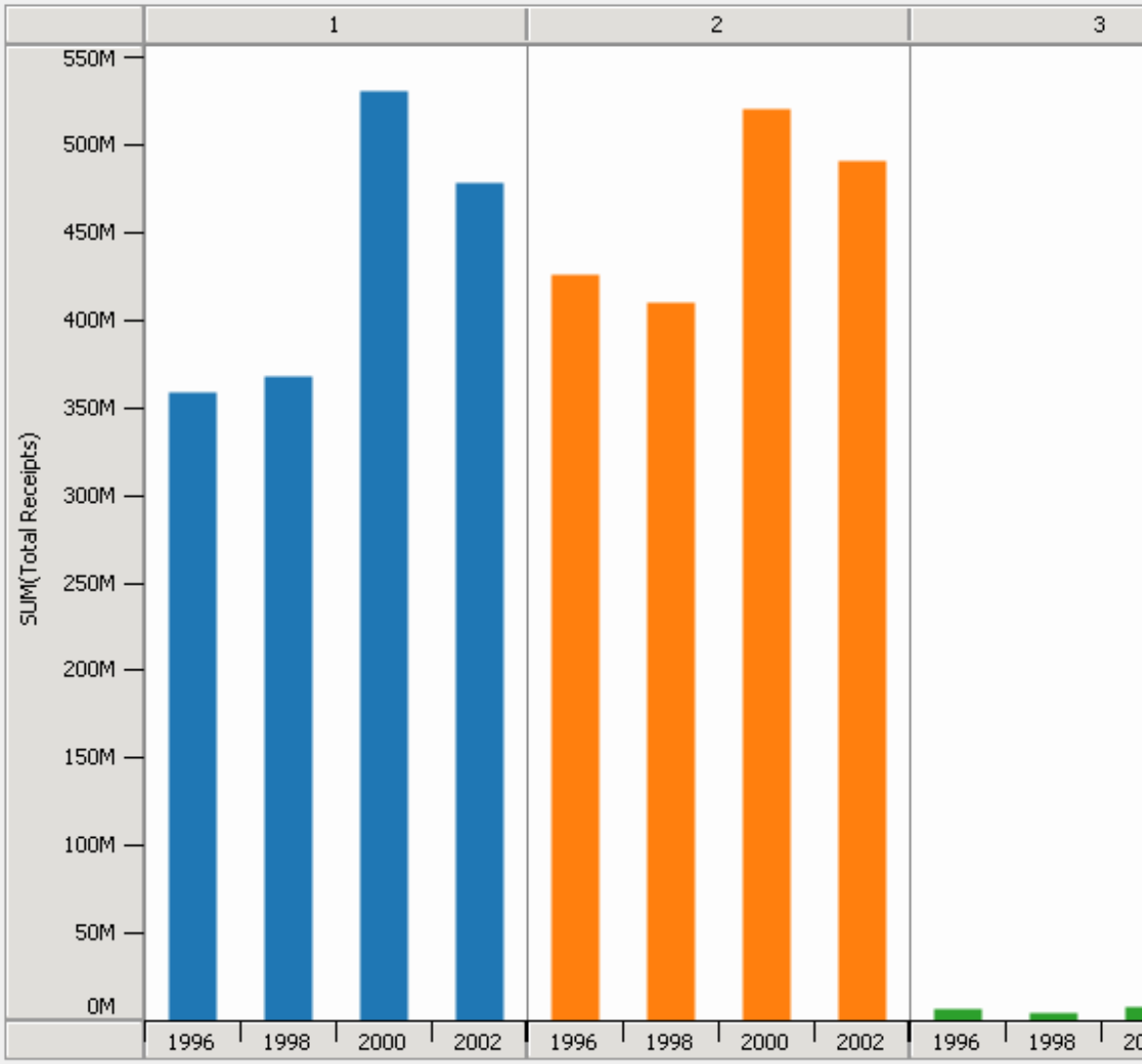
Legend:

1
2
3

Size:

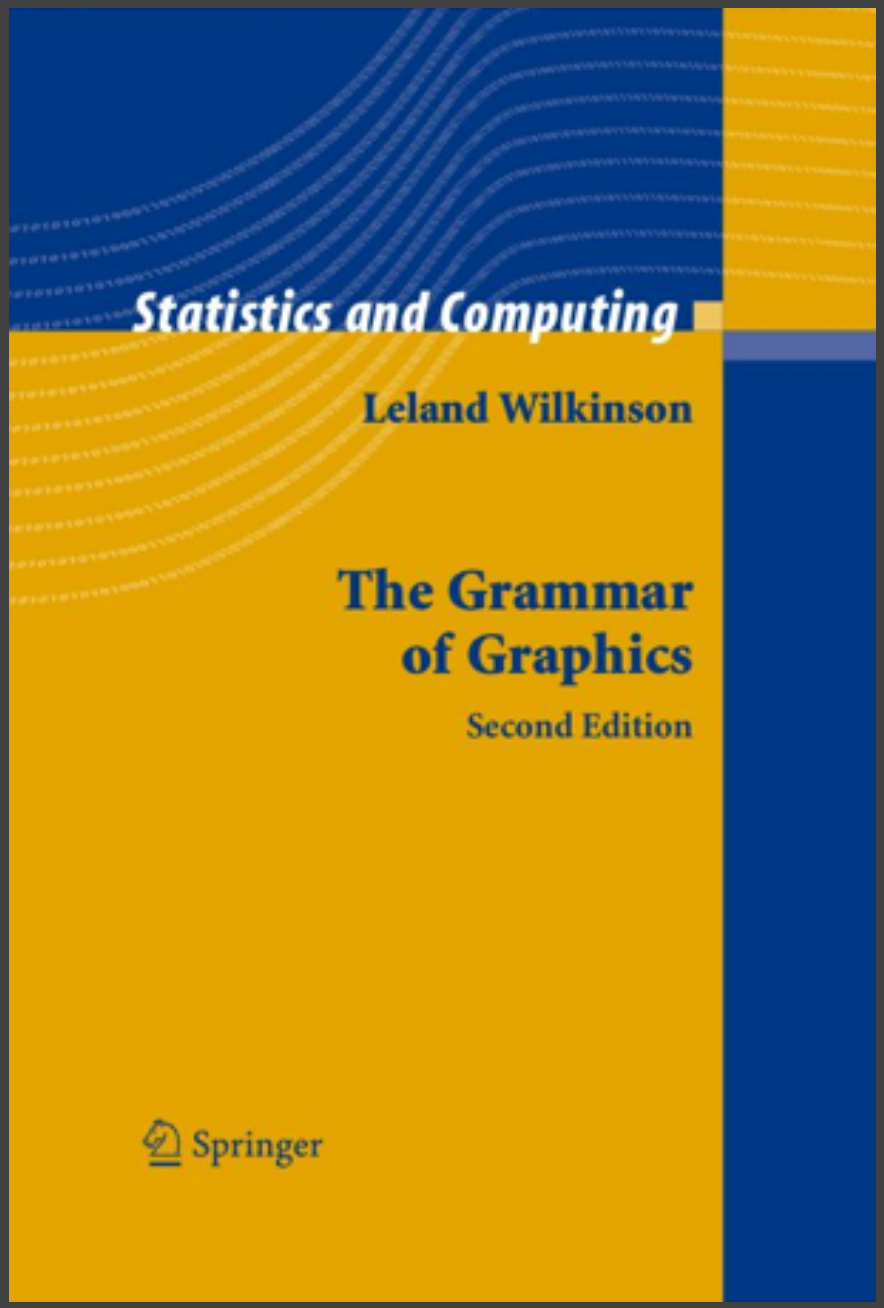
Columns: Party | Year

Rows: SUM(Total..

