

Raising the Bar (Chart)

THE NEXT GENERATION OF VISUALIZATION TOOLS

Jeffrey Heer @jeffrey_heer

Univ. of Washington + Trifacta



HOME

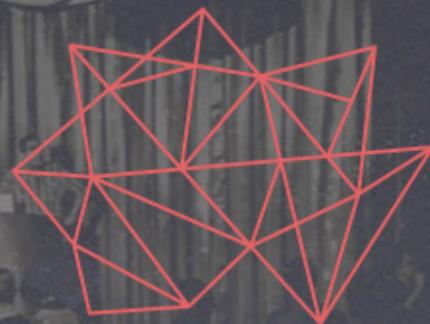
ABOUT

SCHEDULE

TICKETS

VENUE

NEWS



OPENVIS

20 CONFERENCE 15

APR 6-7

BOSTON, MA

[TICKETS]

HOME

ABOUT

SCHEDULE

TICKETS

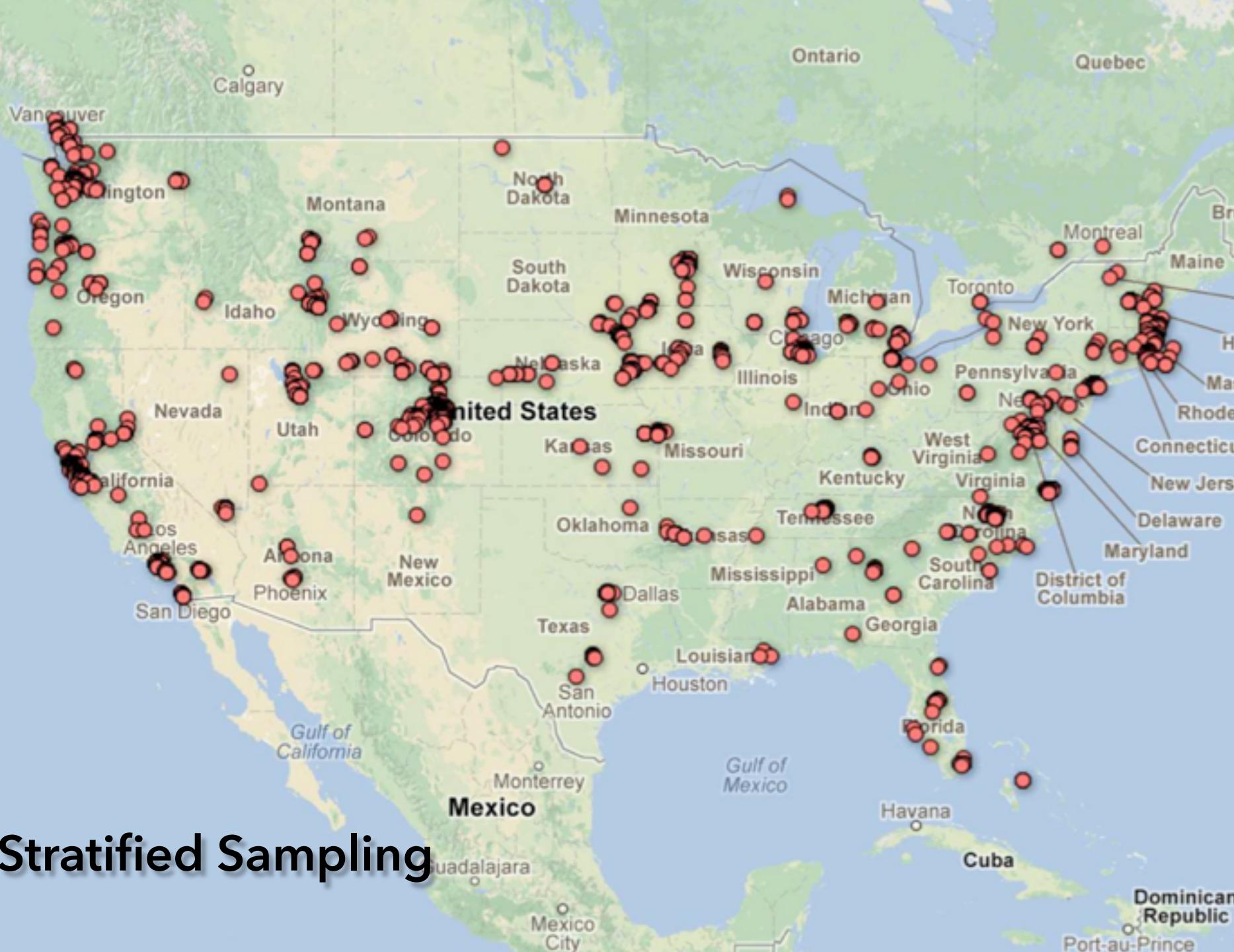
VENUE

NEWS

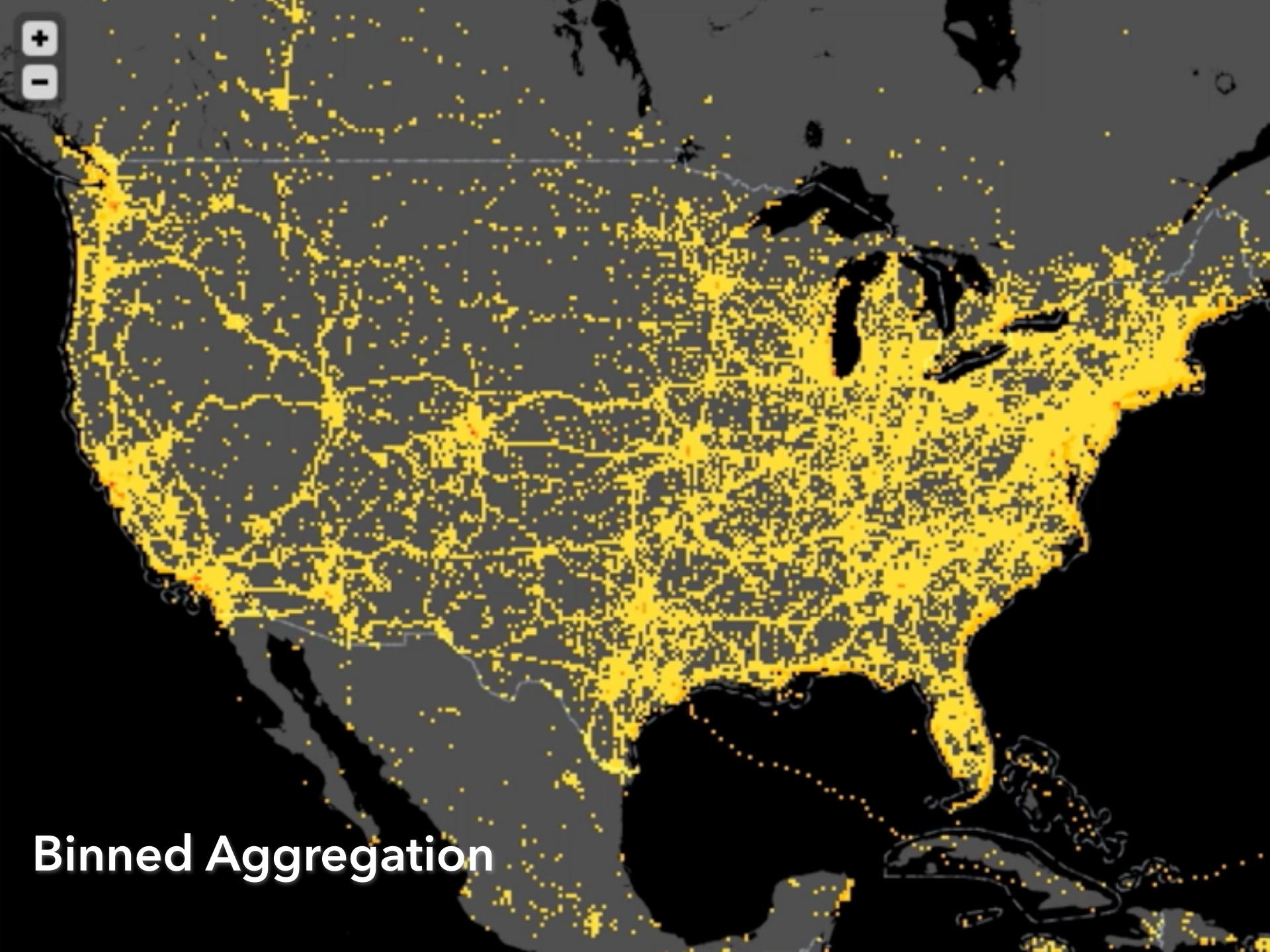


[TICKETS]

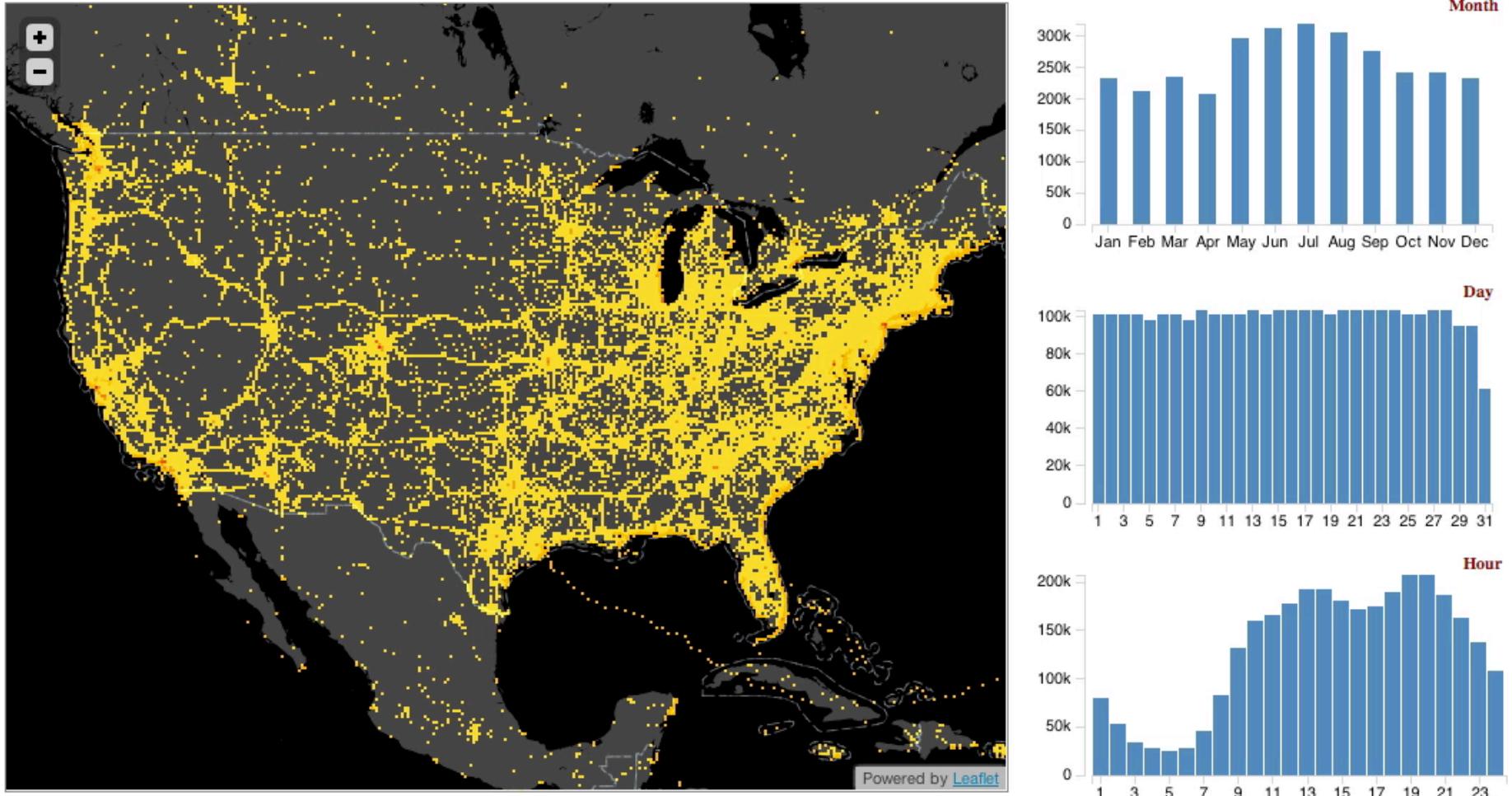
Visualizing Big Data!