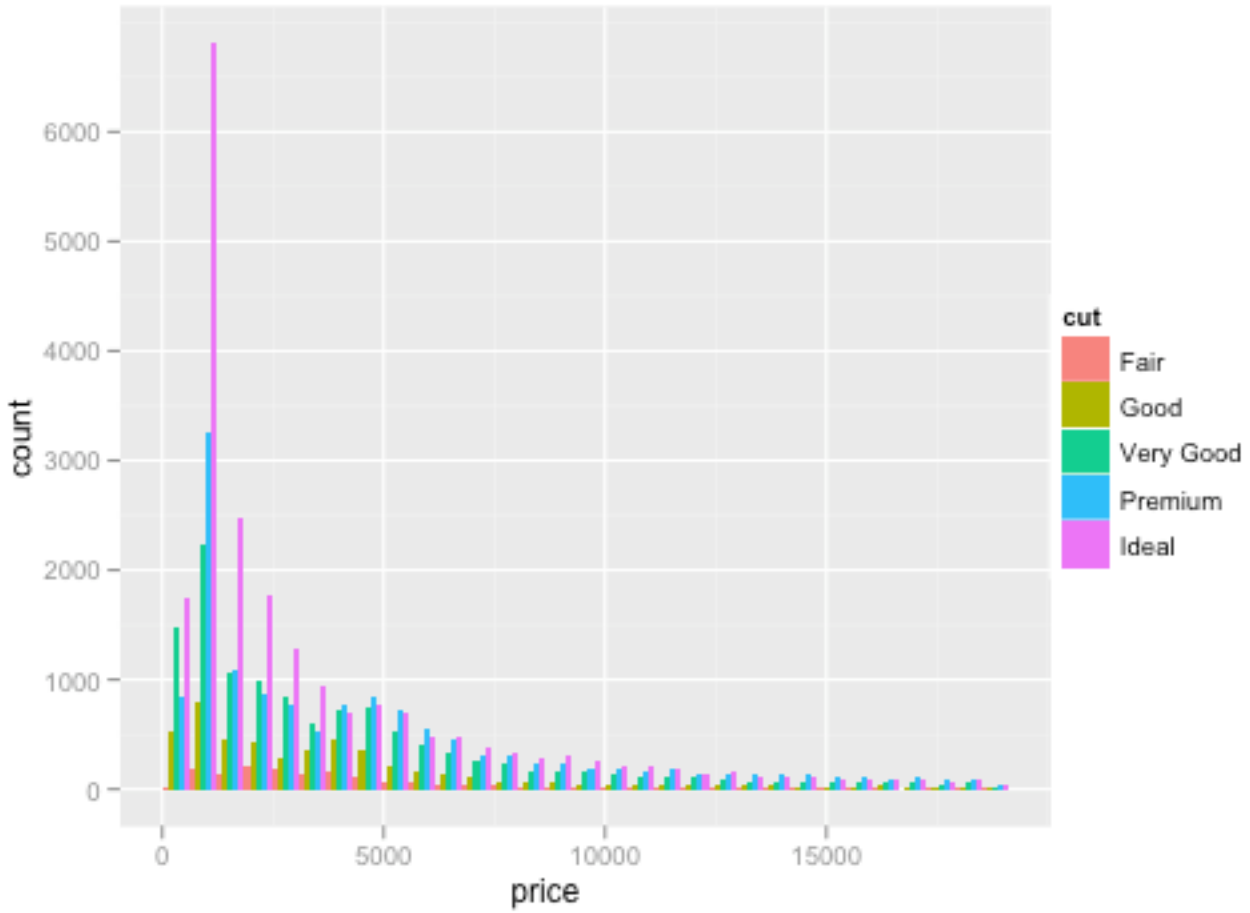


Sheet 1

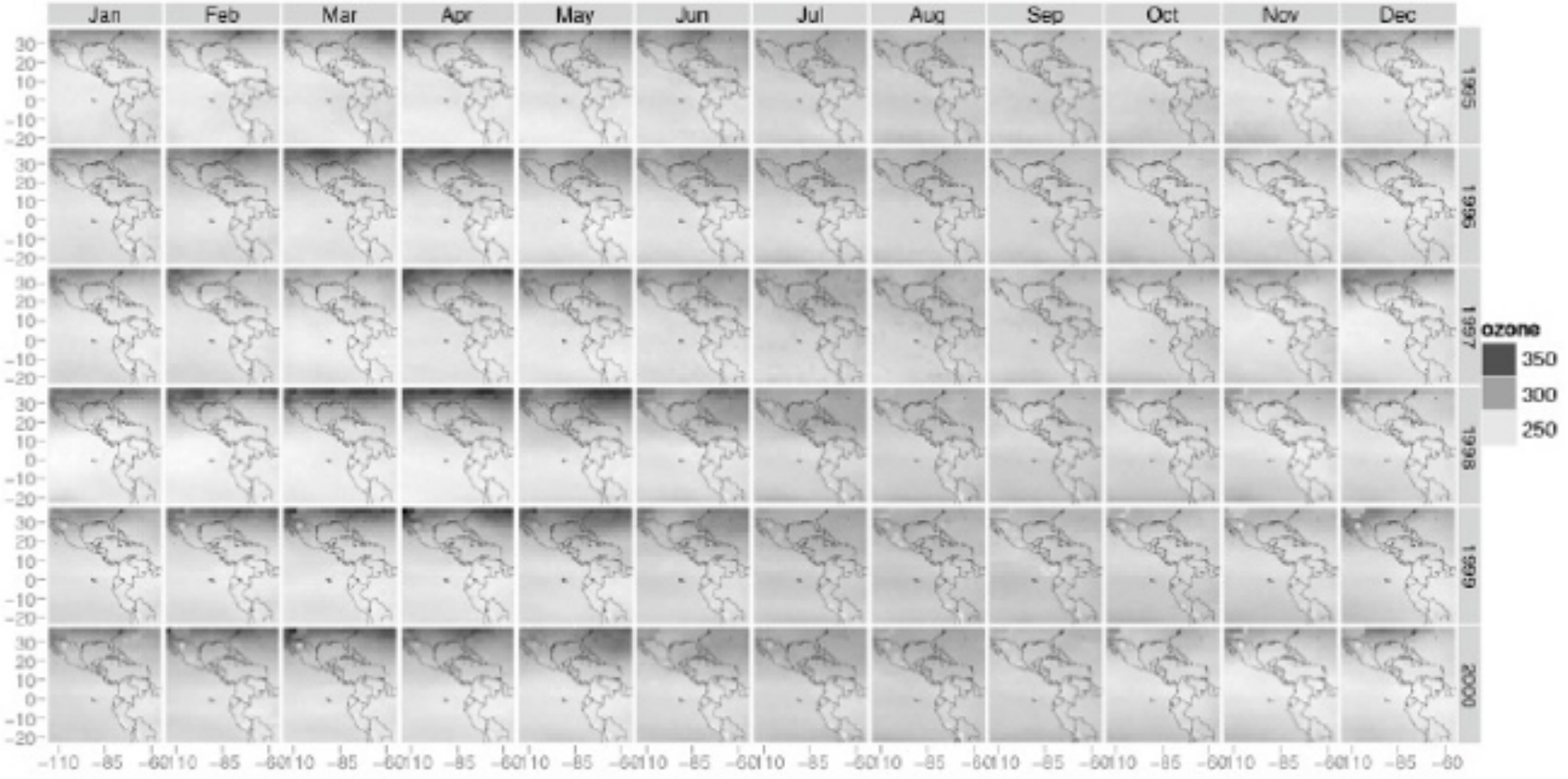
Ready


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
```

Chart Typologies

Excel, Many Eyes, Google Charts

Visual Analysis Grammars

VizQL, ggplot2

Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Processing, OpenGL, Java2D

Chart Typologies

Excel, Many Eyes, Google Charts

Visual Analysis Grammars

VizQL, ggplot2

Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Processing, OpenGL, Java2D

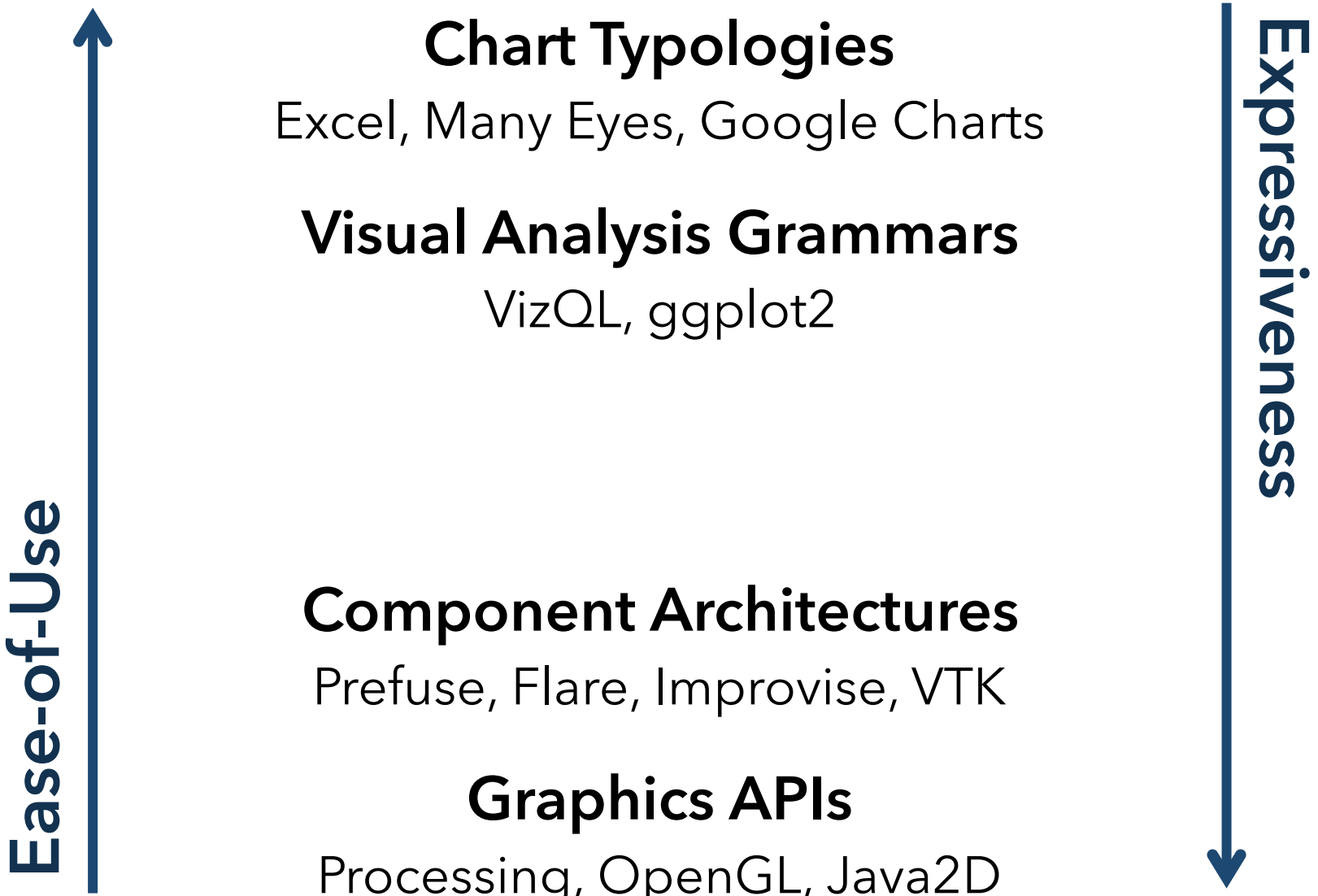

Ease-of-Use
Expressiveness
Chart Typologies
Excel, Many Eyes, Google Charts
Visual Analysis Grammars
VizQL, ggplot2
Component Architectures
Prefuse, Flare, Improvise, VTK
Graphics APIs
Processing, OpenGL, Java2D

Chart Typologies

Excel, Many Eyes, Google Charts

Visual Analysis Grammars

VizQL, ggplot2

?

Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Processing, OpenGL, Java2D

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

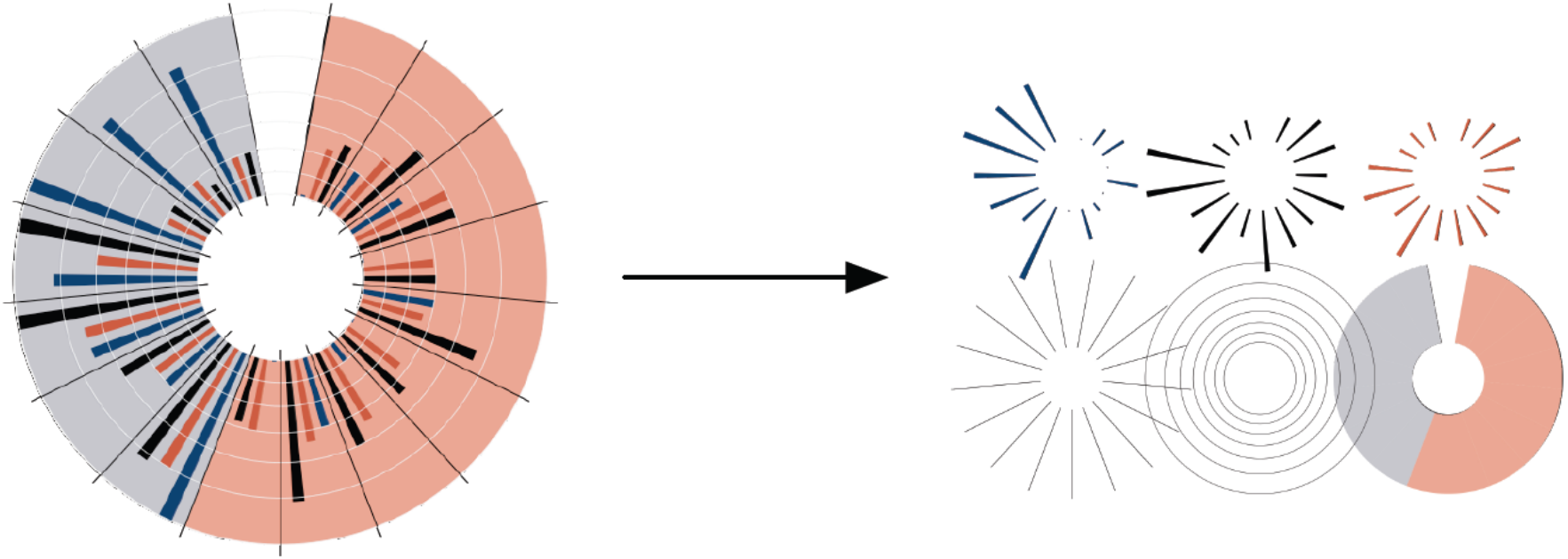
Processing, OpenGL, Java2D

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**.

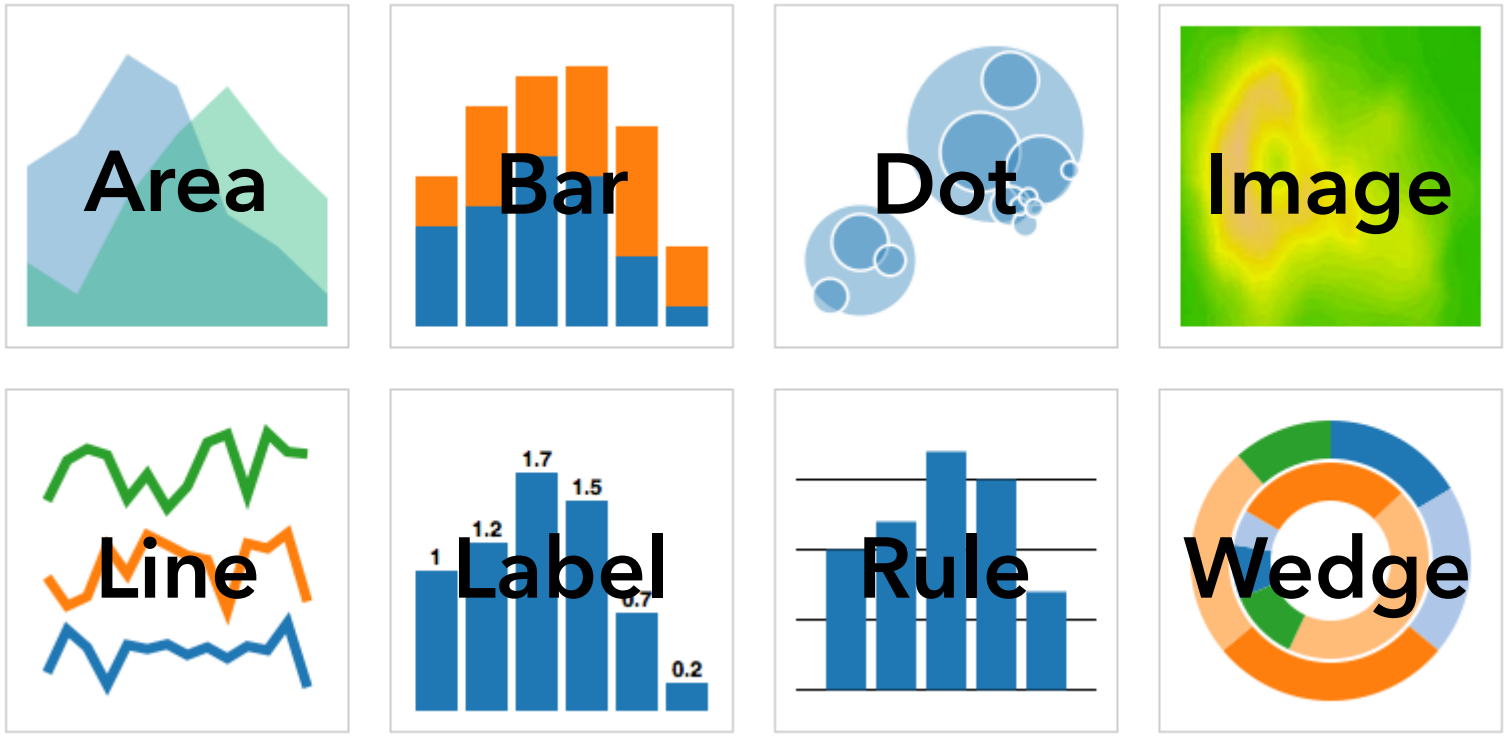
J. W. Tukey, M. B. Wilk
Data Analysis & Statistics, 1965

Protovis: A Grammar for Visualization



A graphic is a composition of data-representative marks.

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

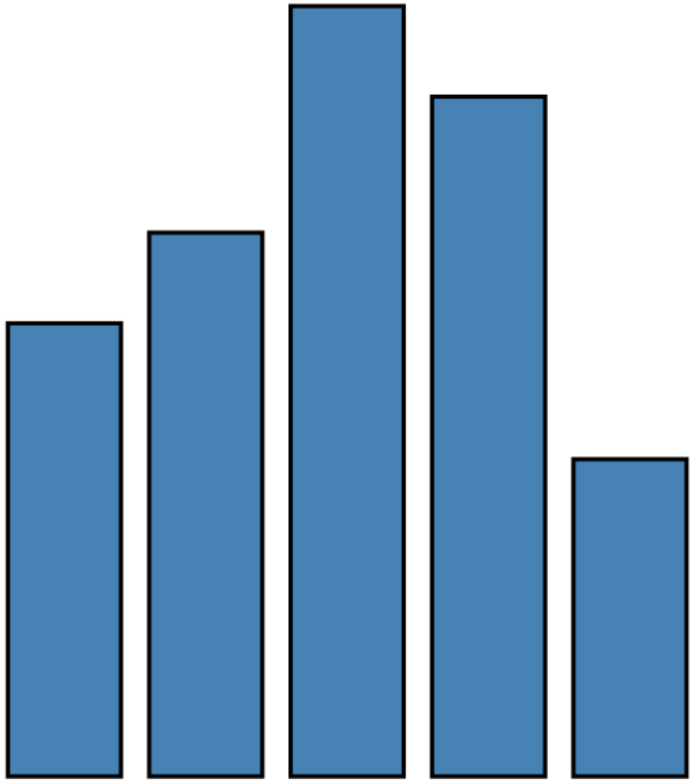


MARKS: Protovis graphical primitives

MARK	$\lambda : D \rightarrow R$
data	λ
visible	λ
left	λ
bottom	λ
width	λ
height	λ
fillStyle	λ
strokeStyle	λ
lineWidth	λ
...	λ

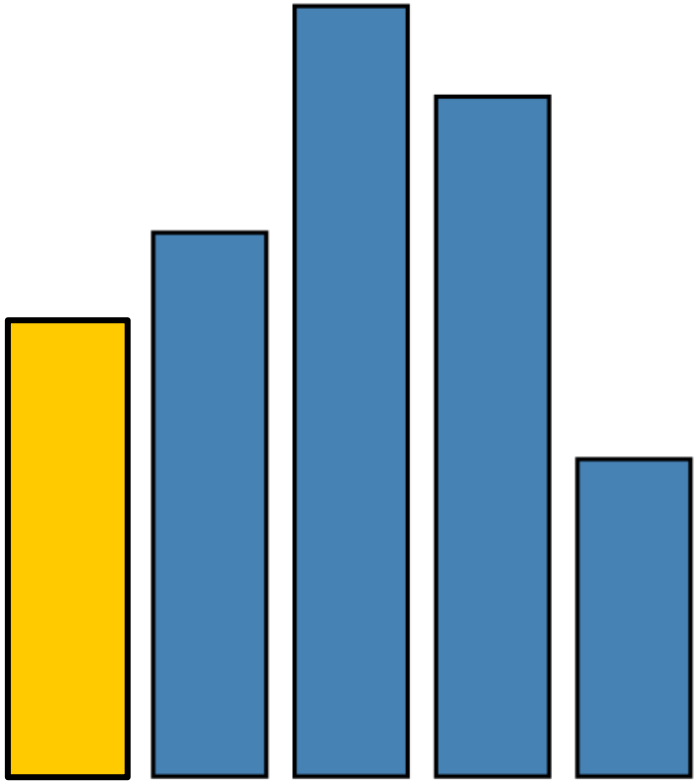
RECT $\lambda : D \rightarrow R$

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				
...	...				



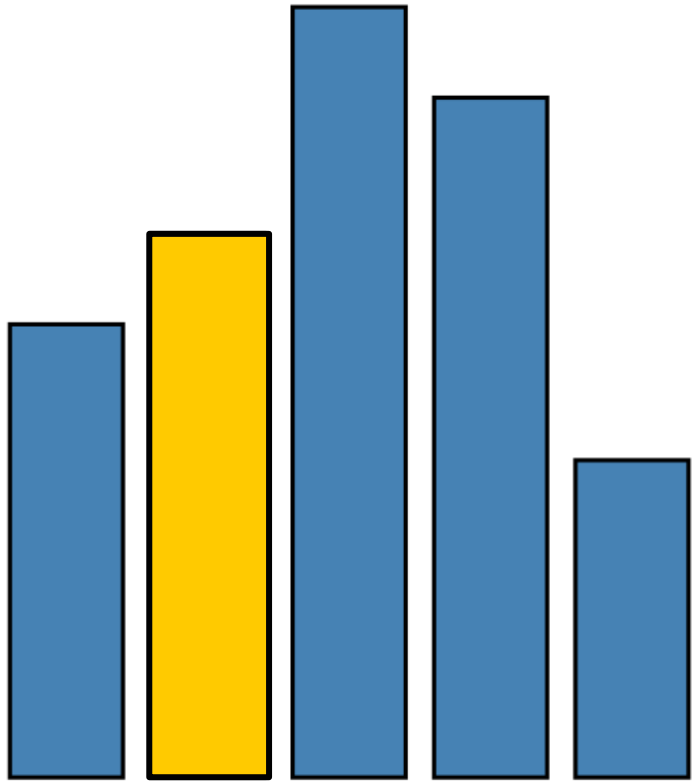
RECT $\lambda : D \rightarrow R$

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				
...	...				



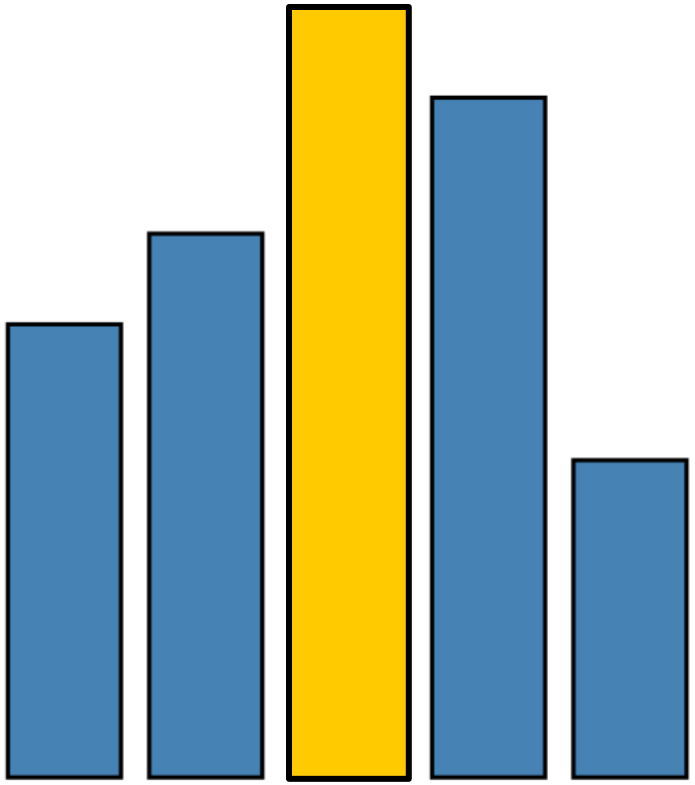
RECT $\lambda : D \rightarrow R$

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				
...	...				



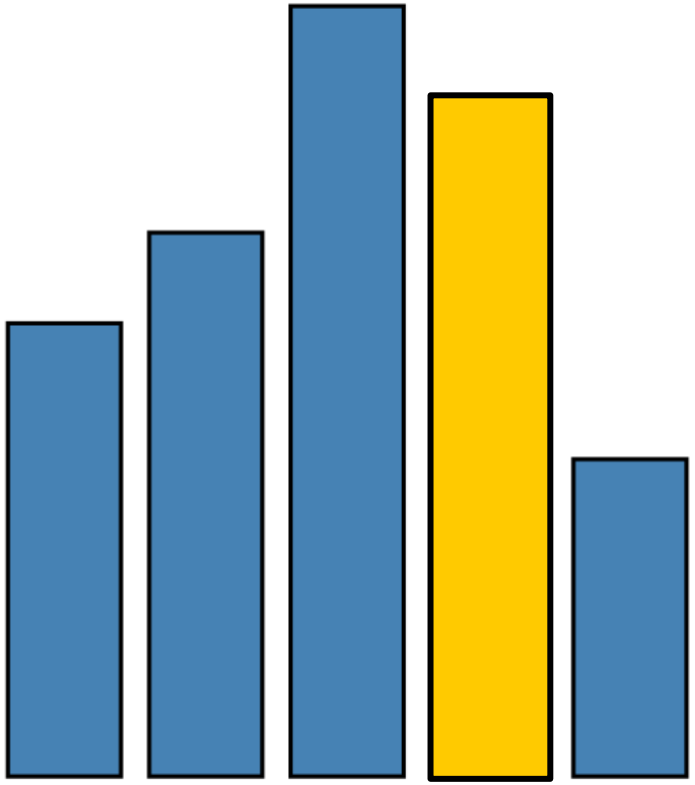
RECT $\lambda : D \rightarrow R$

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				
...	...				



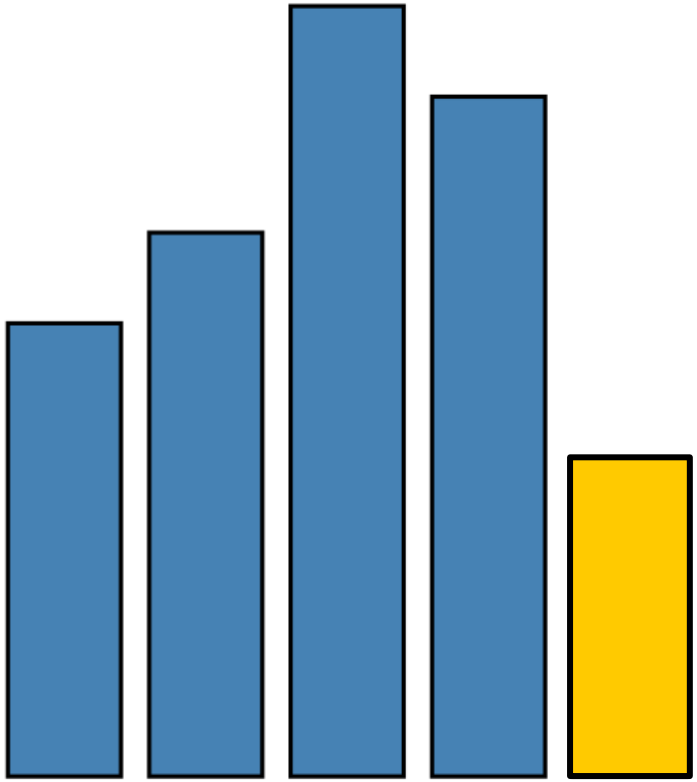
RECT $\lambda : D \rightarrow R$

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				
...	...				



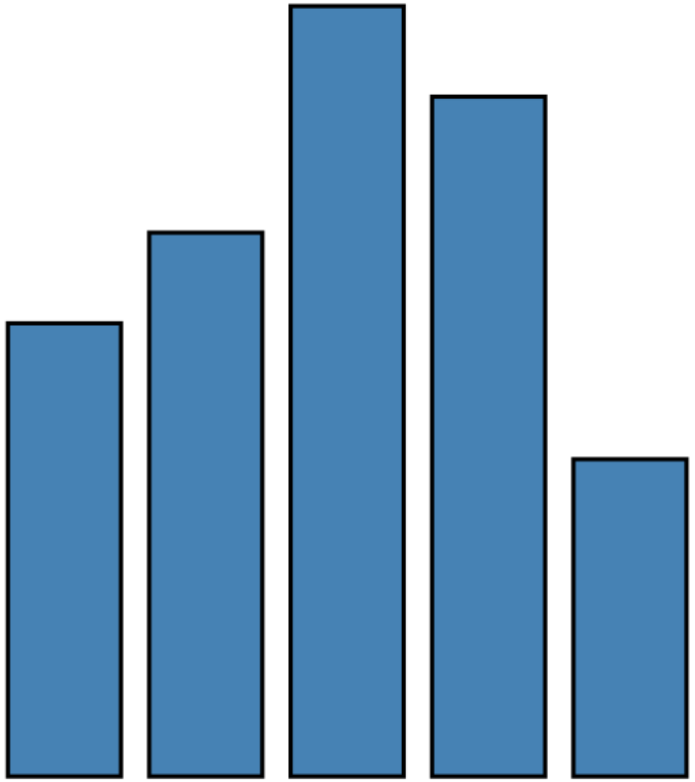
RECT $\lambda : D \rightarrow R$

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				
...	...				

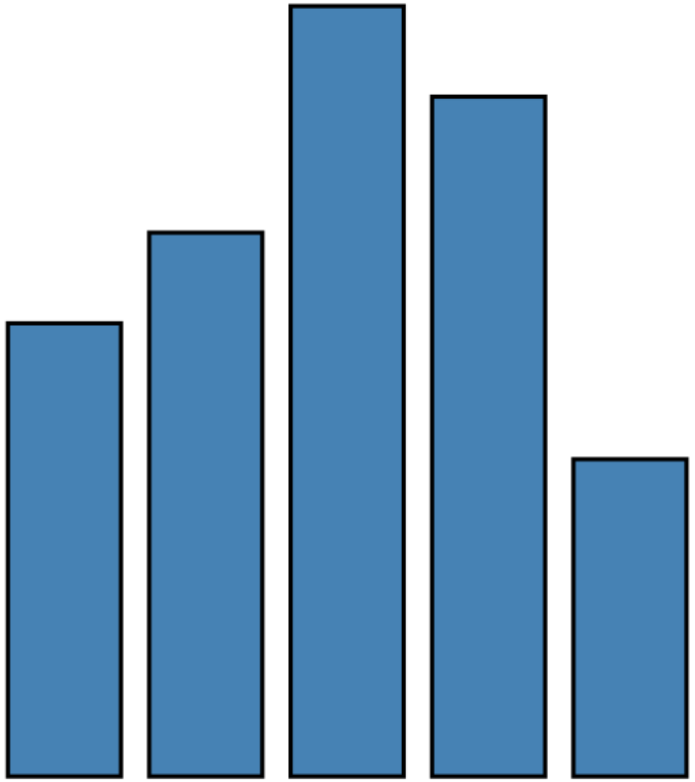


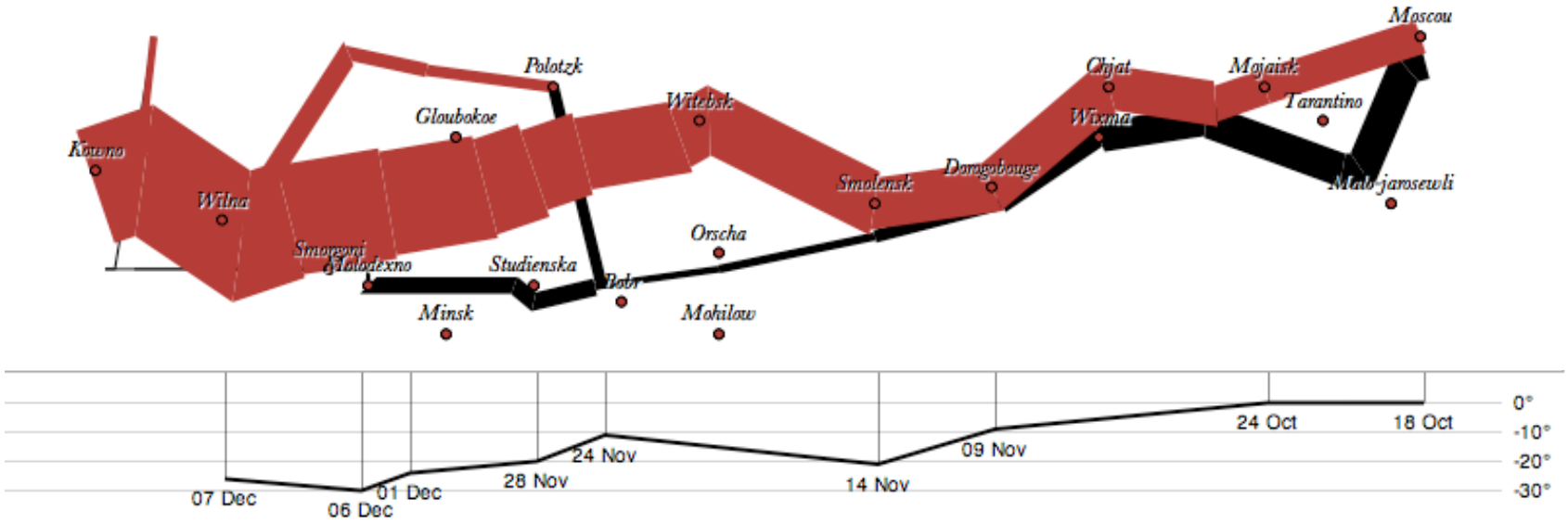
RECT $\lambda : D \rightarrow R$

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				
...	...				



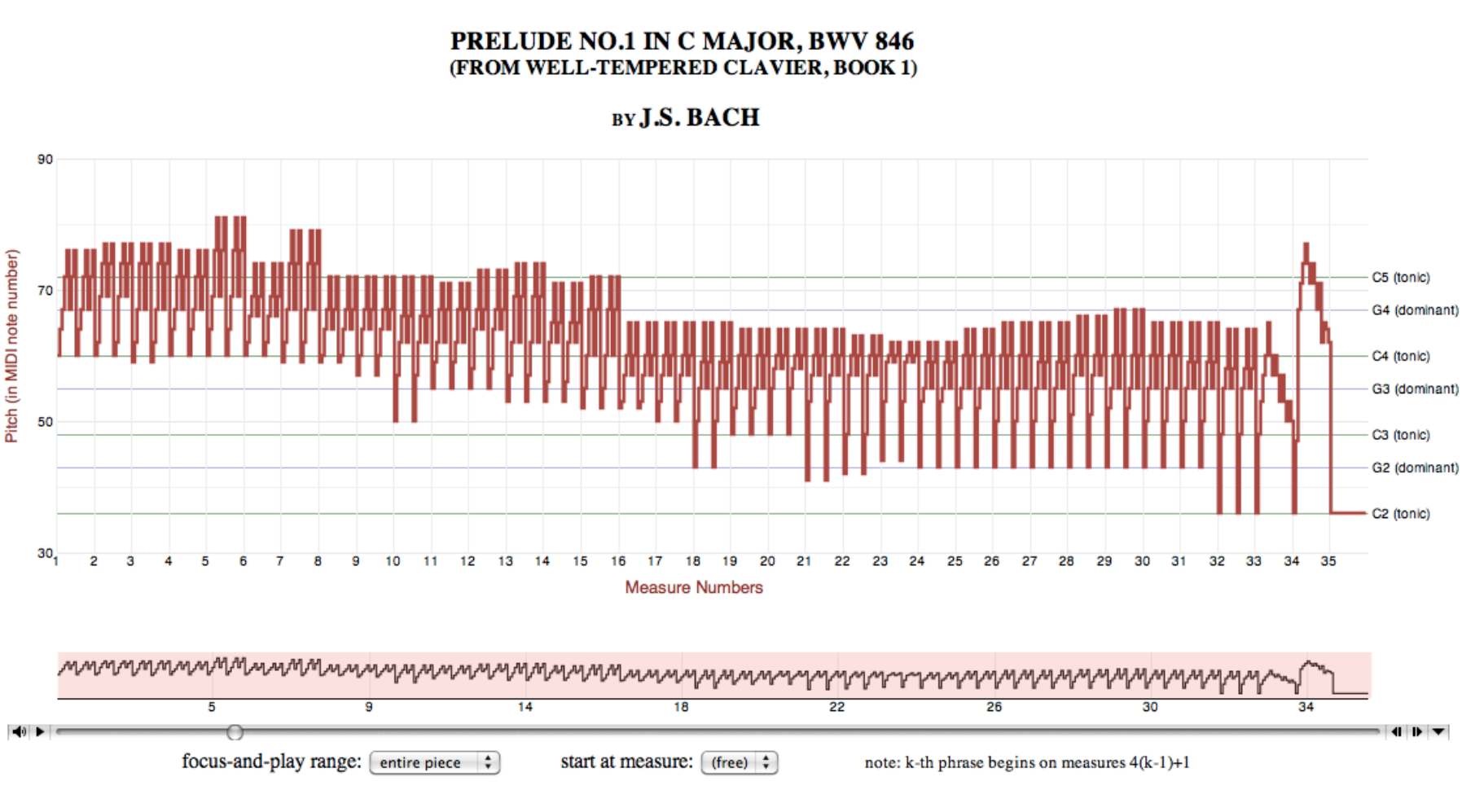
```
var vis = new pv.Panel();
vis.add(pv.Bar)
  .data([1, 1.2, 1.7, 1.5, 0.7])
  .visible(true)
  .left((d) => this.index * 25);
  .bottom(0)
  .width(20)
  .height((d) => d * 80)
  .fillStyle("blue")
  .strokeStyle("black")
  .lineWidth(1.5);
vis.render();
```



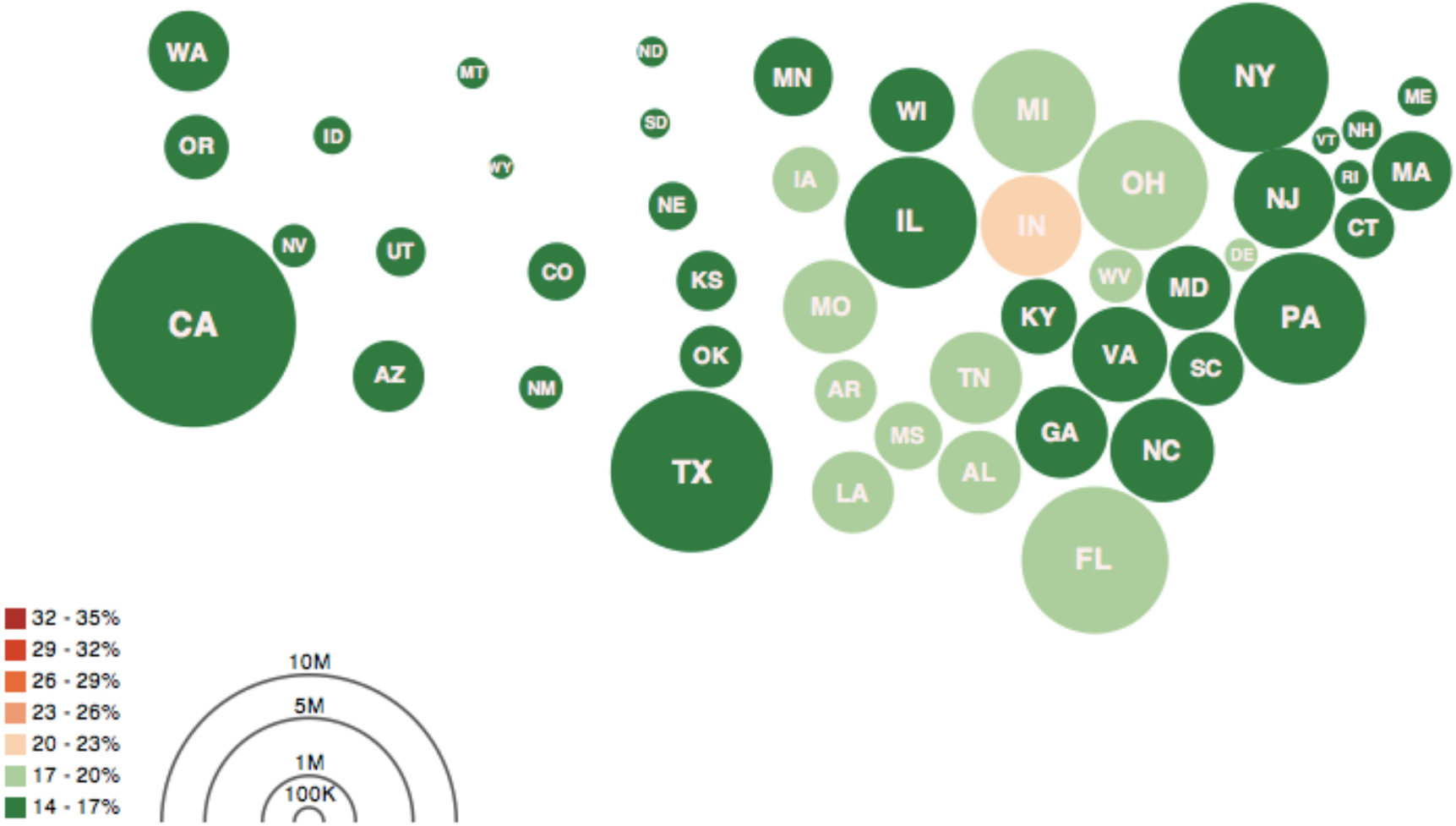



```
var army = pv.nest(napoleon.army, "dir", "group");
var vis = new pv.Panel();

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");
```

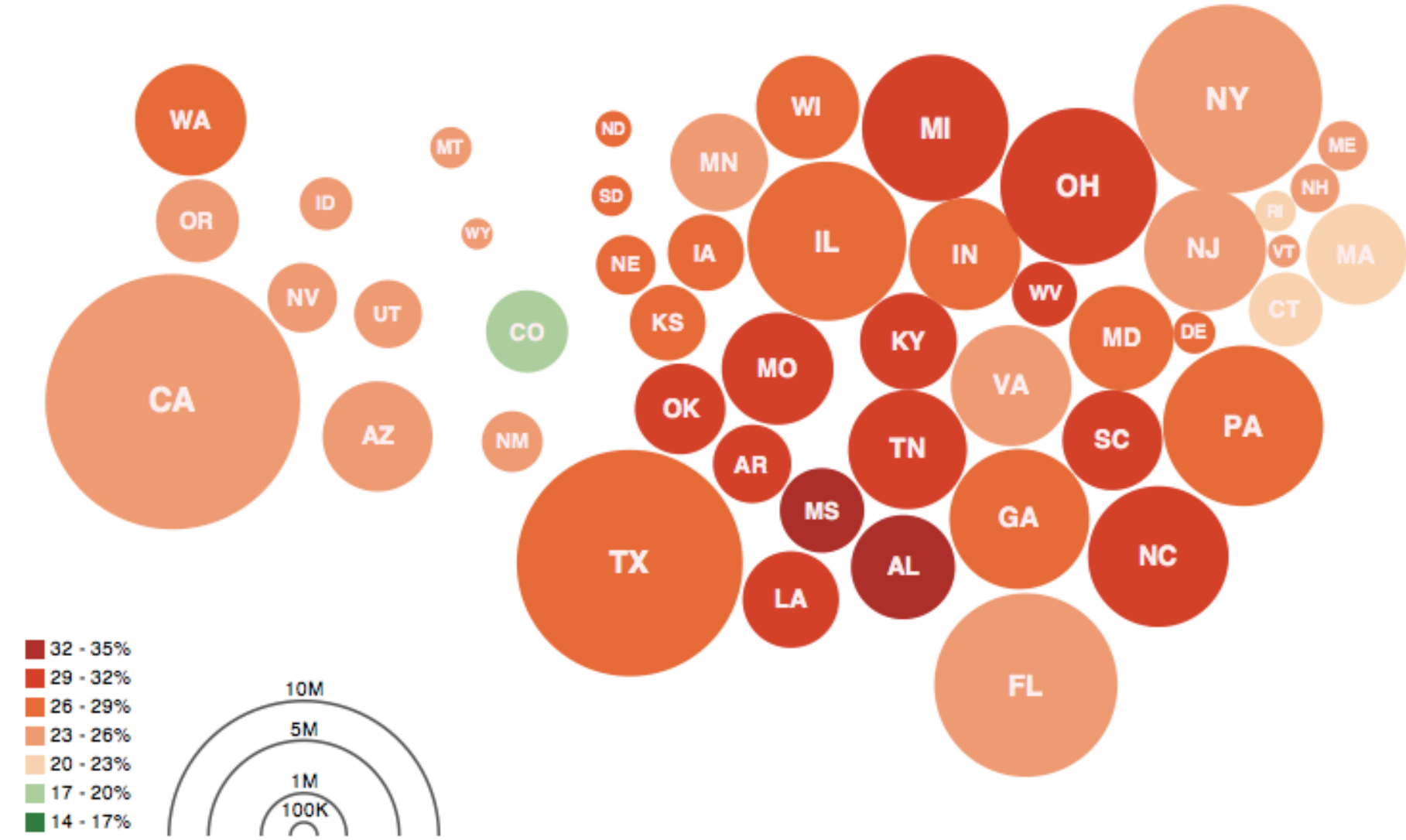
```
vis.add(pv.Rule).data([0,-10,-20,-30])
  .top((d) => 300 - 2*d - 0.5).left(200).right(150)
  .lineWidth(1).strokeStyle("#ccc")
  .anchor("right").add(pv.Label)
    .font("italic 10px Georgia")
    .text((d) => d+"°").textBaseline("center");

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");
```



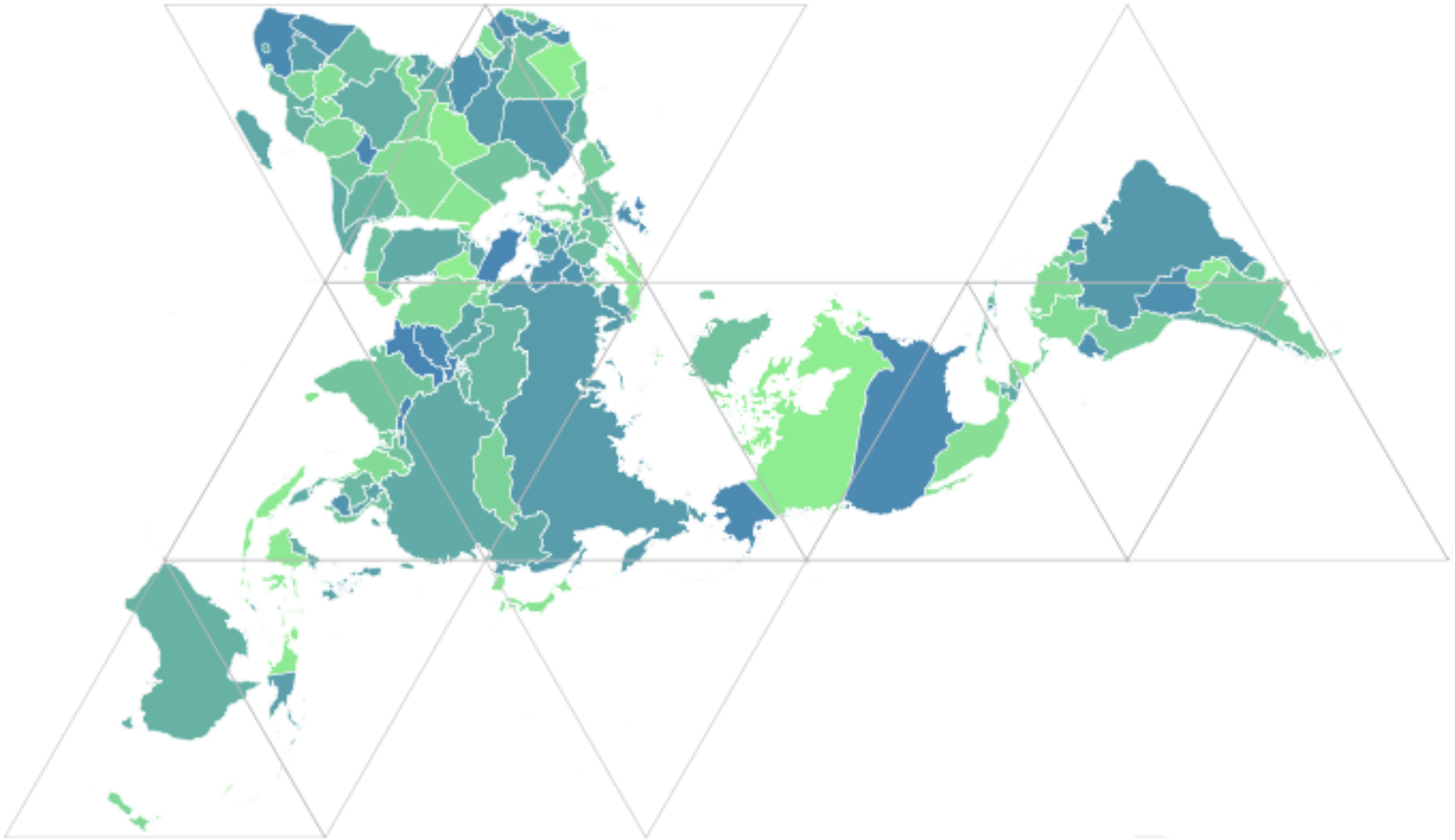
Bach's Prelude #1 in C Major | Jieun Oh



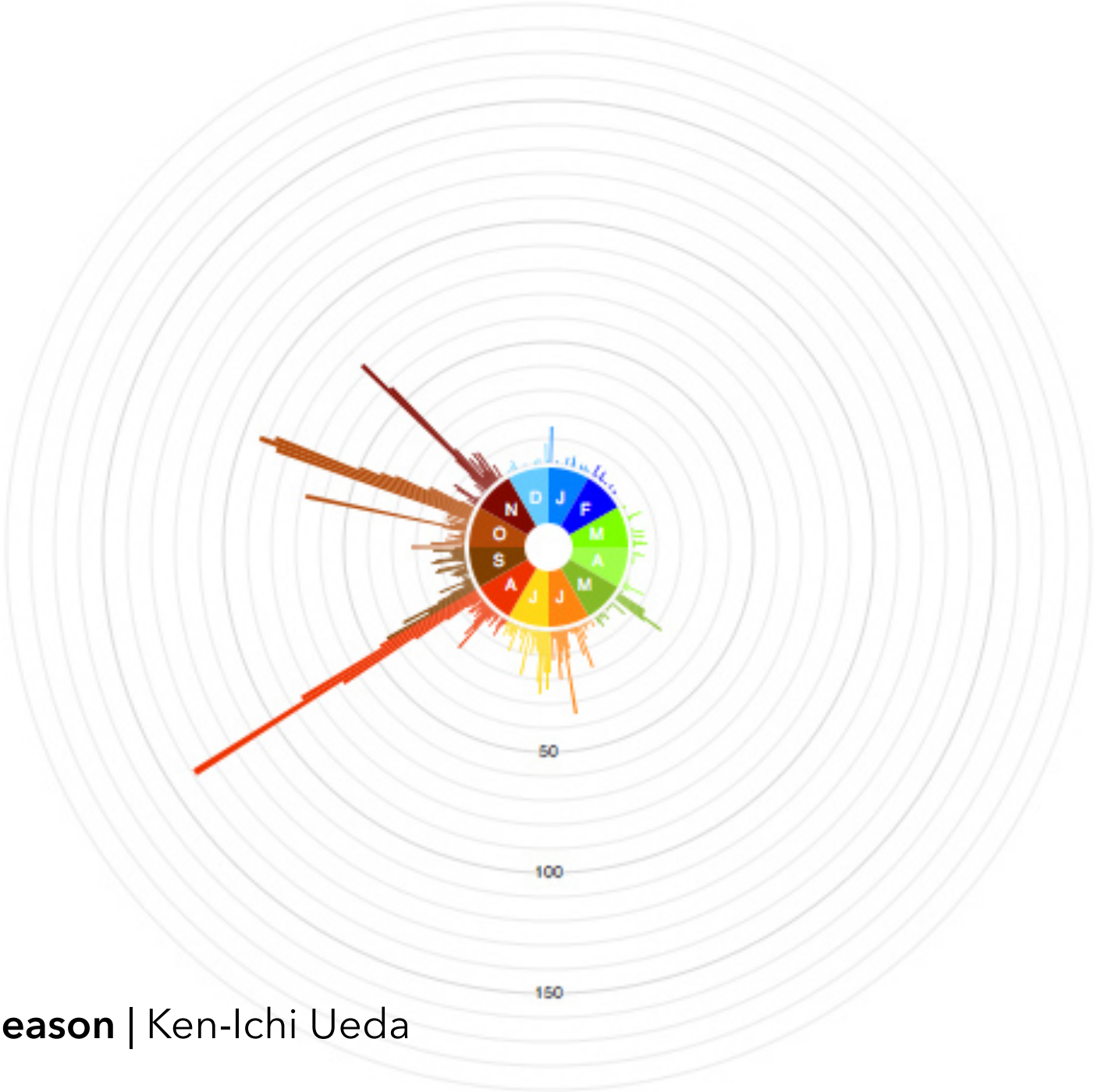
Obesity Map | Vadim Ogievetsky



Obesity Map | Vadim Ogievetsky

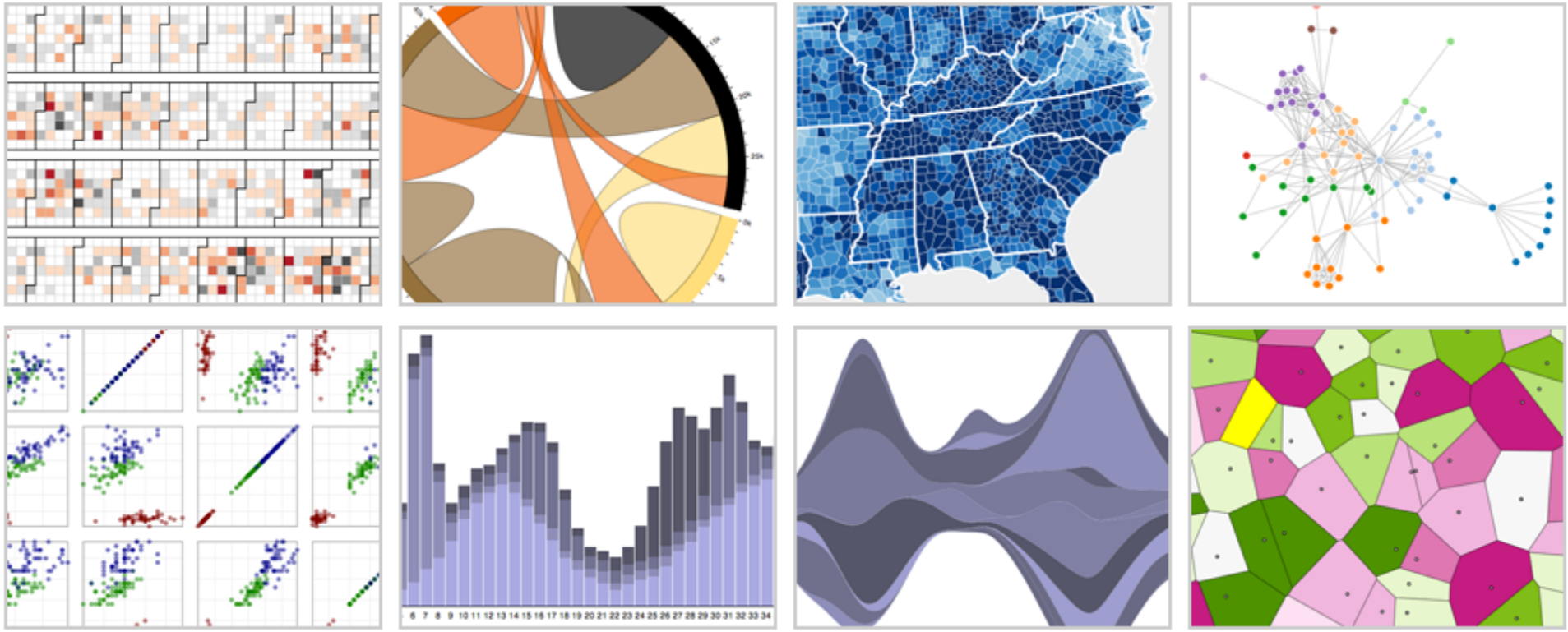


Dymaxion Maps | Vadim Ogievetsky



FlickrSeason | Ken-Ichi Ueda

d3.js Data-Driven Documents



with **Mike Bostock** & 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
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
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”);
```

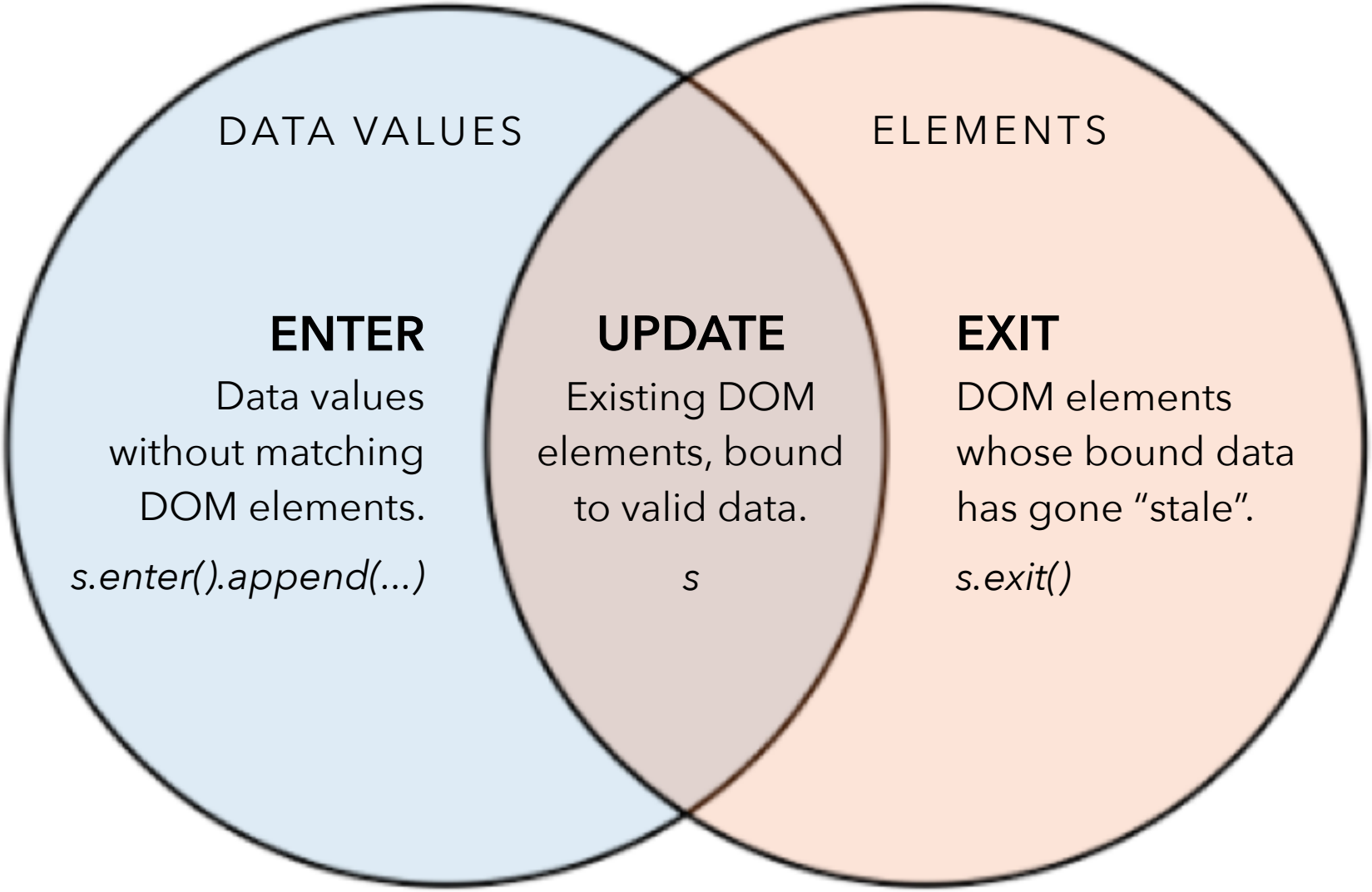
Data Binding

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

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



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

Processing, OpenGL, Java2D

Administrivia

A2: Exploratory Data Analysis

Use visualization software to form & answer questions

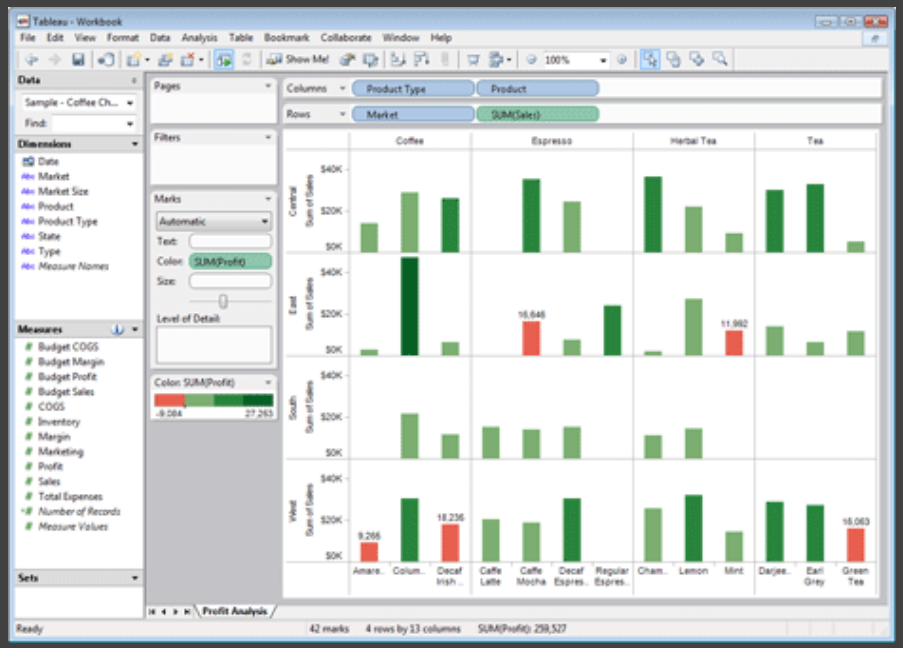
~~**First steps:** (Due Mon 4/10)~~

Step 1: Pick domain & data

Step 2: Pose questions

Step 3: Profile the data

Iterate as needed



Create visualizations

Interact with data

Refine your questions

Author a report

Screenshots of most insightful views *(10+)*

Include titles and captions for each view

Due by 5:00pm

Friday, April 14

Tutorials!

~~**Web Programming: JavaScript, SVG, CSS**~~
~~Thursday, April 6 - 4:30-5:50pm - PAA A118~~

Introduction to D3.js
Thursday, April 13 - 4:30-5:50pm - PAA A118

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

Processing, OpenGL, Java2D

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

Visualization Grammars

Protovis, D3.js

Declarative Languages

Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Processing, OpenGL, Java2D

Programming Toolkits

What is a Declarative Language?

What is a Declarative Language?

Programming by describing *what*, not *how*

What is a Declarative Language?

Programming by describing *what*, not *how*

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

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)
 .enter().append("rect")
  .attr("x", function(d) { return xscale(d.foo); })
  .attr("y", function(d) { return yscale(d.bar); })
```

TODAY'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 Hon

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.

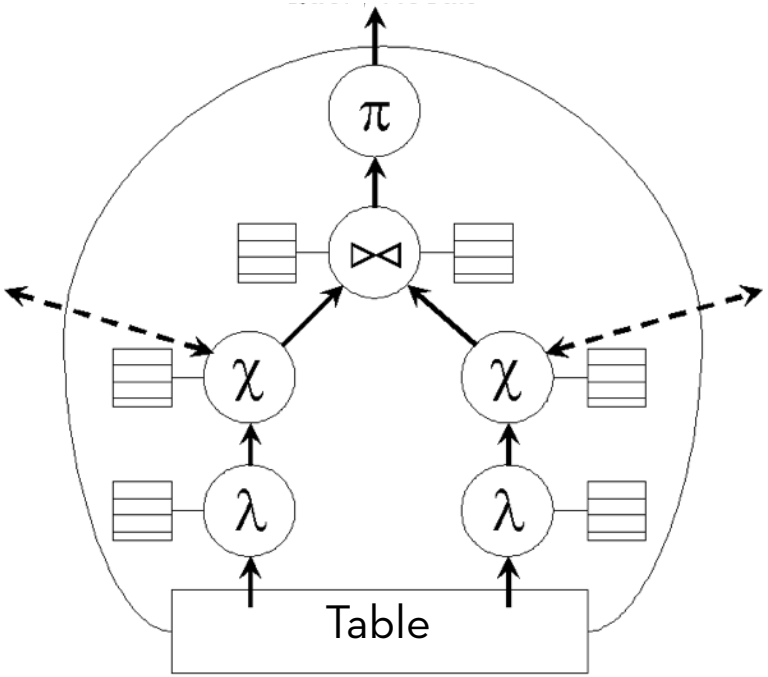
- Brooks: No Second Thoughts | Comments (200)
- Herbert: The Corrosion of America
- Cohen: Turkey Steps Out
- Editorial: Mortgage Mess
- Bloggingheads: Jon Stewart's Power

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/js/test/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_nam
HAVING COUNT(order_id) > 5
ORDER BY COUNT(order_id) DESC
```

SQL

Why Declarative Languages?

Why Declarative Languages?

Faster iteration. Less code. Larger user base.

Why Declarative Languages?

Faster iteration. Less code. Larger user base.

Better visualization. *Smart defaults.*

Why Declarative Languages?

Faster iteration. Less code. Larger user base.

Better visualization. *Smart defaults.*

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

Why Declarative Languages?

Faster iteration. Less code. Larger user base.