Stratified Sampling



Binned Aggregation



imMens: Real-Time Visual Querying of Big Data

with Z. Liu & B. Jiang

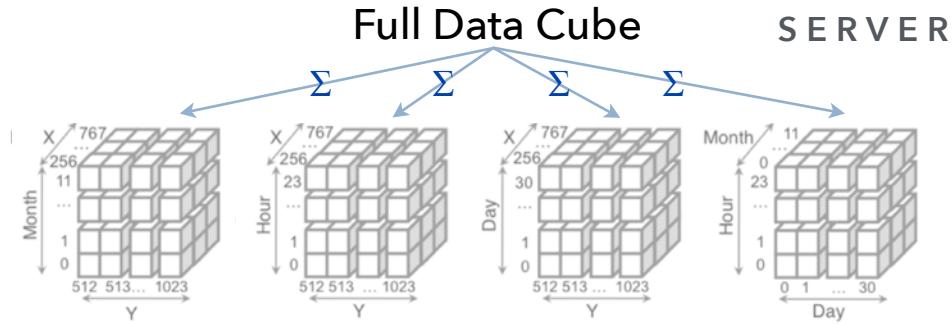
2:15 - 2:45



DANYEL FISHER

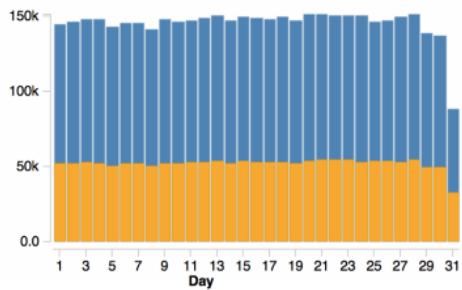
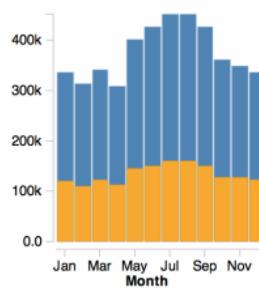
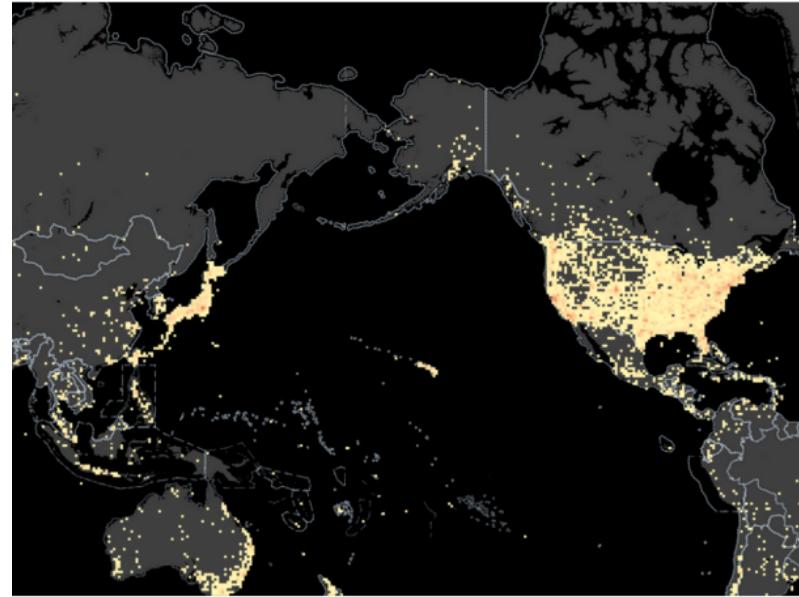
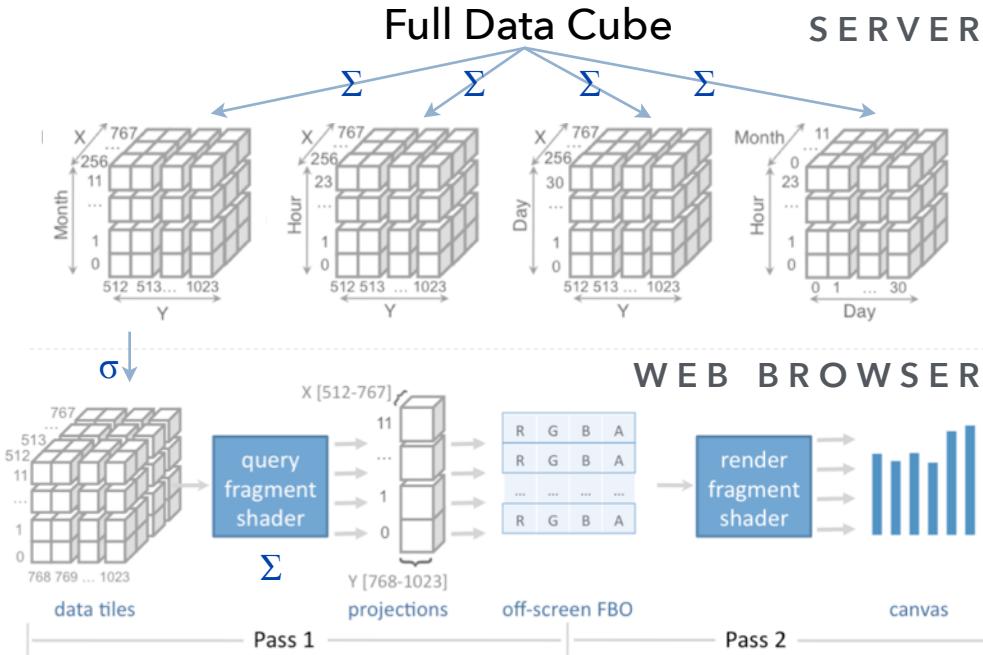
WHY EXPLORING BIG DATA IS
HARD (AND WHAT WE CAN DO
ABOUT IT)