Better visualization. *Smart defaults.*

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

Performance. *Optimization, scalability.*

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.*

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, Polestar, Voyager***

Graphical Interfaces

Visual Analysis Grammars

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

Visualization Grammars

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

Declarative Languages

Component Architectures

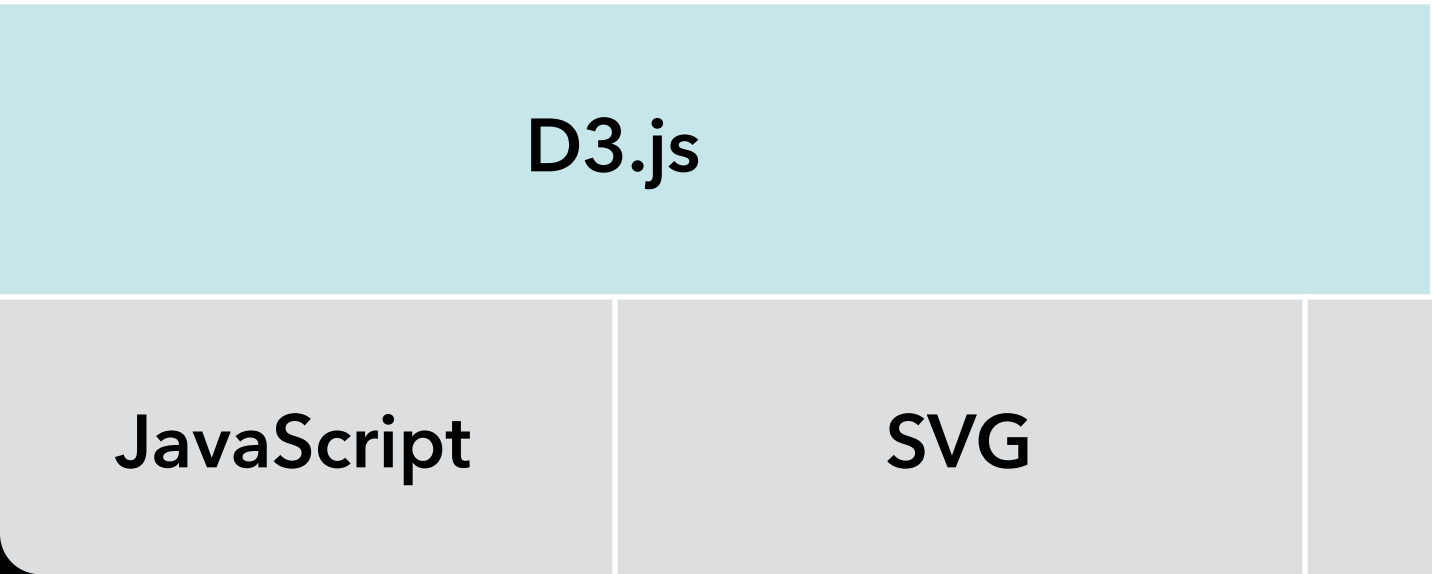
Prefuse, Flare, Improvise, VTK

Graphics APIs

Processing, OpenGL, Java2D

Programming Toolkits

JavaScript	SVG	Canvas


D3.js
JavaScript
SVG
Canvas


Vega
D3.js
JavaScript
SVG
Canvas

Visualization Grammar

Visualization Grammar

Data Input data to visualize

Visualization Grammar

Data	Input data to visualize
Transforms	Grouping, stats, projection, layout

Visualization Grammar

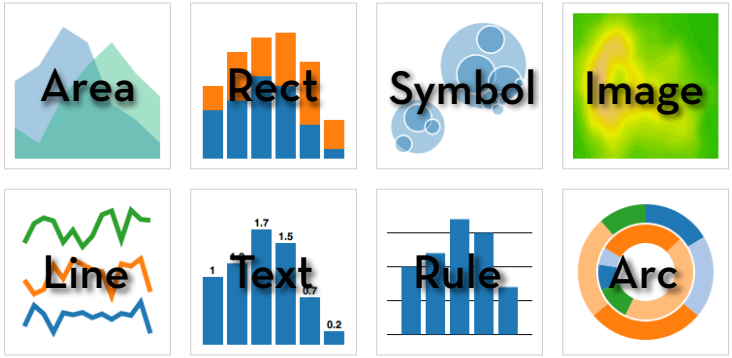
Data	Input data to visualize
Transforms	Grouping, stats, projection, layout
Scales	Map data values to visual values

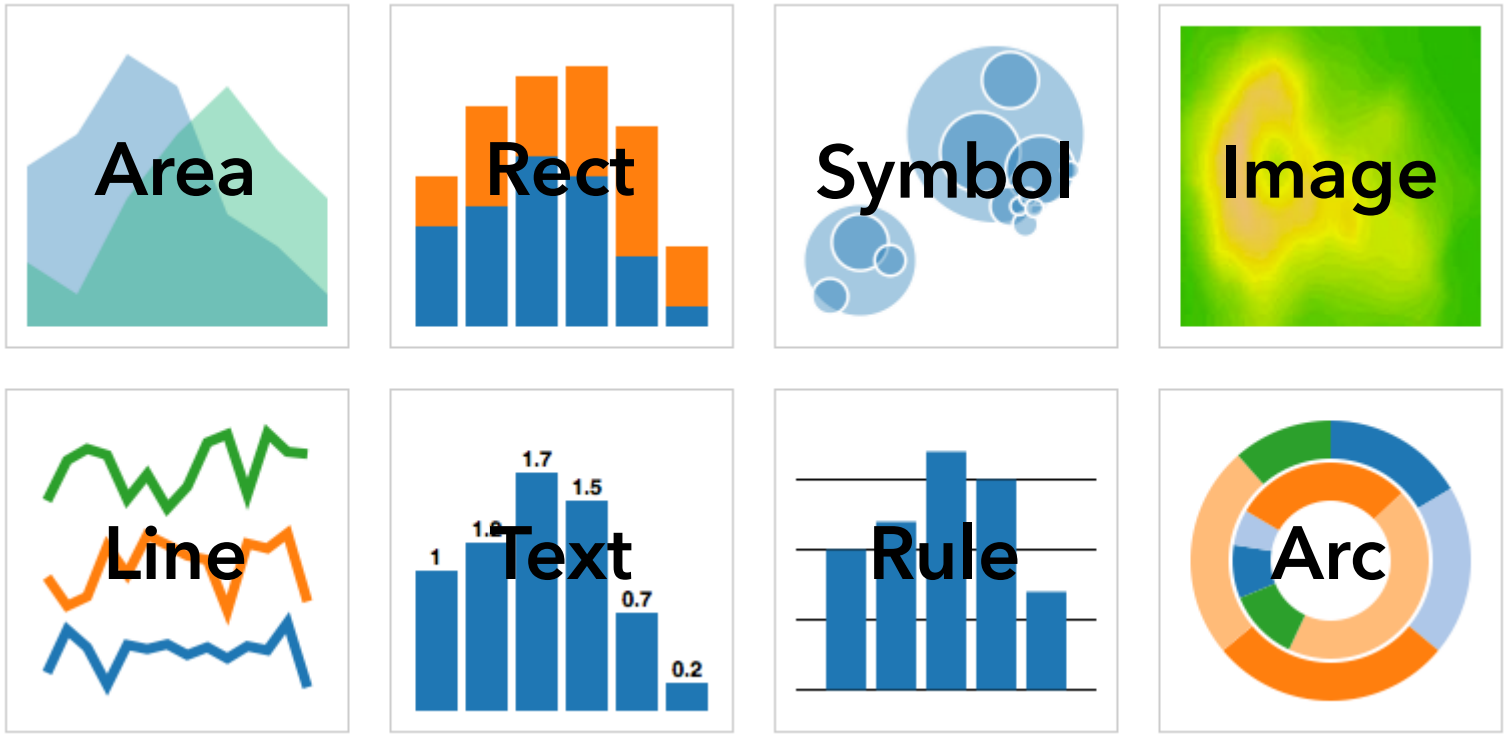
Visualization Grammar

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

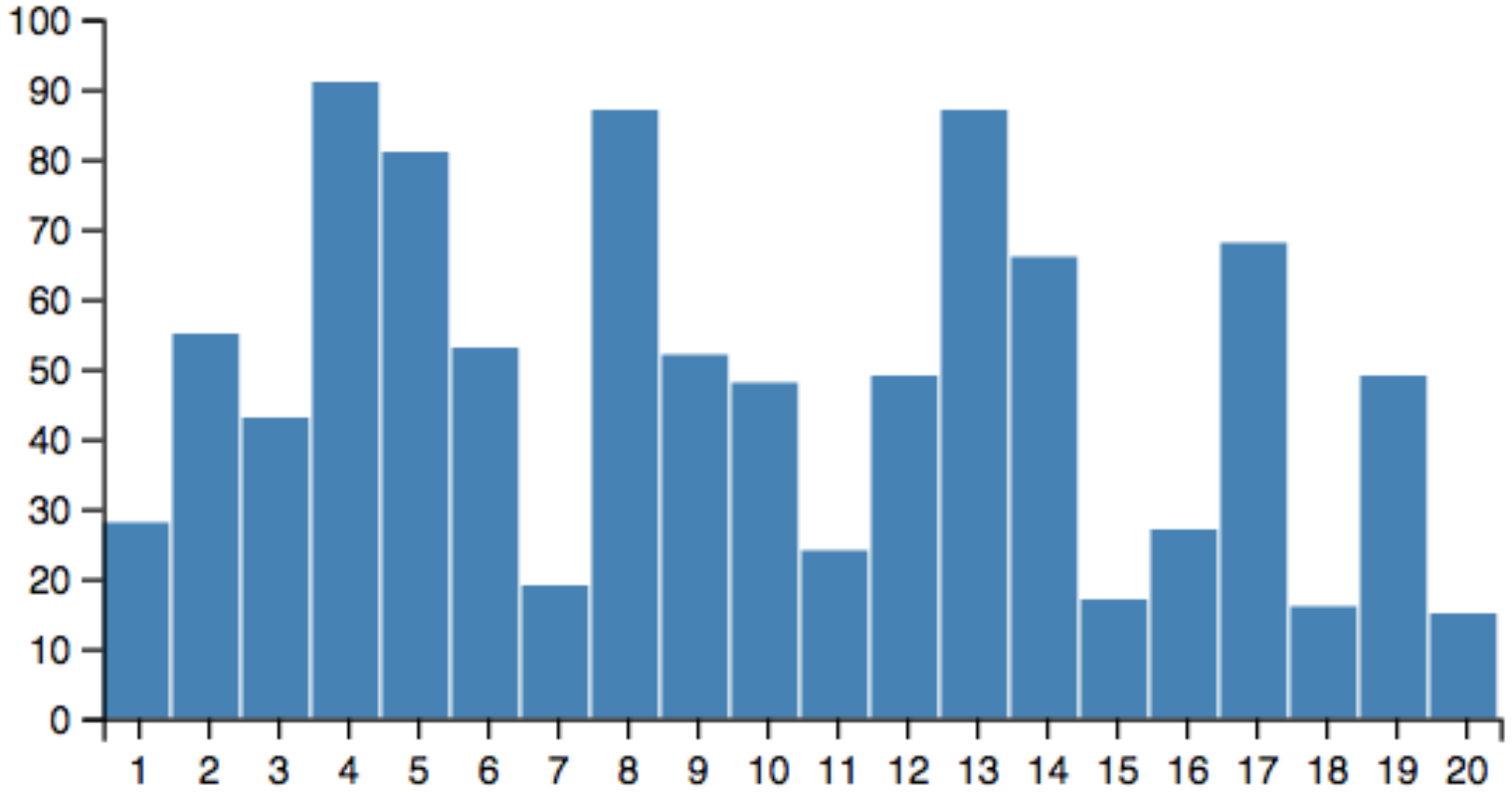
Visualization Grammar

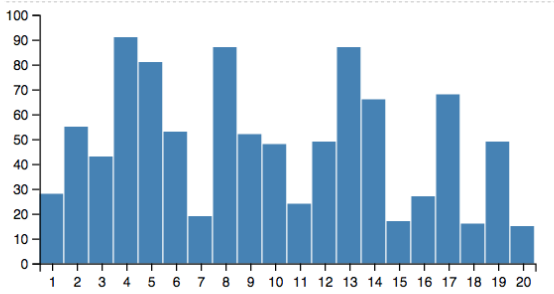
Data	Input data to visualize
Transforms	Grouping, stats, projection, layout
Scales	Map data values to visual values
Guides	Axes & legends visualize scales
Marks	Data-representative graphics



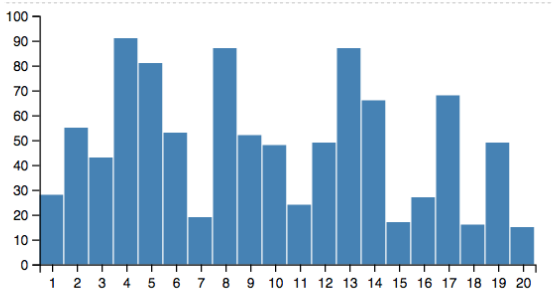


MARKS: Graphical Primitives


100
90
80
70
60
50
40
30
20
10
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20

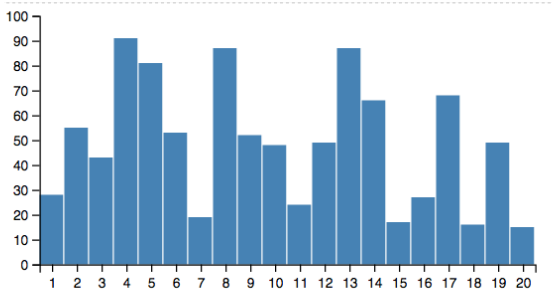

100
90
80
70
60
50
40
30
20
10
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

```
{
  "width": 400, "height": 200,
  "data": [
    {"name": "table", "url": "/data/sample.json"}
  ],
  "scales": [
    {
      "name": "x", "type": "band",
      "range": "width",
      "domain": {"data": "table", "field": "x"}
    },
    {
      "name": "y", "type": "linear",
      "range": "height", "nice": true,
      "domain": {"data": "table", "field": "y"}
    }
  ],
  "axes": [
    {"orient": "left", "scale": "x"},
    {"orient": "bottom", "scale": "y"}
  ],
  "marks": [{
    "type": "rect",
    "from": {"data": "table"},
    "encode": {
      "enter": {
        "x": {"scale": "x", "field": "x"},
        "width": {"scale": "x", "band": 1, "offset": -1},
        "y": {"scale": "y", "field": "y"},
        "y2": {"scale": "y", "value": 0},
        "fill": {"value": "steelblue"}
      }
    }
  }]
}
```



Data + Transforms

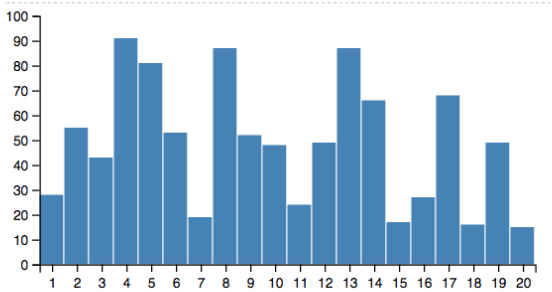
```
{
  "width": 400, "height": 200,
  "data": [
    {"name": "table", "url": "/data/sample.json"}
  ],
  "scales": [
    {
      "name": "x", "type": "band",
      "range": "width",
      "domain": {"data": "table", "field": "x"}
    },
    {
      "name": "y", "type": "linear",
      "range": "height", "nice": true,
      "domain": {"data": "table", "field": "y"}
    }
  ],
  "axes": [
    {"orient": "left", "scale": "x"},
    {"orient": "bottom", "scale": "y"}
  ],
  "marks": [{
    "type": "rect",
    "from": {"data": "table"},
    "encode": {
      "enter": {
        "x": {"scale": "x", "field": "x"},
        "width": {"scale": "x", "band": 1, "offset": -1},
        "y": {"scale": "y", "field": "y"},
        "y2": {"scale": "y", "value": 0},
        "fill": {"value": "steelblue"}
      }
    }
  }]
}
```



```
{
  "width": 400, "height": 200,
  "data": [
    {"name": "table", "url": "/data/sample.json"}
  ],
  "scales": [
    {
      "name": "x", "type": "band",
      "range": "width",
      "domain": {"data": "table", "field": "x"}
    },
    {
      "name": "y", "type": "linear",
      "range": "height", "nice": true,
      "domain": {"data": "table", "field": "y"}
    }
  ],
  "axes": [
    {"orient": "left", "scale": "x"},
    {"orient": "bottom", "scale": "y"}
  ],
  "marks": [{
    "type": "rect",
    "from": {"data": "table"},
    "encode": {
      "enter": {
        "x": {"scale": "x", "field": "x"},
        "width": {"scale": "x", "band": 1, "offset": -1},
        "y": {"scale": "y", "field": "y"},
        "y2": {"scale": "y", "value": 0},
        "fill": {"value": "steelblue"}
      }
    }
  }]
}
```

Data + Transforms

Scales

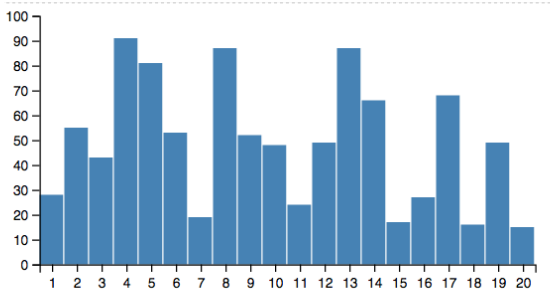


```
{
  "width": 400, "height": 200,
  "data": [
    {"name": "table", "url": "/data/sample.json"}
  ],
  "scales": [
    {
      "name": "x", "type": "band",
      "range": "width",
      "domain": {"data": "table", "field": "x"}
    },
    {
      "name": "y", "type": "linear",
      "range": "height", "nice": true,
      "domain": {"data": "table", "field": "y"}
    }
  ],
  "axes": [
    {"orient": "left", "scale": "x"},
    {"orient": "bottom", "scale": "y"}
  ],
  "marks": [{
    "type": "rect",
    "from": {"data": "table"},
    "encode": {
      "enter": {
        "x": {"scale": "x", "field": "x"},
        "width": {"scale": "x", "band": 1, "offset": -1},
        "y": {"scale": "y", "field": "y"},
        "y2": {"scale": "y", "value": 0},
        "fill": {"value": "steelblue"}
      }
    }
  }]
}
```

Data + Transforms

Scales

Guides



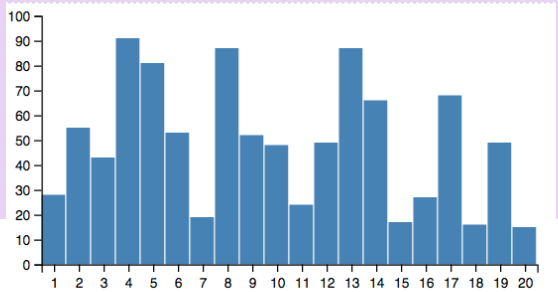
```
{
  "width": 400, "height": 200,
  "data": [
    {"name": "table", "url": "/data/sample.json"}
  ],
  "scales": [
    {
      "name": "x", "type": "band",
      "range": "width",
      "domain": {"data": "table", "field": "x"}
    },
    {
      "name": "y", "type": "linear",
      "range": "height", "nice": true,
      "domain": {"data": "table", "field": "y"}
    }
  ],
  "axes": [
    {"orient": "left", "scale": "x"},
    {"orient": "bottom", "scale": "y"}
  ],
  "marks": [{
    "type": "rect",
    "from": {"data": "table"},
    "encode": {
      "enter": {
        "x": {"scale": "x", "field": "x"},
        "width": {"scale": "x", "band": 1, "offset": -1},
        "y": {"scale": "y", "field": "y"},
        "y2": {"scale": "y", "value": 0},
        "fill": {"value": "steelblue"}
      }
    }
  }]
}
```

Data + Transforms

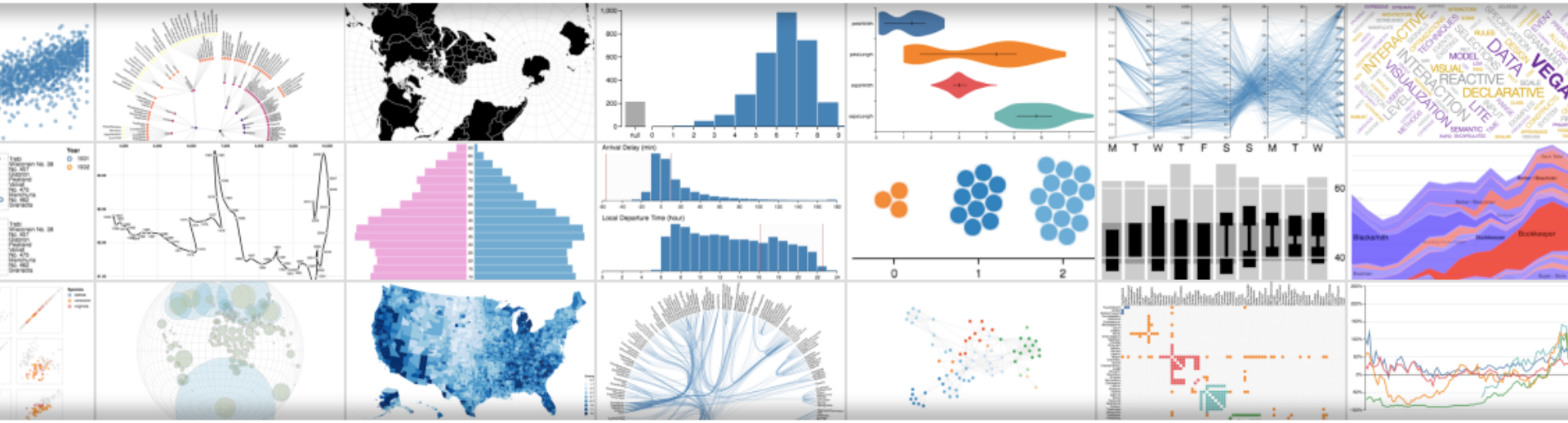
Scales

Guides

Marks



Vega – A Visualization Grammar



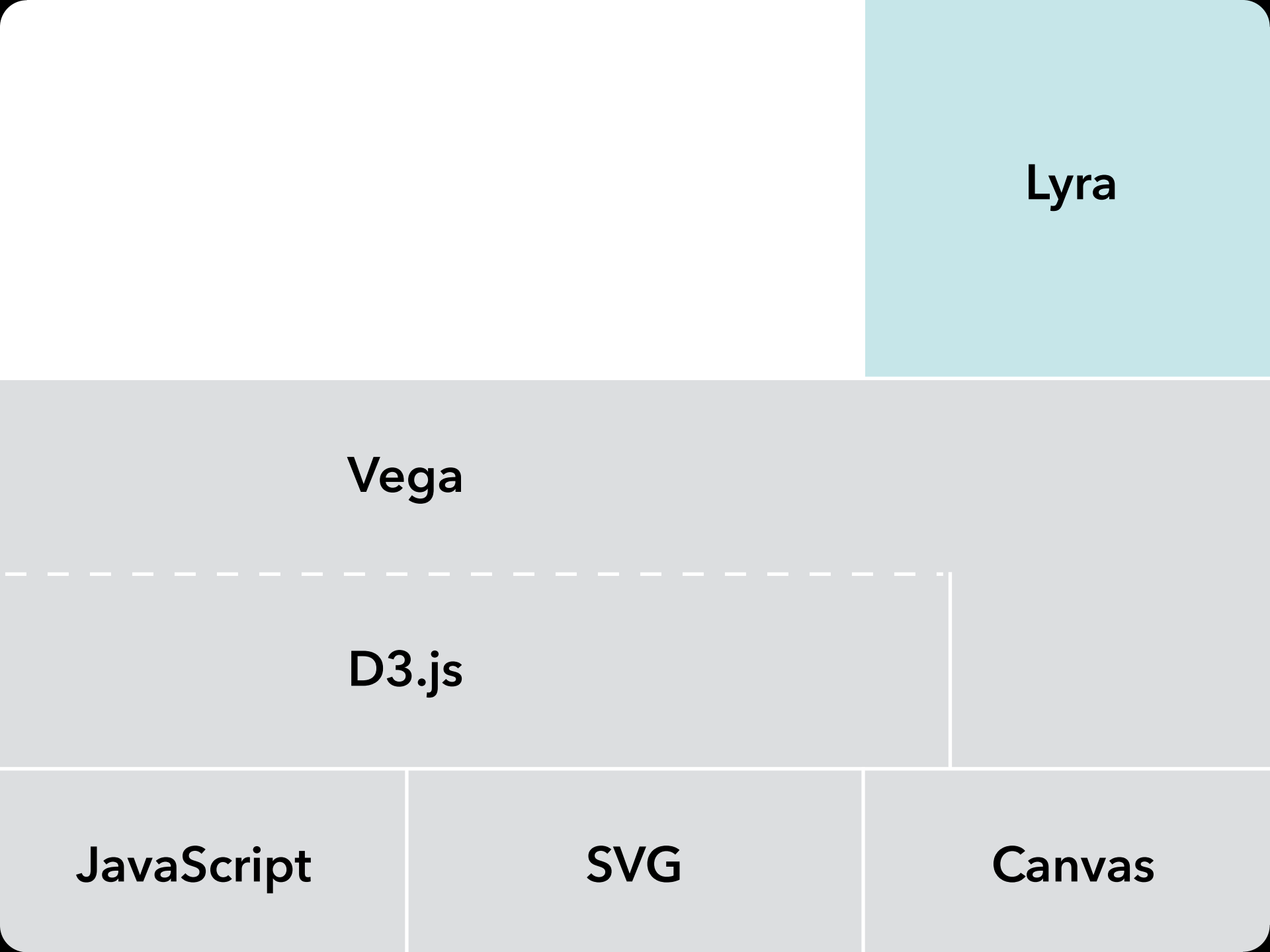
Vega is a *visualization grammar*, a declarative language for creating, saving, and sharing interactive visualization designs. With Vega, you can describe the visual appearance and interactive behavior of a visualization in a JSON format, and generate web-based views using Canvas or SVG.

Vega provides basic building blocks for a wide variety of visualization designs: data loading and transformation, scales, map projections, axes, legends, and graphical marks such as rectangles, lines, plotting symbols, *etc*. Interaction techniques can be specified using reactive signals that dynamically modify a visualization in response to input event streams.

Version 3.0.0-beta.28

A Vega *specification* defines an interactive visualization in a JSON format. Specifications are parsed by Vega's JavaScript *runtime* to generate both static images or interactive web-based views. Vega provides a convenient representation for computational generation of visualizations, and can serve as a foundation for new APIs and visual analysis tools.


Vega
D3.js
JavaScript
SVG
Canvas


Lyra
Vega
D3.js
JavaScript
SVG
Canvas

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.

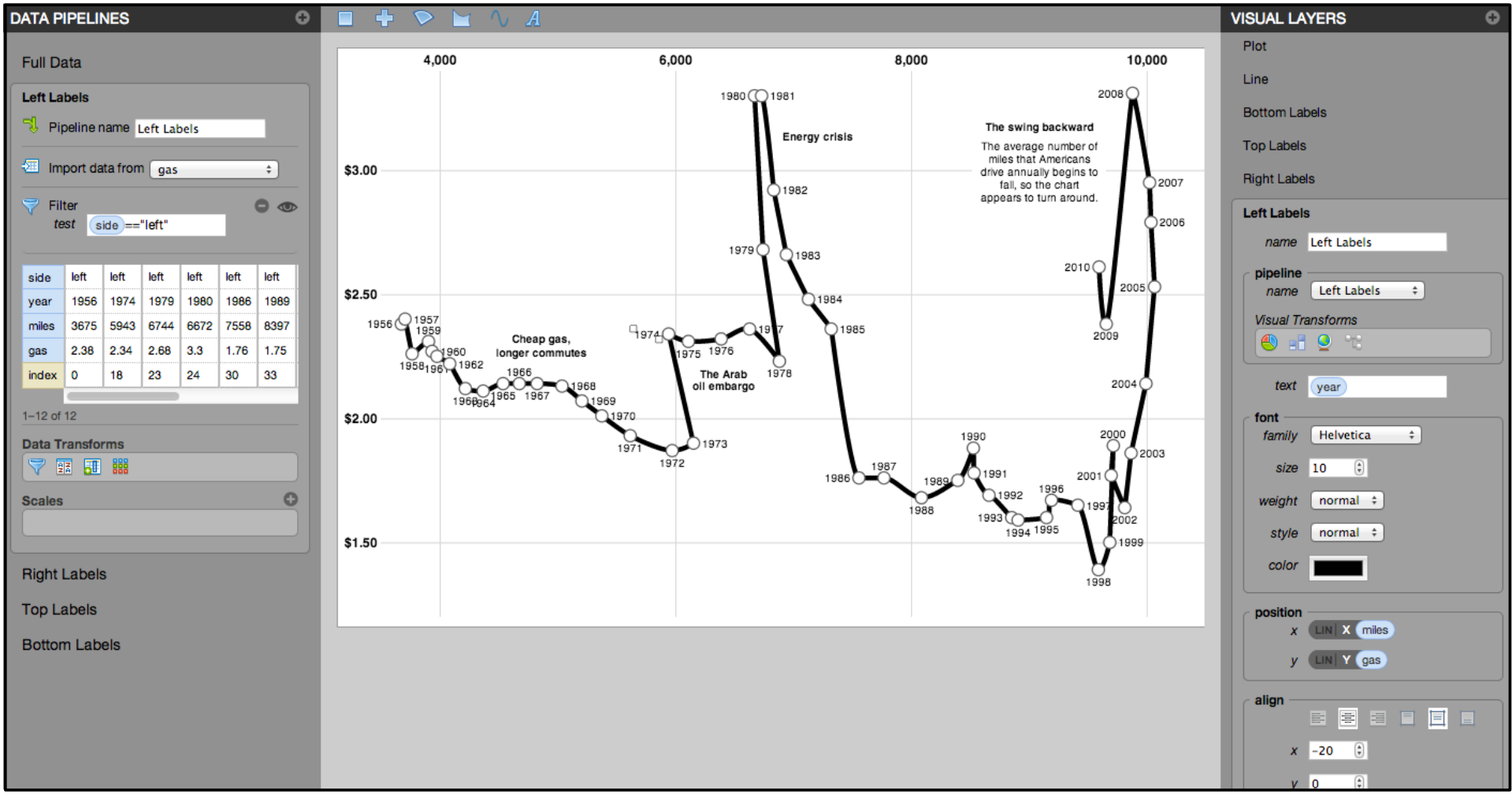
idl.cs.washington.edu/projects/lyra

ABSTRACT

Lyra is an interactive environment that enables custom visualization design without writing any code. Graphical "marks" can be bound to data fields using property drop zones; dynamically positioned using connectors; and directly moved, rotated, and resized using handles. Lyra also provides a data pipeline interface for iterative visual specification of data transformations and layout algorithms. Lyra is more expressive than interactive systems like Tableau, allowing designers to create custom visualizations comparable to hand-coded visualizations built with D3 or Processing. These visualizations can then be easily published and reused on the Web.

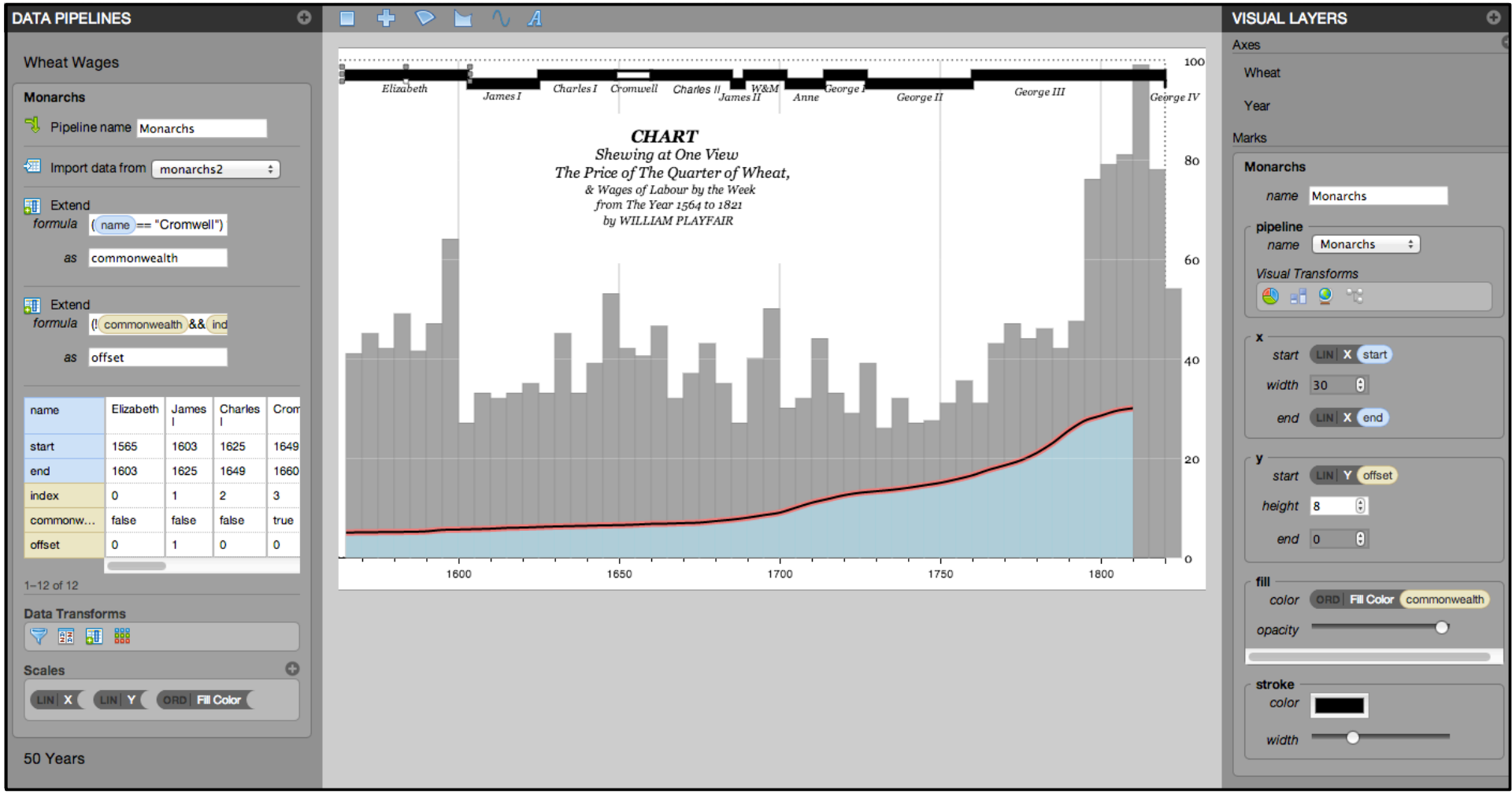
Lyra: An Interactive Visualization Design Environment

Lyra A Visualization Design Environment



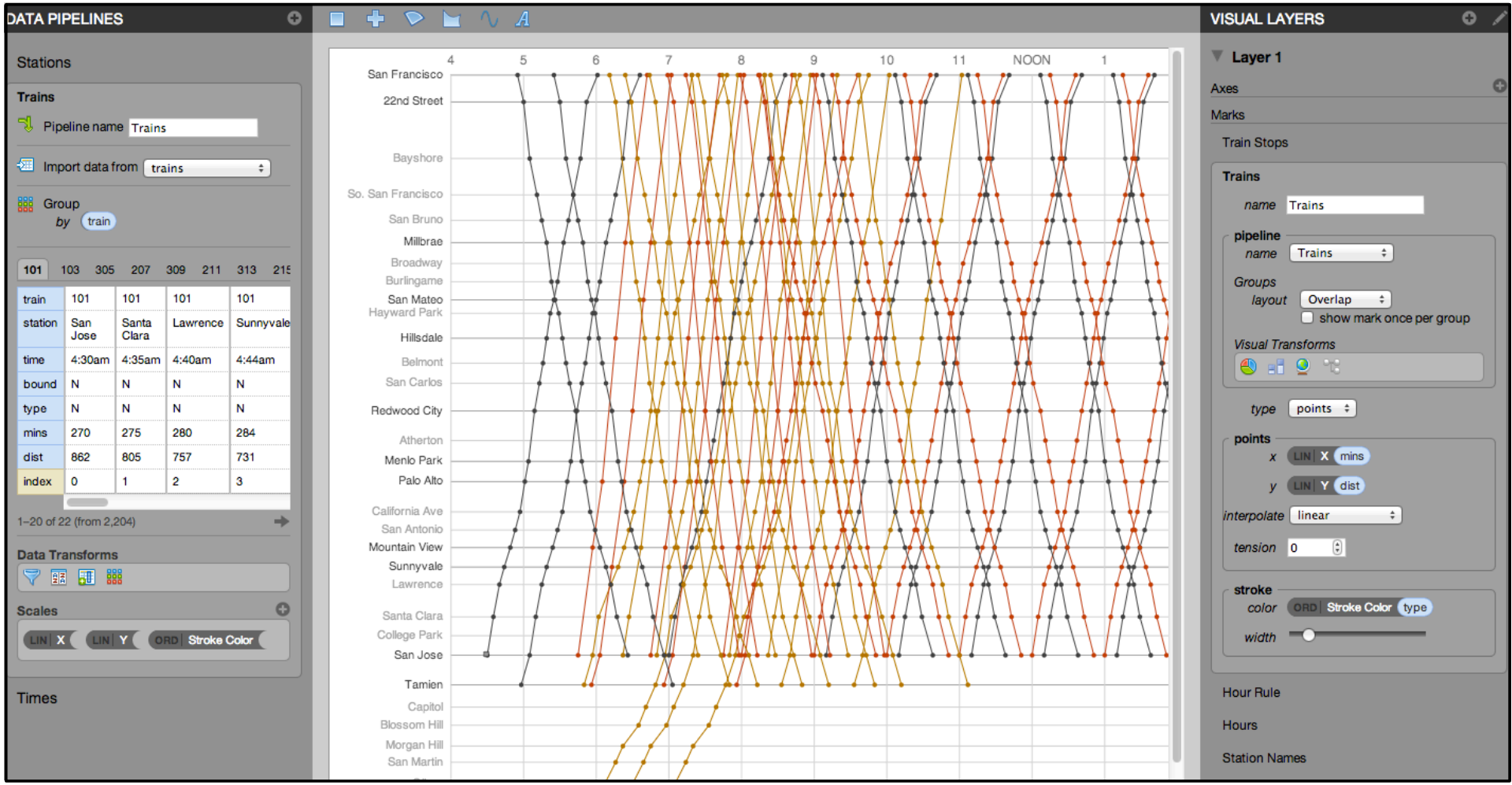
Driving Shifts into Reverse by Hannah Fairfield, NYTimes

Lyra A Visualization Design Environment



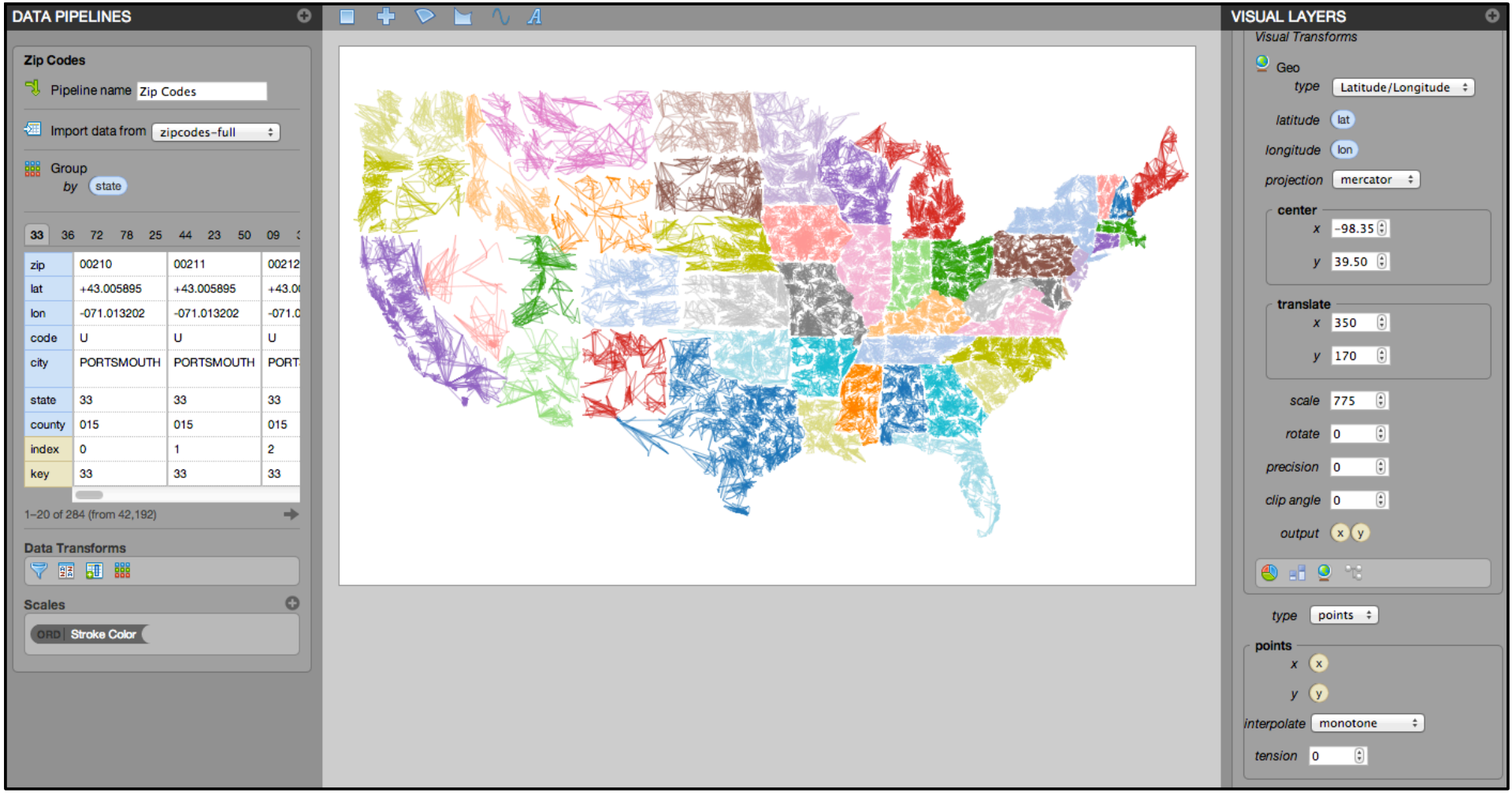
by William Playfair

Lyra A Visualization Design Environment



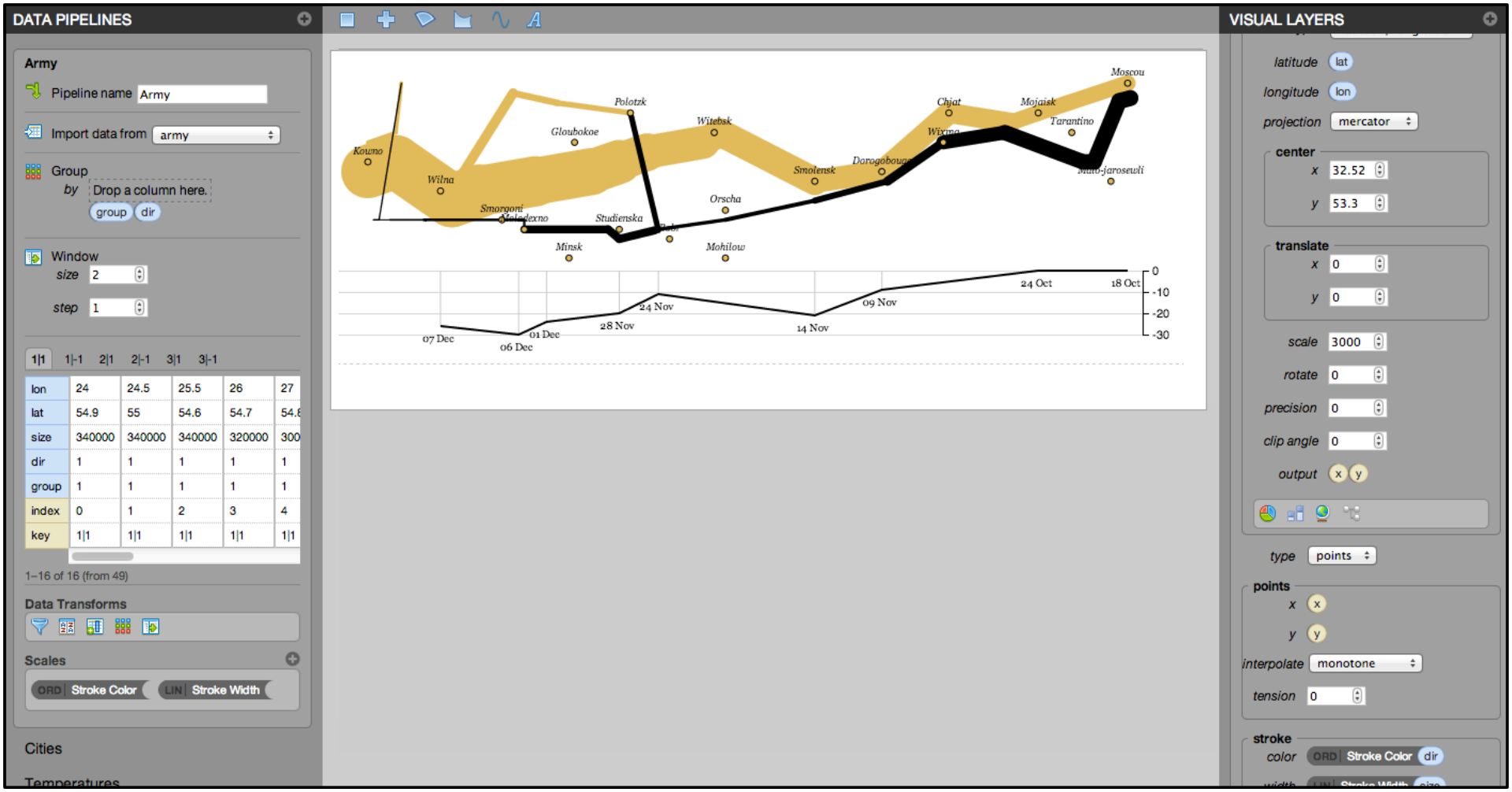
based on the **Railway Timetable** by E. J. Marey

Lyra A Visualization Design Environment

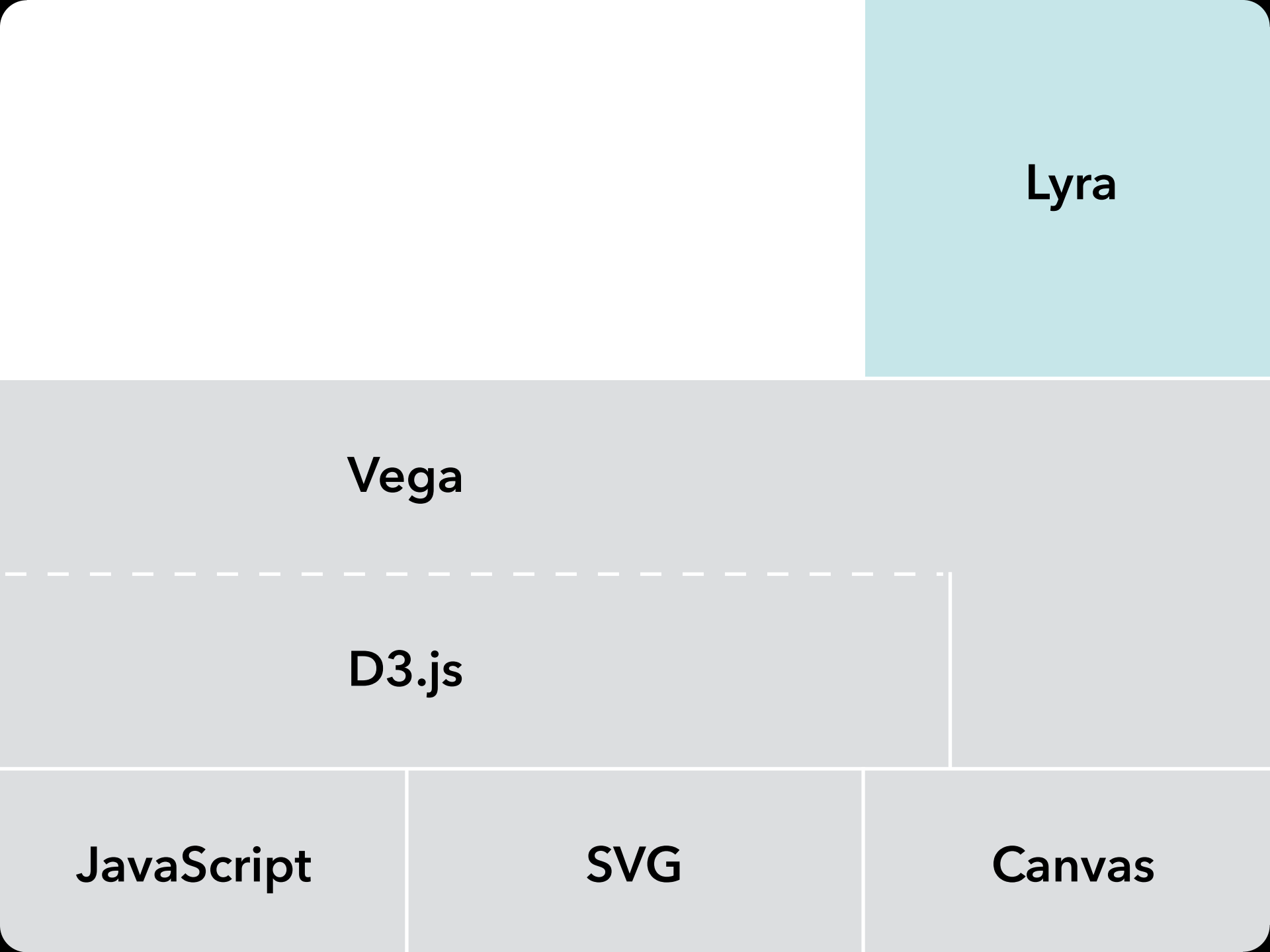


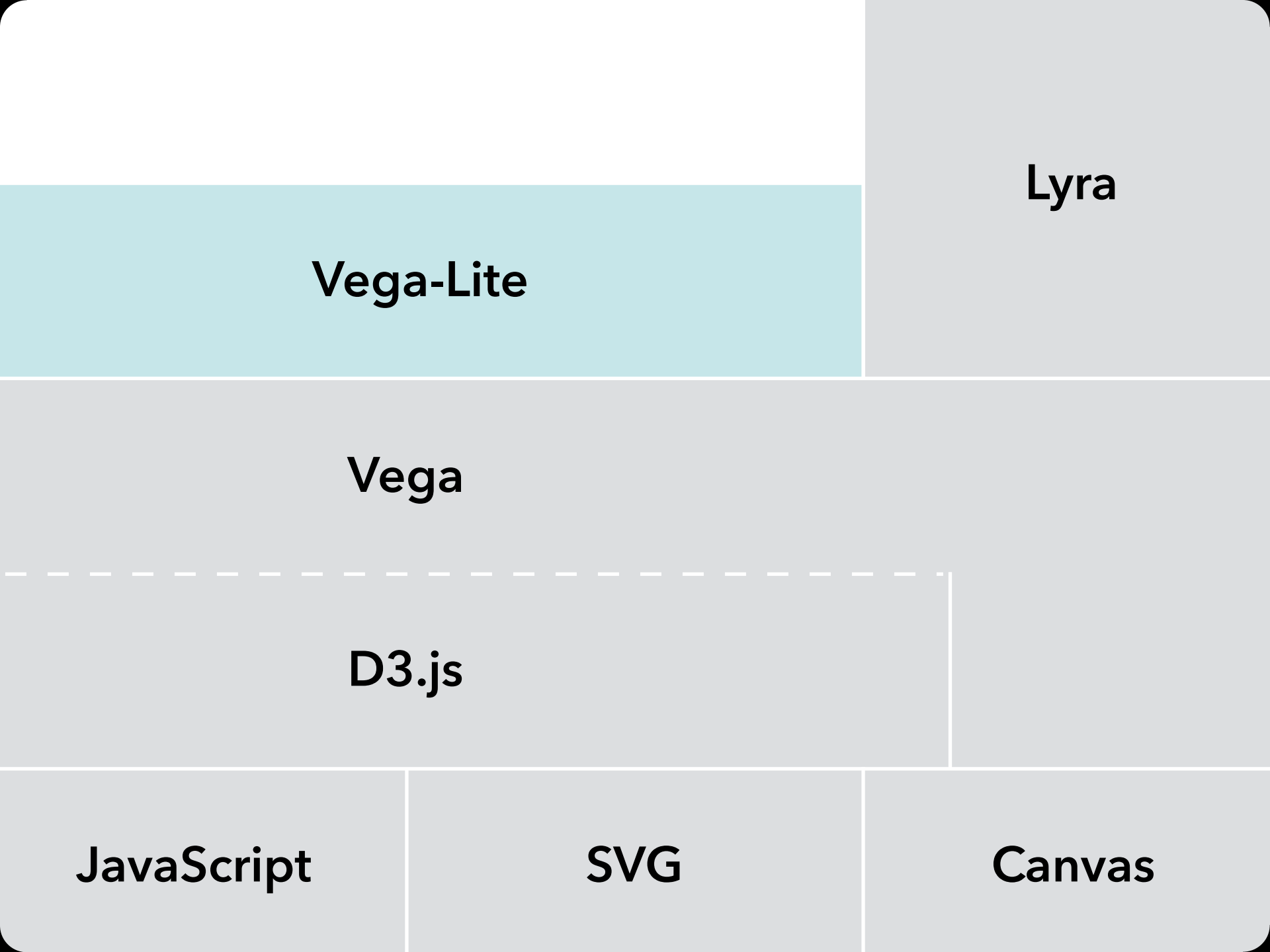
ZipScribble by Robert Kosara

Lyra A Visualization Design Environment



Napoleon's March by Charles Minard


Lyra
Vega
D3.js
JavaScript
SVG
Canvas


Lyra
Vega-Lite
Vega
D3.js
JavaScript
SVG
Canvas

Vega-Lite

A formal model for statistical graphics
Inspired by *Grammar of Graphics* & *Tableau*

Includes **data transformation** & **encoding**

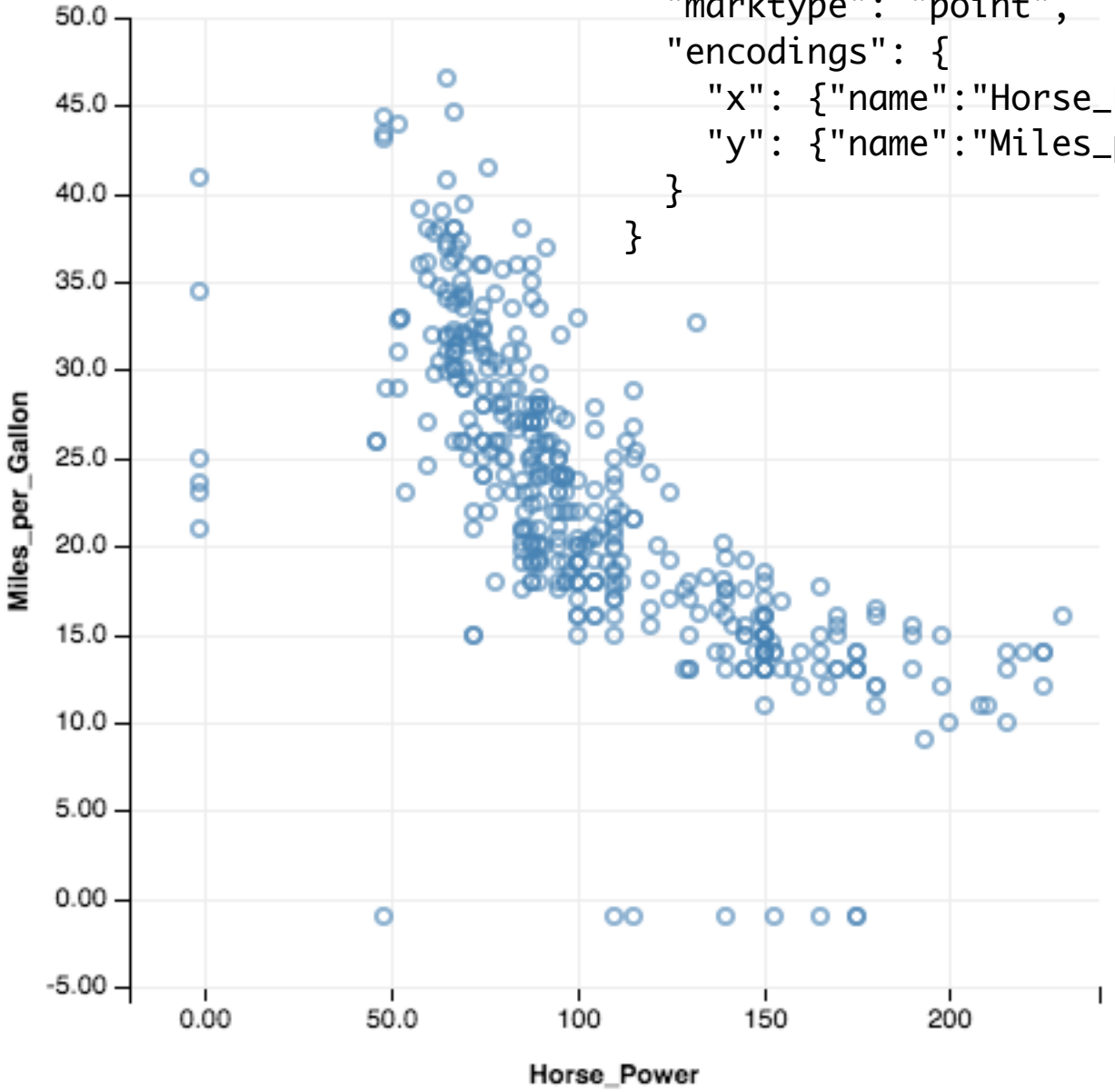
Vega-Lite

A formal model for statistical graphics
Inspired by *Grammar of Graphics* & *Tableau*

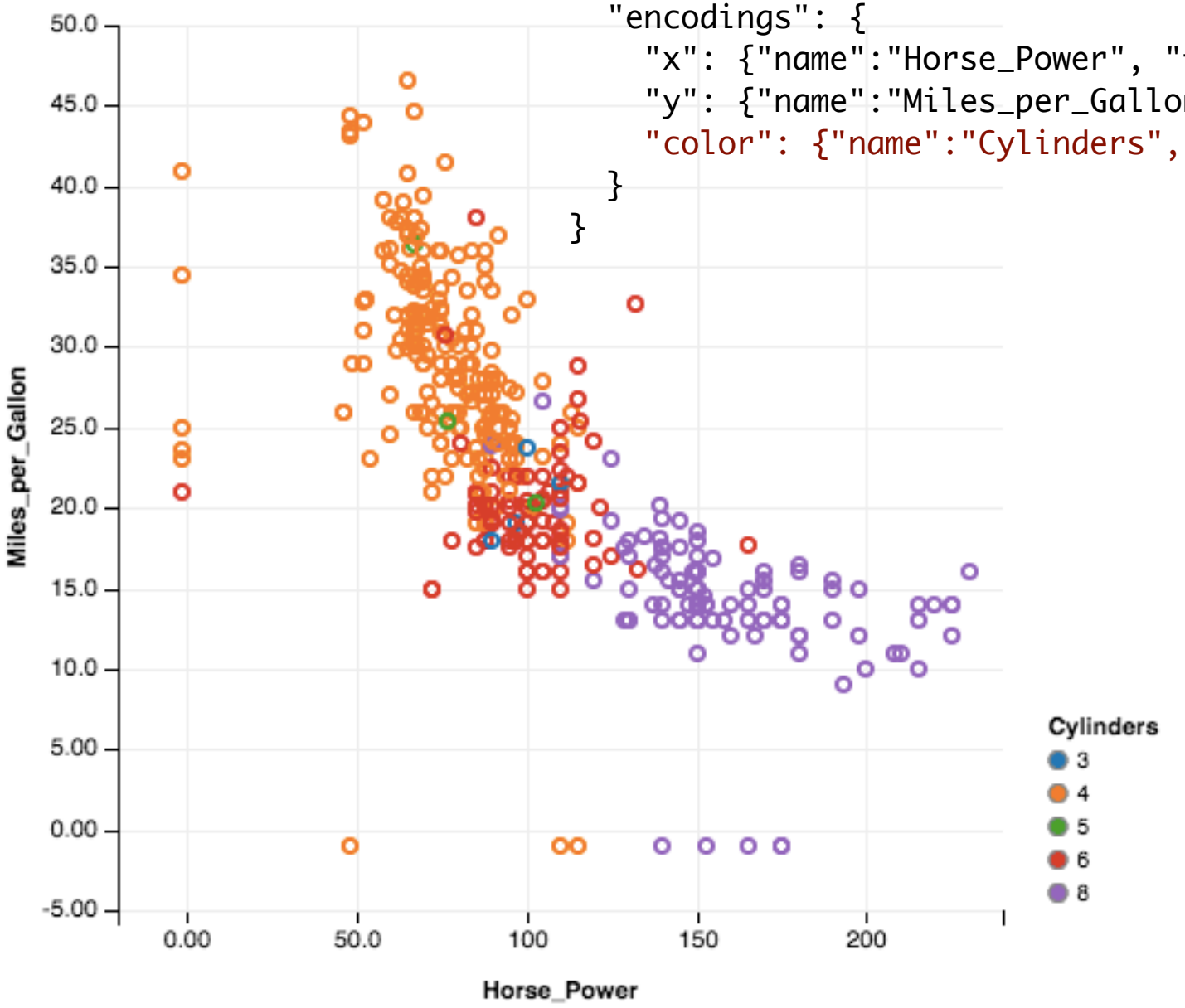
Includes **data transformation** & **encoding**

Uses a simple, concise **JSON format** that compiles to full-blown **Vega specifications**

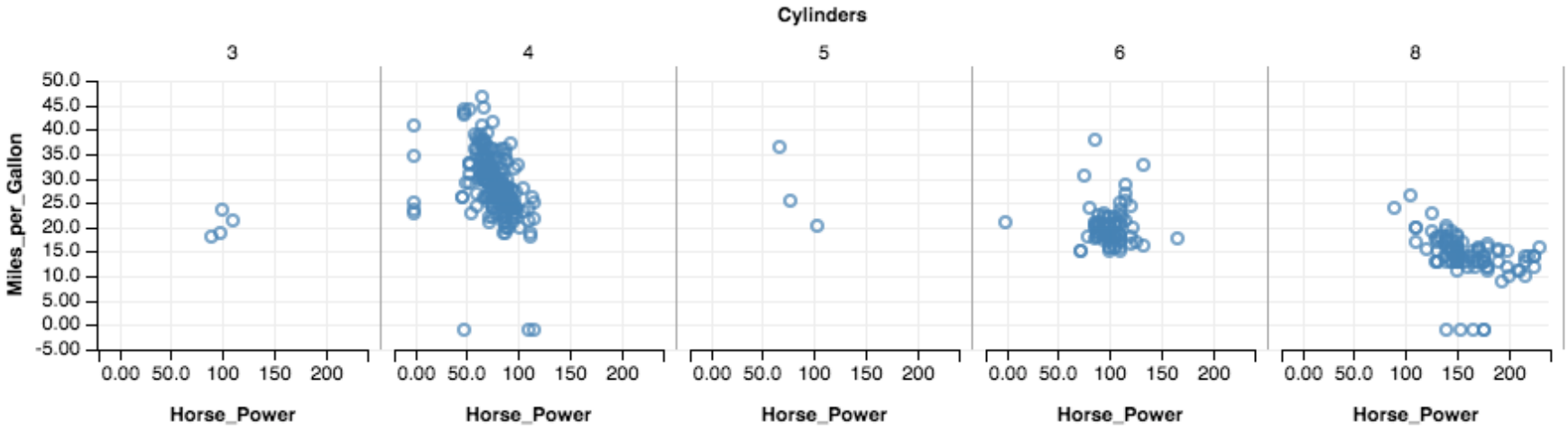
Easy **programmatic generation**



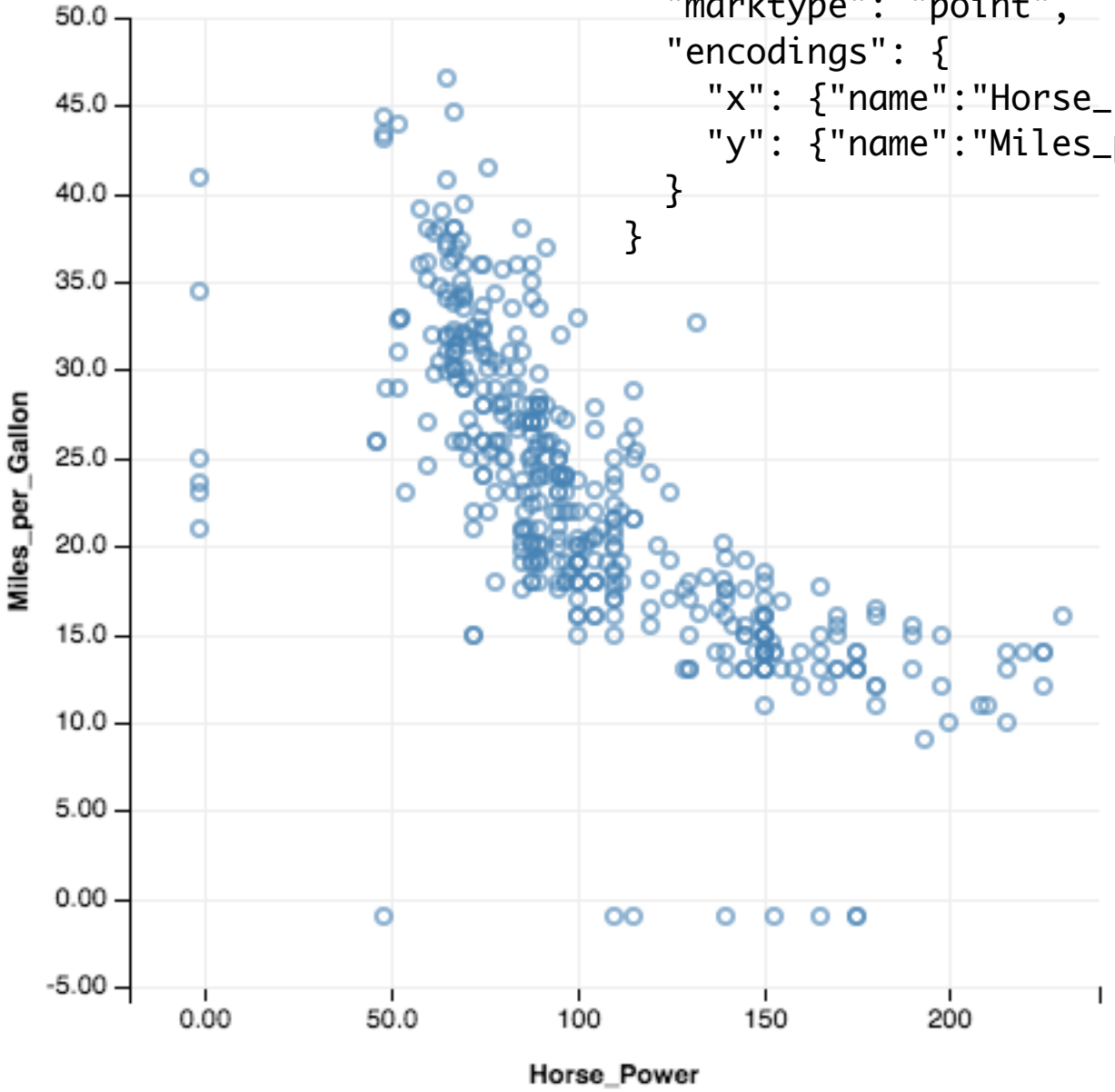

```
{
  "marktype": "point",
  "encodings": {
    "x": {"name":"Horse_Power", "type":"Q"},
    "y": {"name":"Miles_per_Gallon", "type":"Q"}
  }
}
```



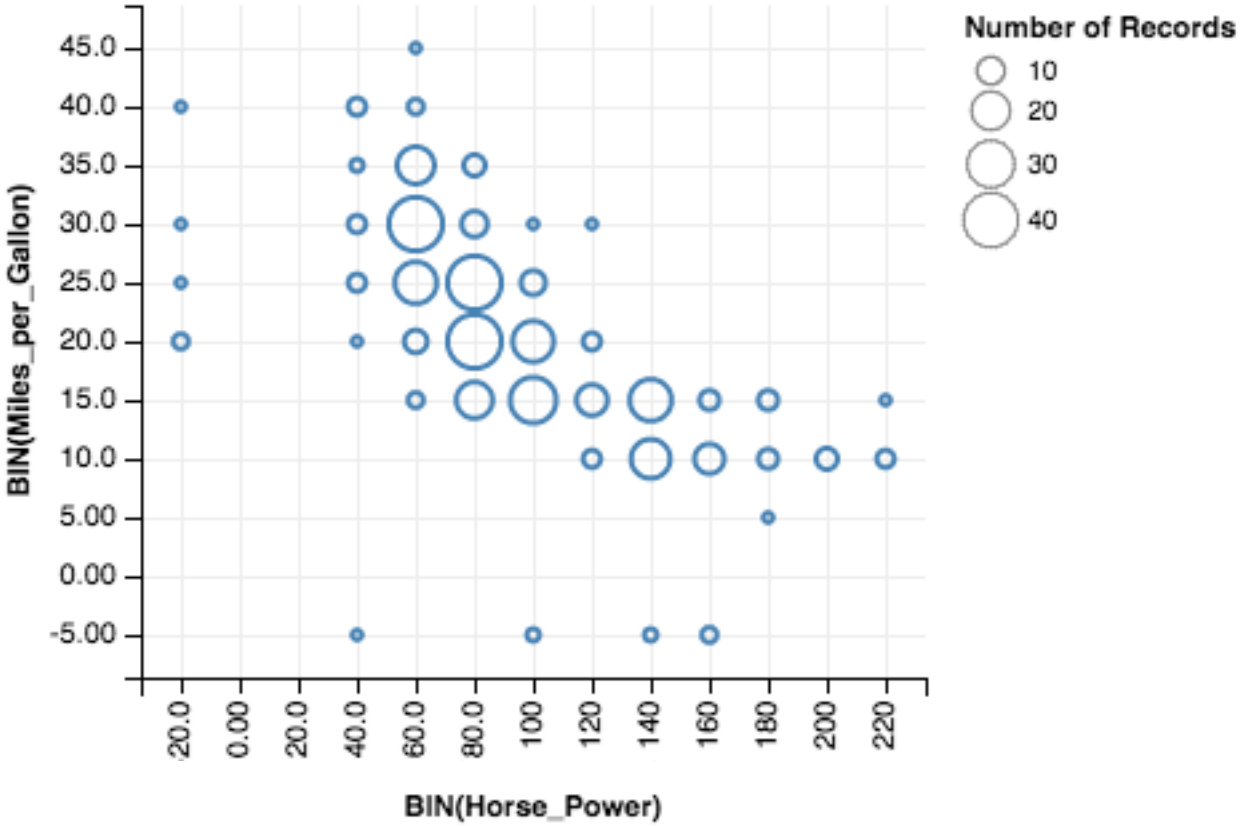

```
{
  "marktype": "point",
  "encodings": {
    "x": {"name":"Horse_Power", "type":"Q"},
    "y": {"name":"Miles_per_Gallon", "type":"Q"},
    "color": {"name":"Cylinders", "type":"O"}
  }
}
```



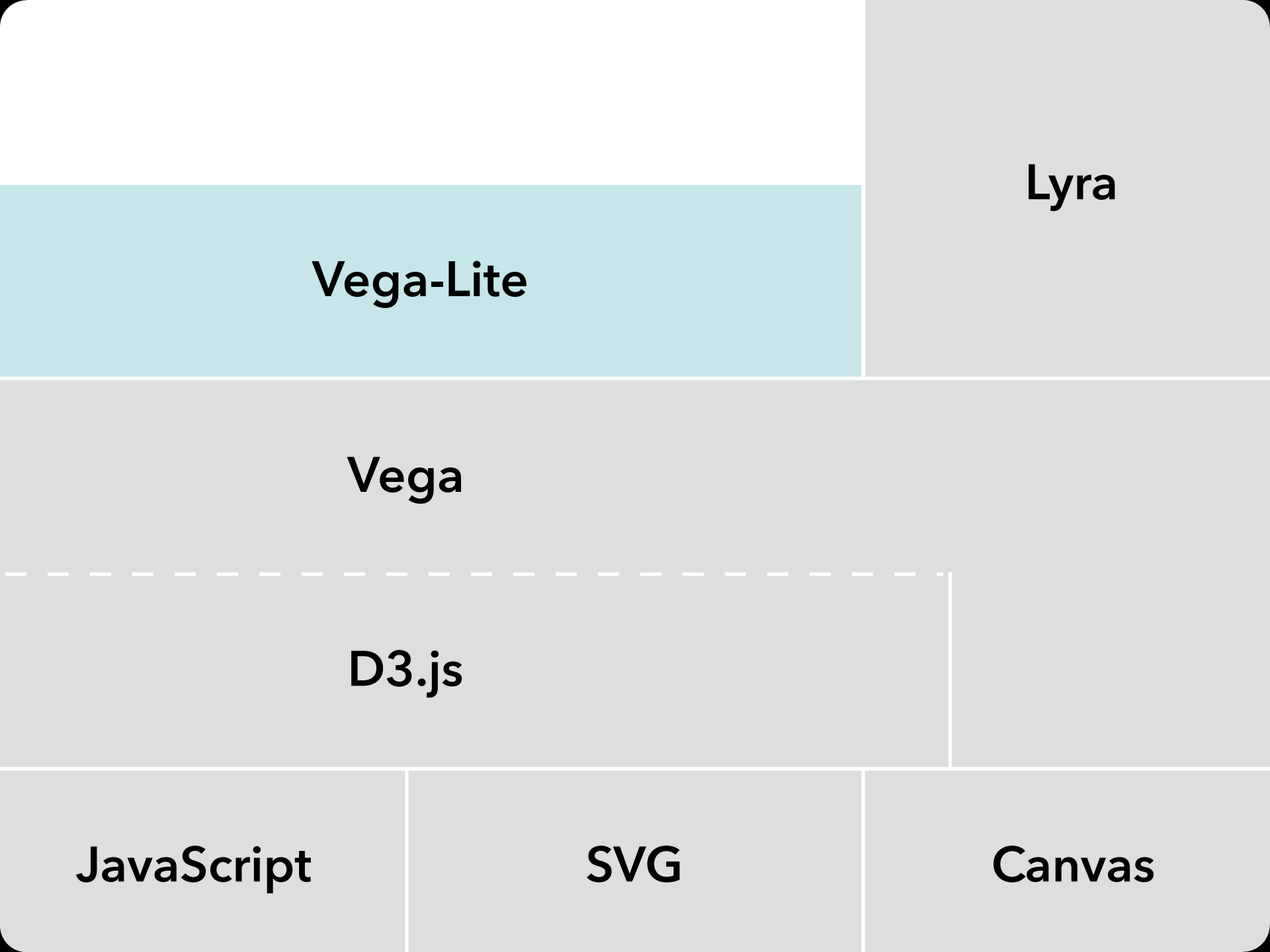

```
{
  "marktype": "point",
  "encodings": {
    "x": {"name":"Horse_Power", "type":"Q"},
    "y": {"name":"Miles_per_Gallon", "type":"Q"},
    "col": {"name":"Cylinders", "type":"O"}
  }
}
```

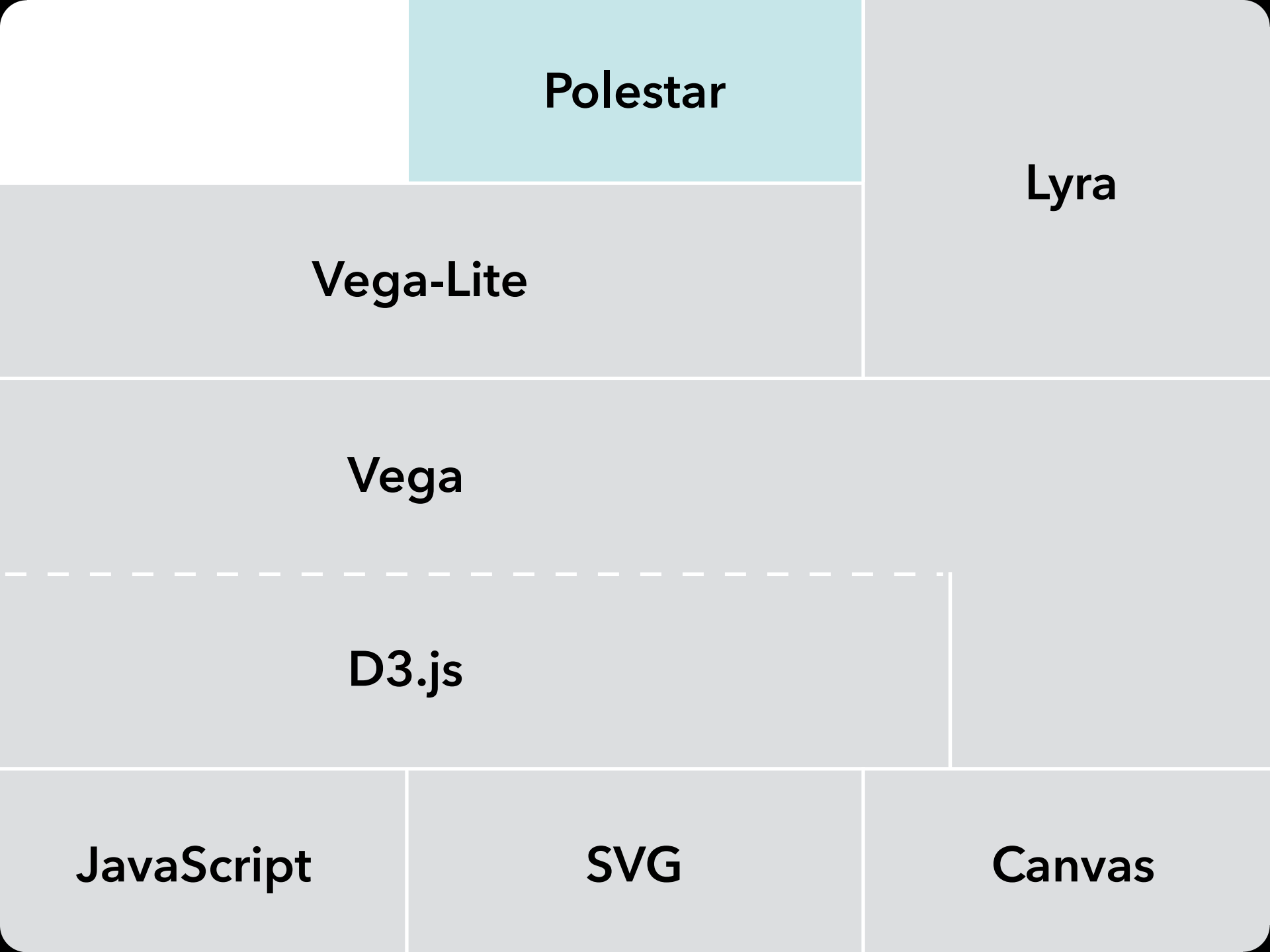



```
{
  "marktype": "point",
  "encodings": {
    "x": {"name":"Horse_Power", "type":"Q"},
    "y": {"name":"Miles_per_Gallon", "type":"Q"}
  }
}
```




```
{
  "marktype": "point",
  "encodings": {
    "x": {"name":"Horse_Power", "type":"Q", "bin":{"maxbins":15}},
    "y": {"name":"Miles_per_Gallon", "type":"Q", "bin":{"maxbins":15}},
    "size": {"name":"*", "type":"Q", "aggr":"count"}
  }
}
```


Lyra
Vega-Lite
Vega
D3.js
JavaScript
SVG
Canvas


Polestar
Lyra
Vega-Lite
Vega
D3.js
JavaScript
SVG
Canvas

Polestar

A graphical interface for **Vega-Lite**

Rapid visualization via drag-and-drop

Named in honor of **Polaris**, the research project that led to **Tableau**.

Pole★

https://uwdata.github.io/polestar

Pole★

Bookmarks (0) Undo Redo

DATA

Cars

- Abc Cylinders
- Abc name
- Abc Origin
- Year
- # Acceleration
- # Displacement
- # Horse Power
- # Miles per Gallon
- # Weight in lbs
- # COUNT

ENCODING

Clear

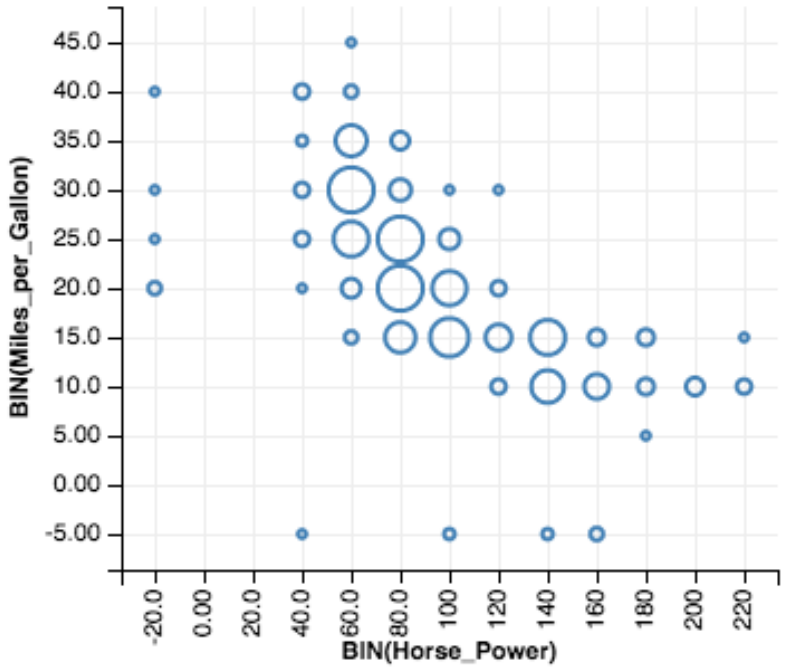
Positional

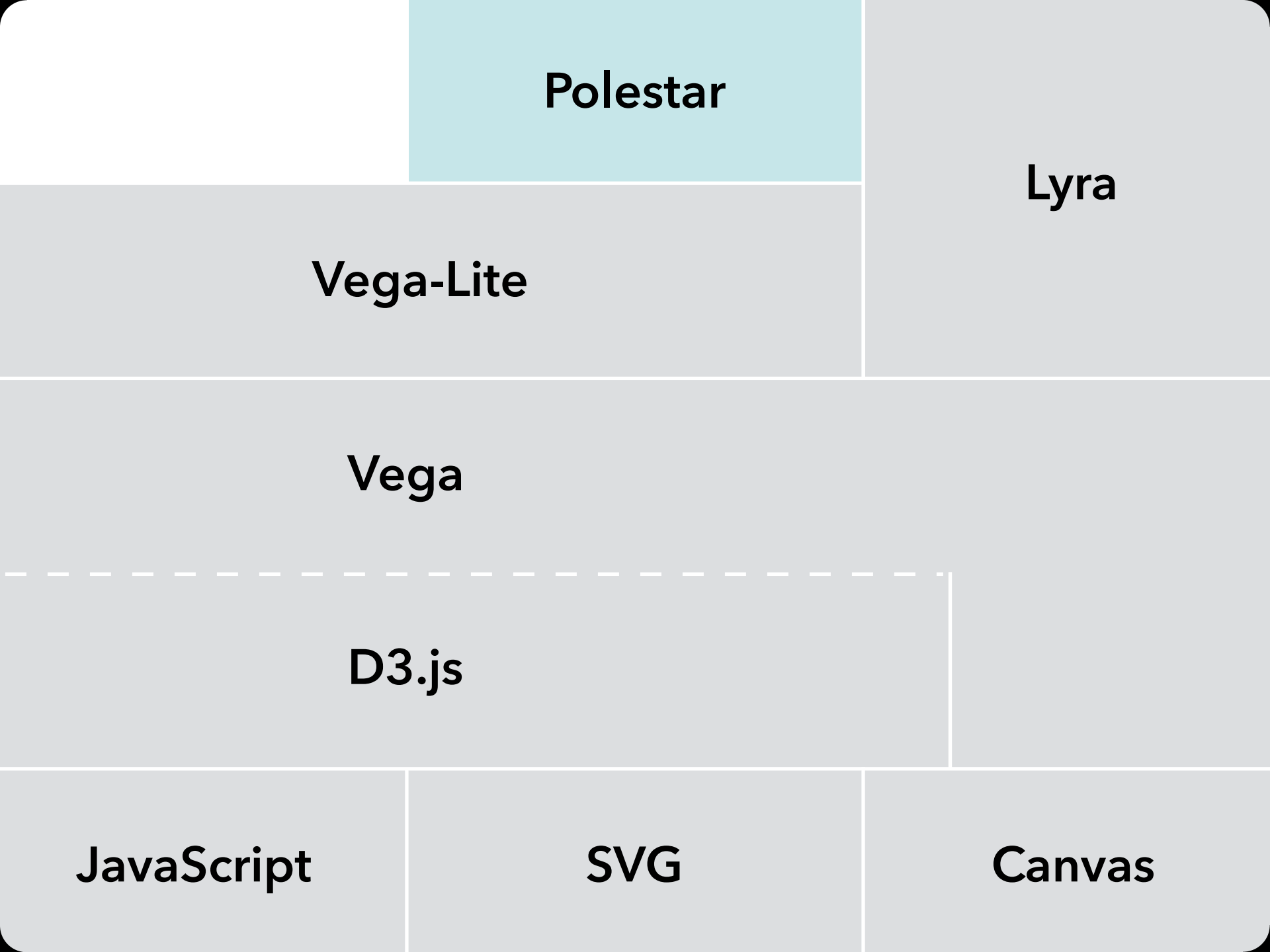
- x: # BIN (Horse
- y: # BIN (Miles per
- col: drop a field here
- row: drop a field here

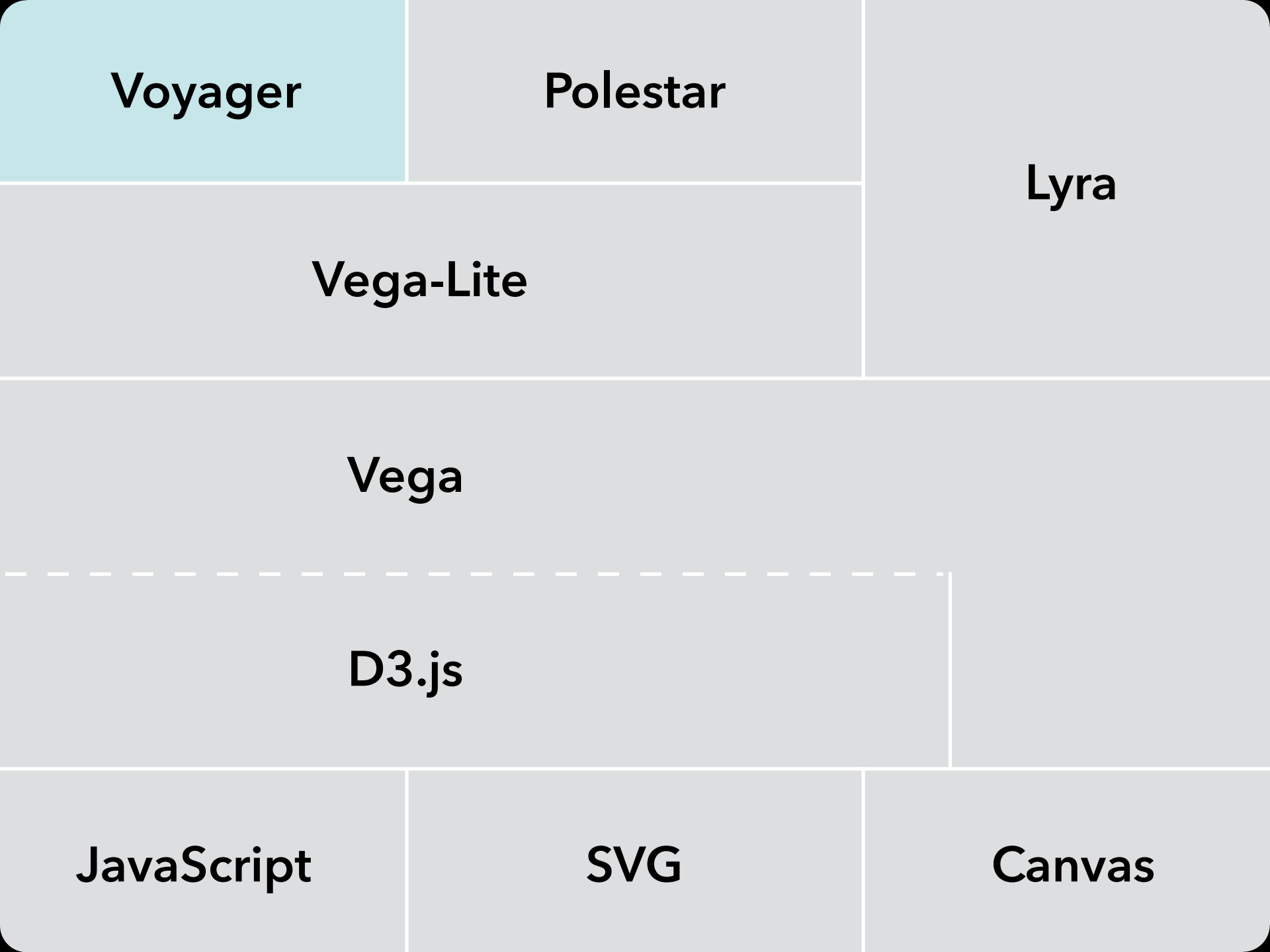
Marks

point

- size: # COUNT
- color: drop a field here
- shape: drop a field here
- detail: drop a field here
- text: drop a field here




Polestar
Lyra
Vega-Lite
Vega
D3.js
JavaScript
SVG
Canvas


Voyager
Polestar
Lyra
Vega-Lite
Vega
D3.js
JavaScript
SVG
Canvas

Voyager

Reduce tedious manual specification

Voyager

Reduce tedious manual specification

Support early-stage data exploration

Encourage *data coverage*

Discourage *premature fixation*

Voyager

Reduce tedious manual specification

Support early-stage data exploration
Encourage *data coverage*
Discourage *premature fixation*

Approach: browse a gallery of visualizations

Voyager

Reduce tedious manual specification

Support early-stage data exploration
Encourage *data coverage*
Discourage *premature fixation*

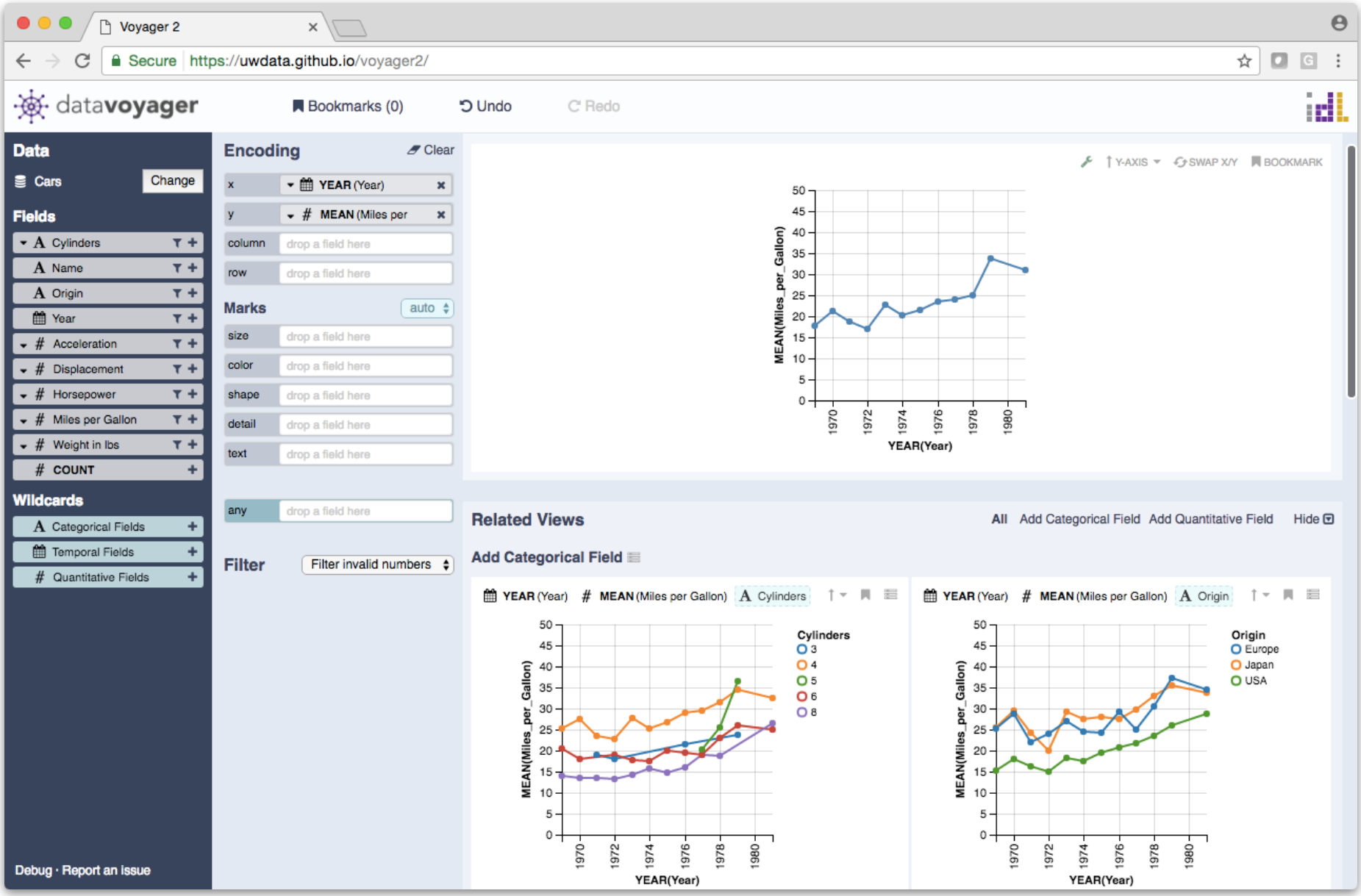
Approach: browse a gallery of visualizations
Challenge - *combinatorial explosion!*

Voyager

Reduce tedious manual specification

Support early-stage data exploration
Encourage *data coverage*
Discourage *premature fixation*

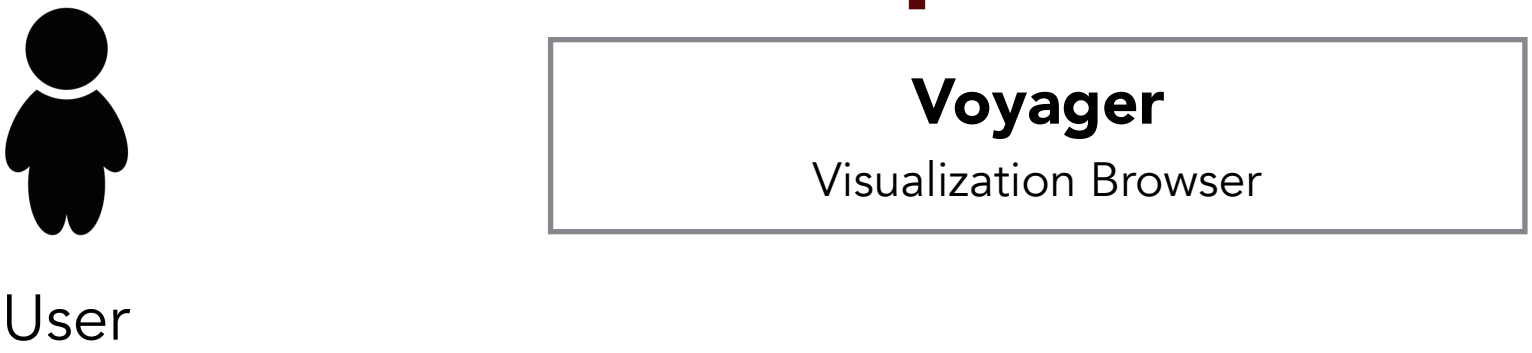
Approach: browse a gallery of visualizations
Challenge - *combinatorial explosion!*
Automatic recommendation of useful views
+ **end-user steering** to focus exploration



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

User


Data Set
Voyager
Visualization Browser
User


CompassQL
Recommendation Engine
Data Schema
& Statistics
Voyager
Visualization Browser
User

1. Select **data variables**
2. Apply **transformations**
3. Pick visual **encodings**

User

CompassQL

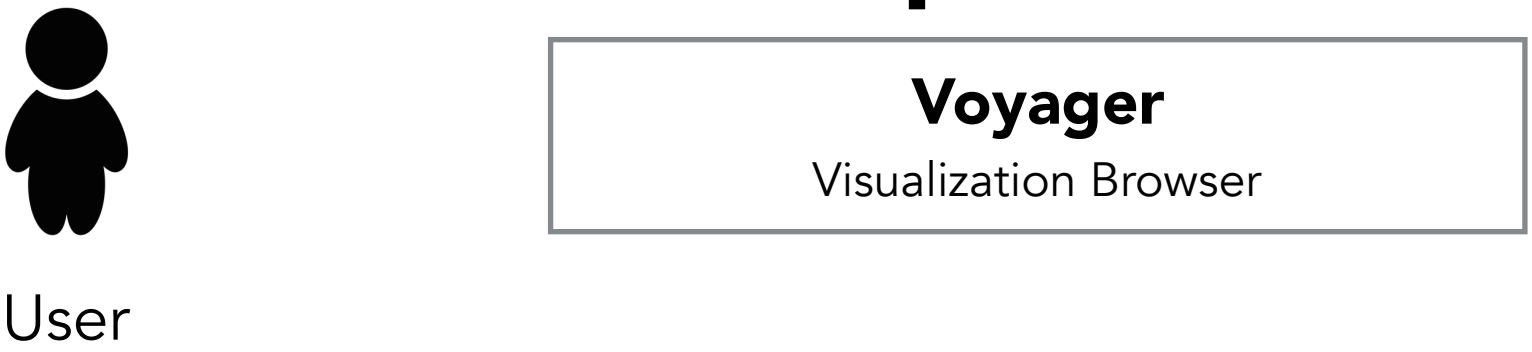
Recommendation Engine

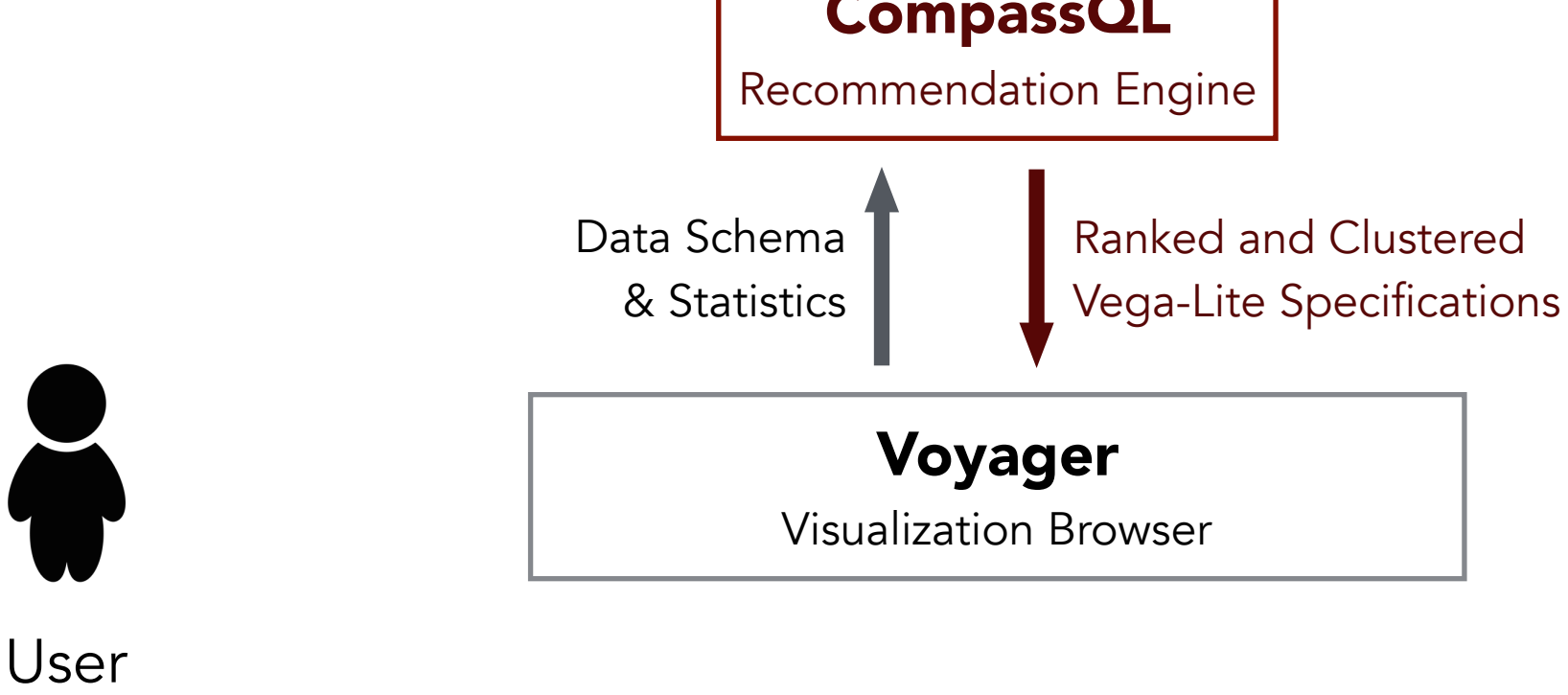
Data Schema
& Statistics

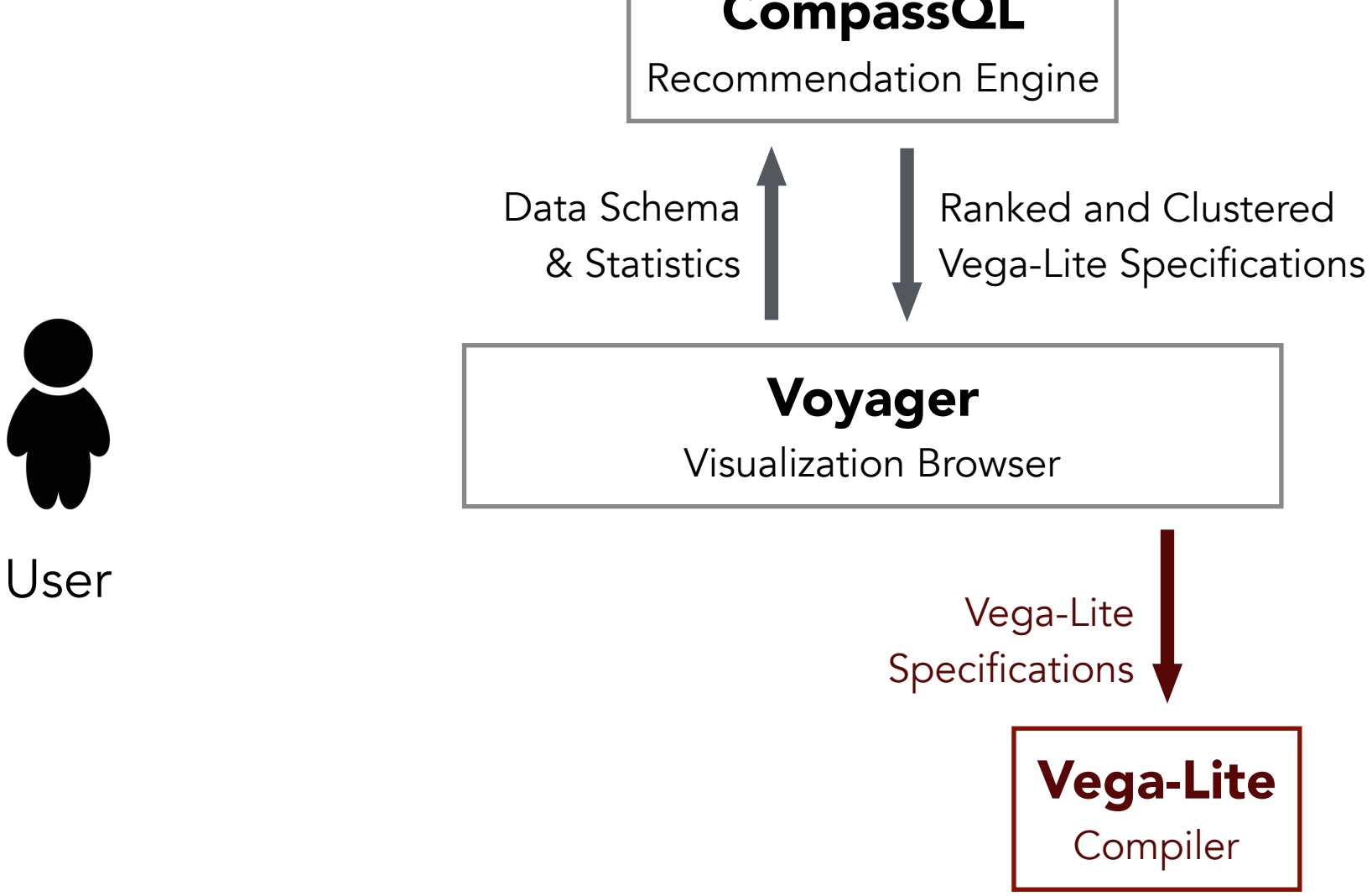
Voyager

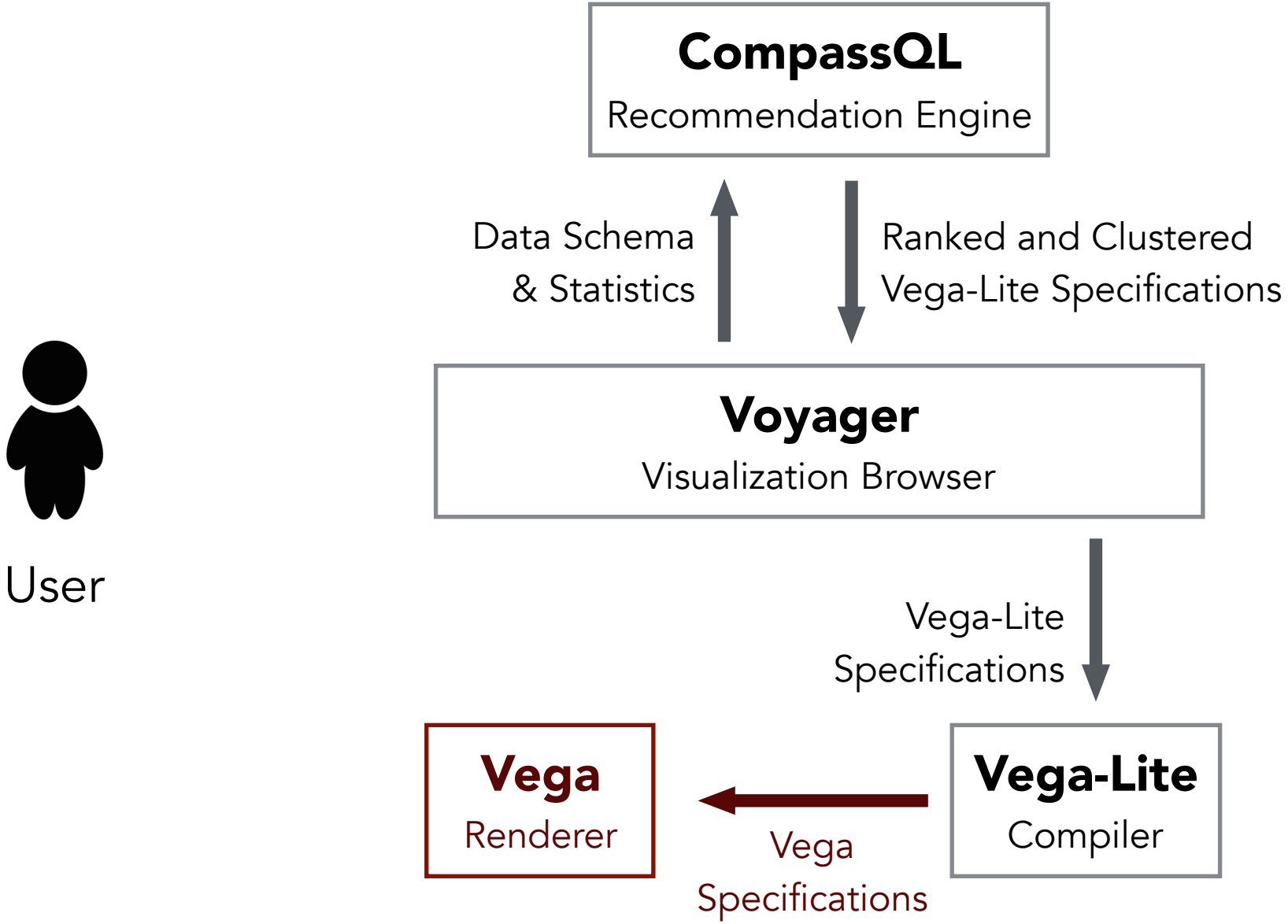
Visualization Browser

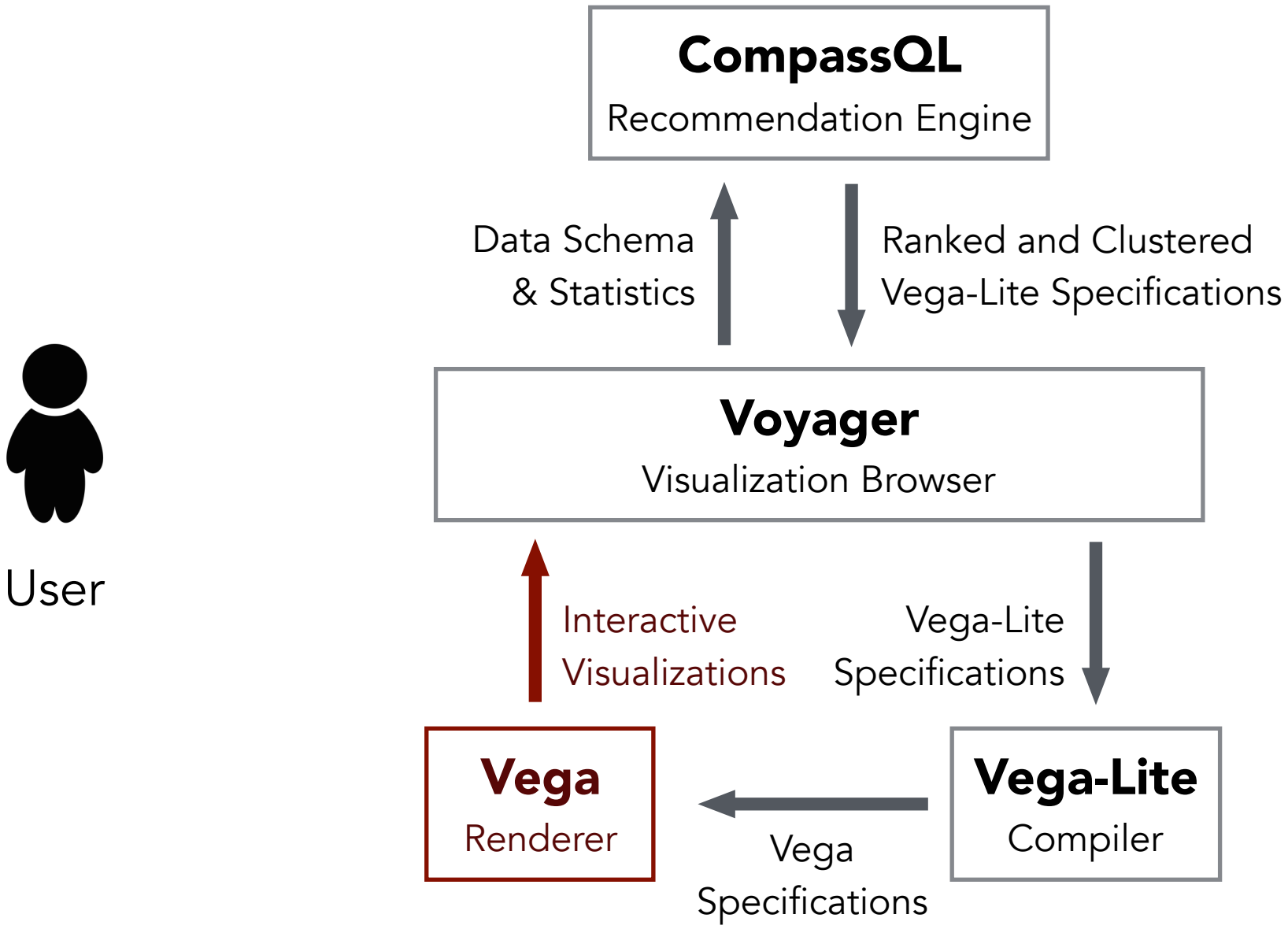
Constrain & rank choices by **data type**, **statistics** & **perceptual principles.**

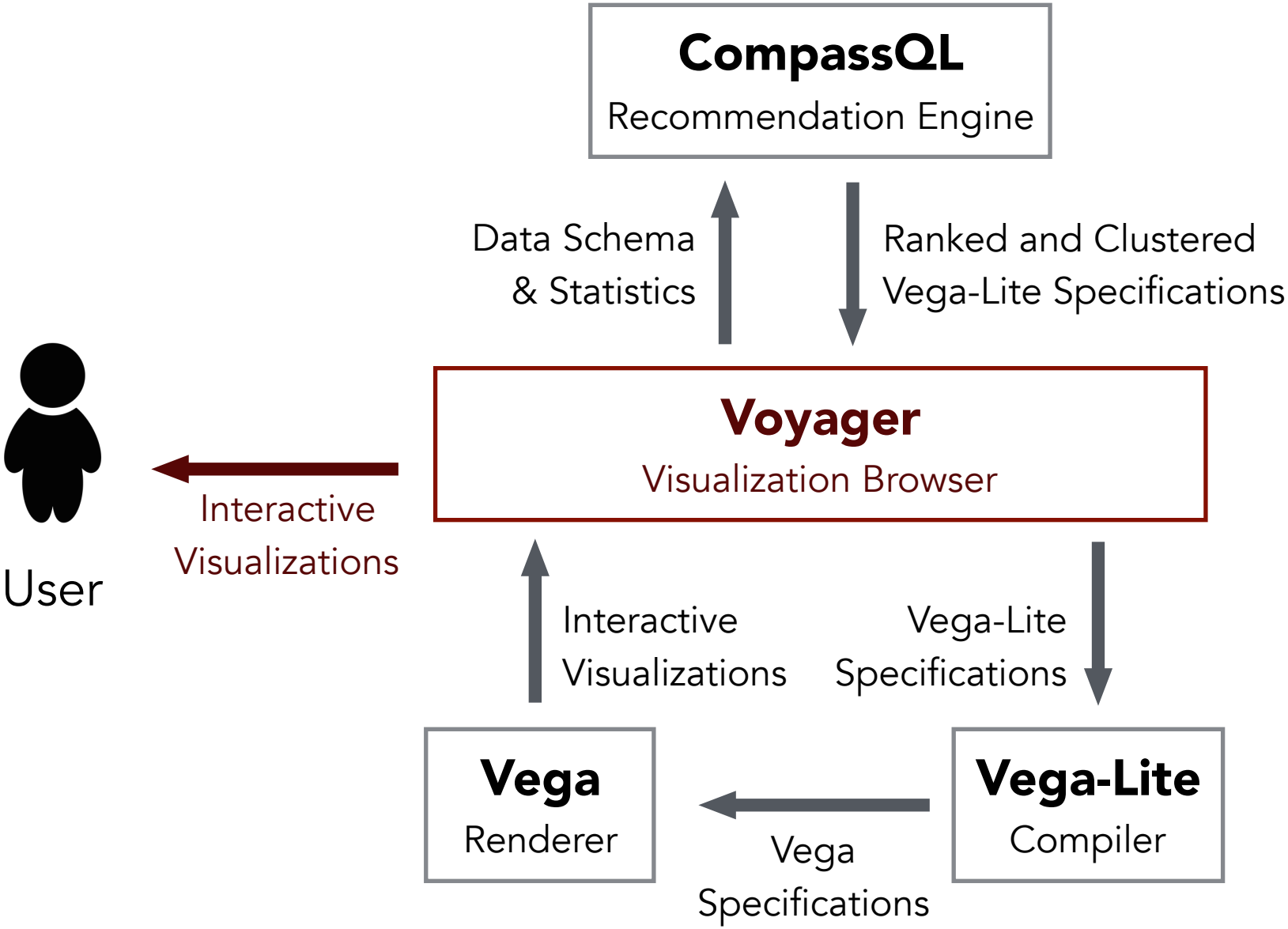


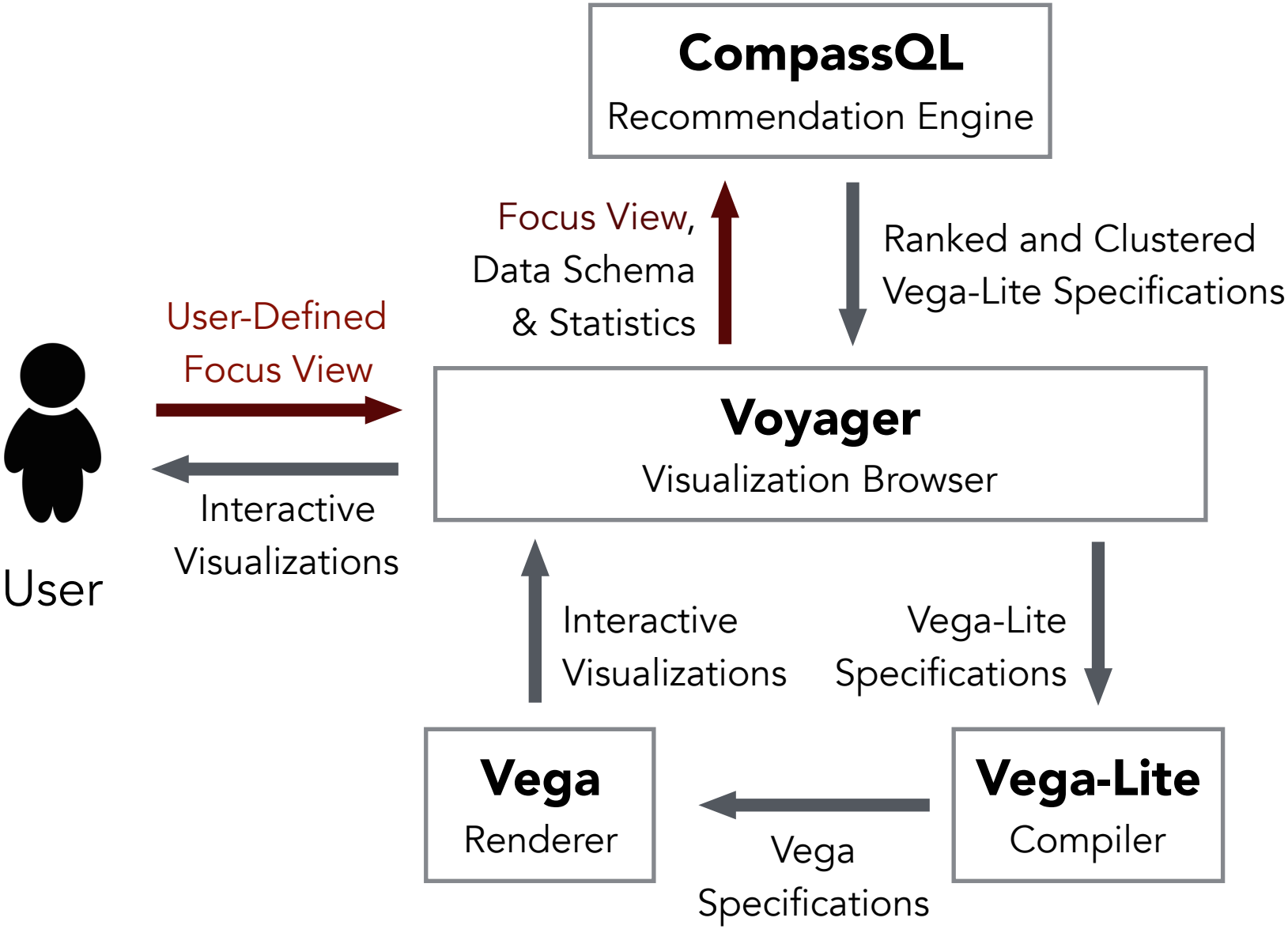

CompassQL
Recommendation Engine
Data Schema
& Statistics
Ranked and Clustered
Vega-Lite Specifications
Voyager
Visualization Browser
User


CompassQL
Recommendation Engine
Data Schema
& Statistics
Ranked and Clustered
Vega-Lite Specifications
Voyager
Visualization Browser
User
Vega-Lite
Specifications
Vega-Lite
Compiler


CompassQL
Recommendation Engine
Data Schema
& Statistics
Ranked and Clustered
Vega-Lite Specifications
Voyager
Visualization Browser
User
Vega-Lite
Specifications
Vega
Renderer
Vega
Specifications
Vega-Lite
Compiler


CompassQL
Recommendation Engine
Data Schema
& Statistics
Ranked and Clustered
Vega-Lite Specifications
Voyager
Visualization Browser
User
Interactive
Visualizations
Vega-Lite
Specifications
Vega
Renderer
Vega
Specifications
Vega-Lite
Compiler

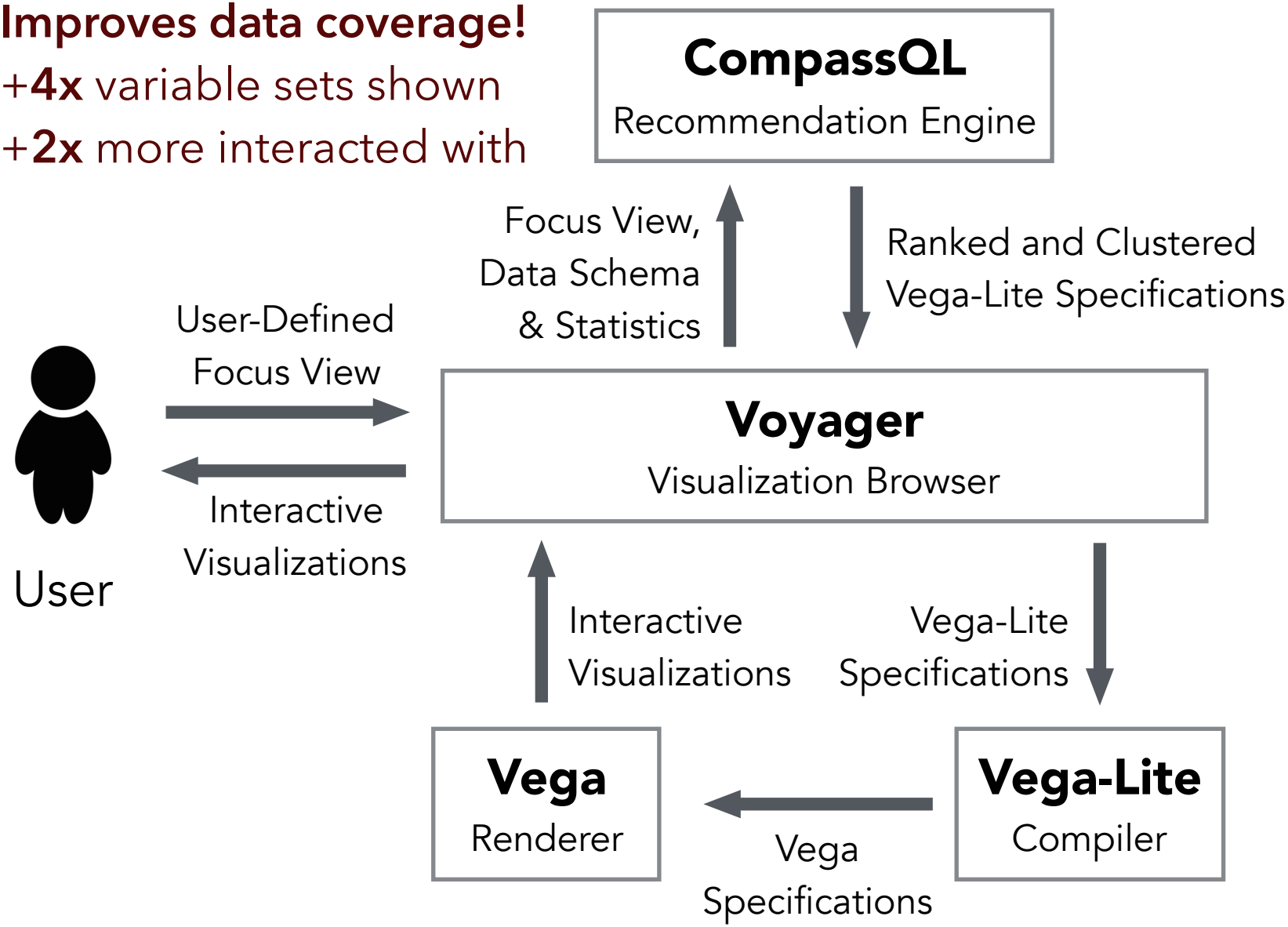

CompassQL
Recommendation Engine
Data Schema
& Statistics
Ranked and Clustered
Vega-Lite Specifications
Voyager
Visualization Browser
User
Interactive
Visualizations
Interactive
Visualizations
Vega-Lite
Specifications
Vega
Renderer
Vega-Lite
Compiler
Vega
Specifications

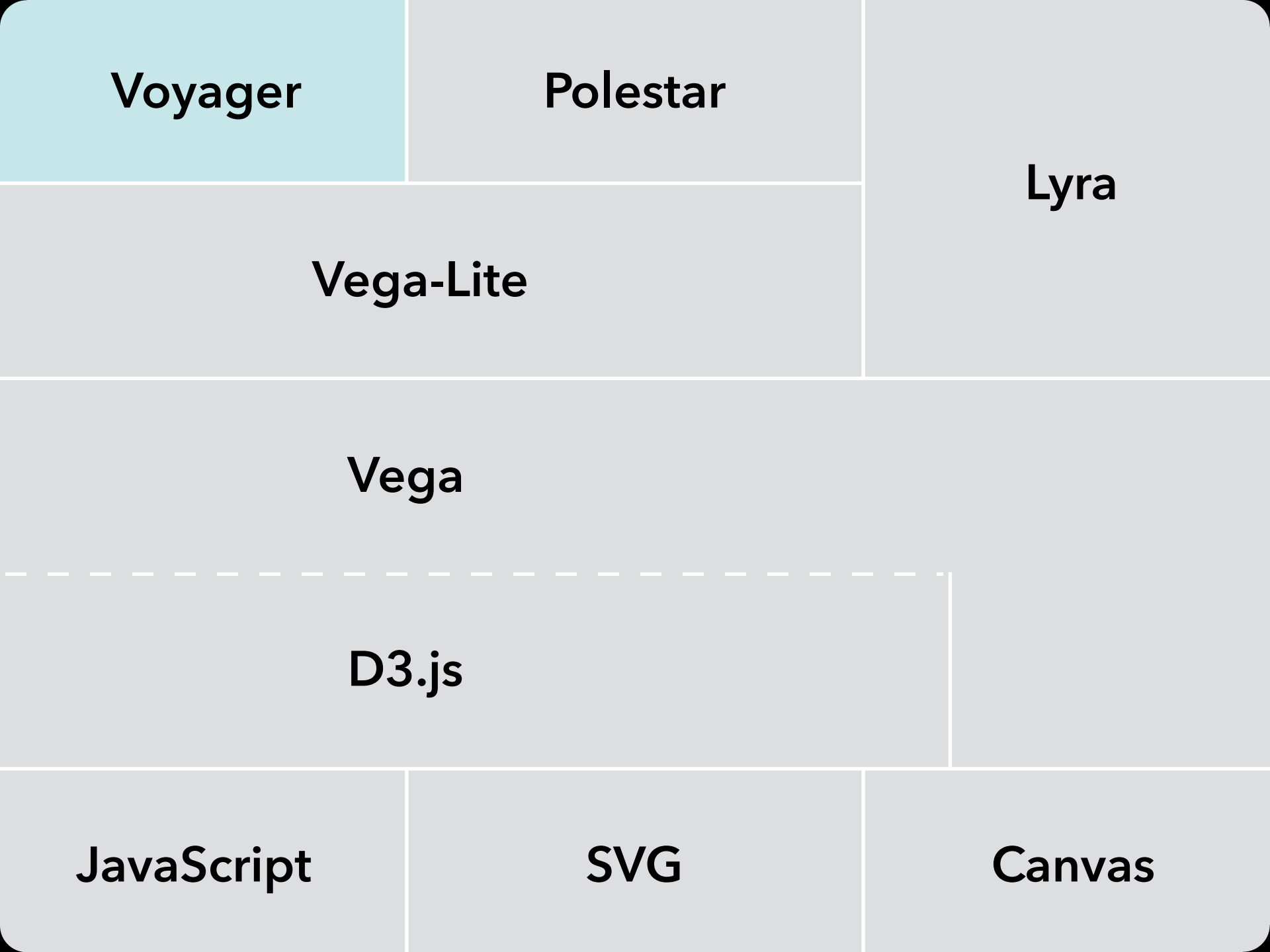

CompassQL
Recommendation Engine
Focus View,
Data Schema
& Statistics
Ranked and Clustered
Vega-Lite Specifications
User-Defined
Focus View
Voyager
Visualization Browser
Interactive
Visualizations
User
Interactive
Visualizations
Vega-Lite
Specifications
Vega
Renderer
Vega-Lite
Compiler
Vega
Specifications

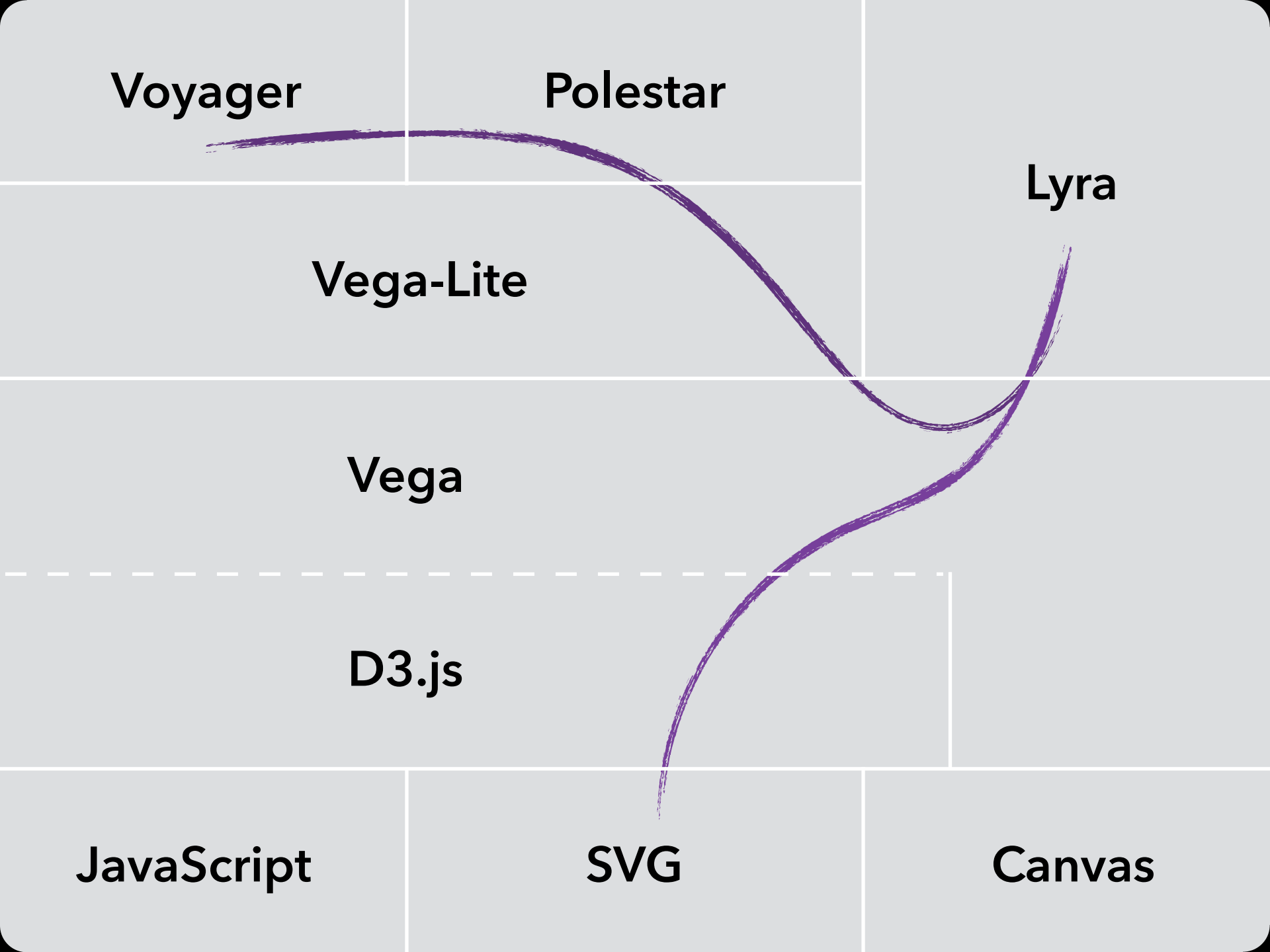
Improves data coverage!

+**4x** variable sets shown

+**2x** more interacted with




Voyager
Polestar
Lyra
Vega-Lite
Vega
D3.js
JavaScript
SVG
Canvas


Voyager
Polestar
Lyra
Vega-Lite
Vega
D3.js
JavaScript
SVG
Canvas

Open Challenges

Designing interactions interactively
How to convey + depict interactions?

Enhancing the "gallery" experience
Rapid assessment of multiple graphics
Embedding large views in small spaces?

Improving visualization recommenders
Learning from users, domain adaptation

Debugging tools [Hoffswell et al. *EuroVis'16*]