imMens - Real-time Querying of Big Data



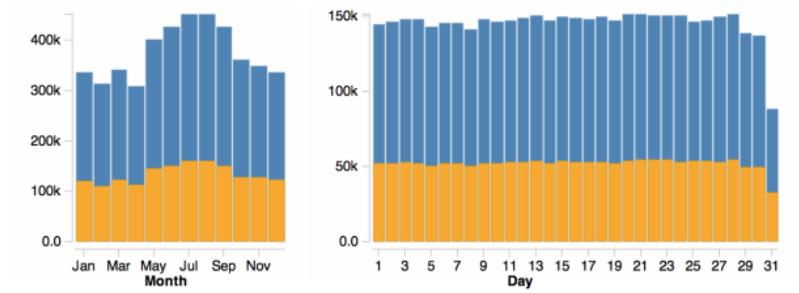
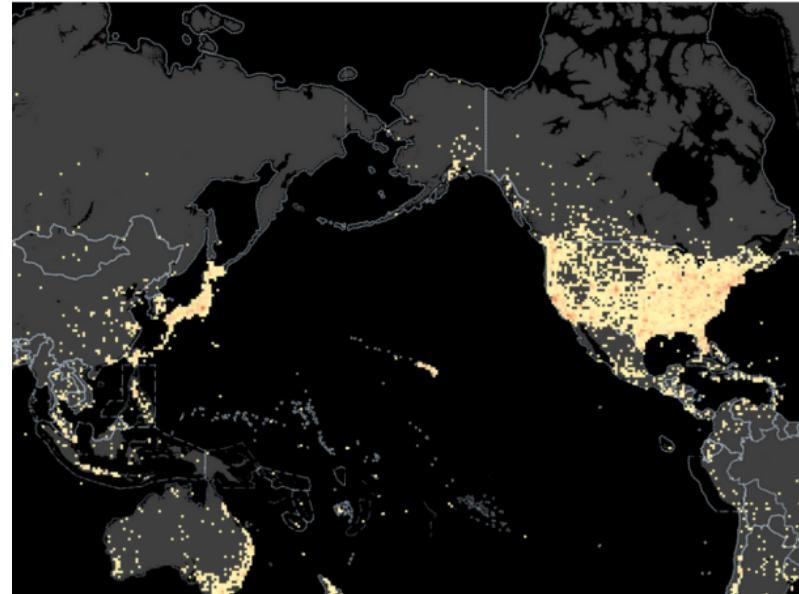
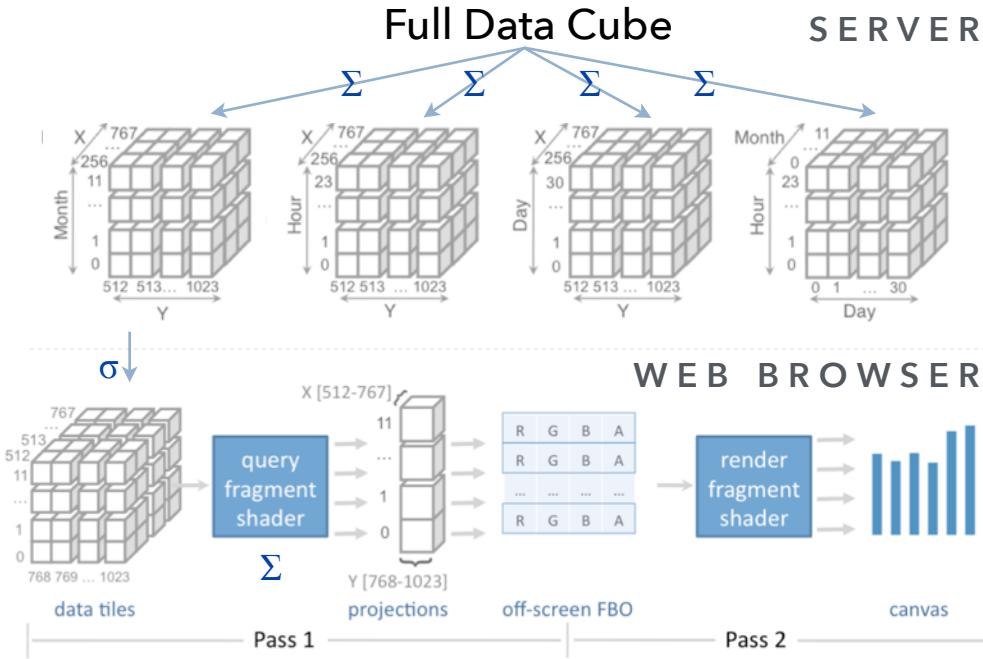
with **Z. Liu, B. Jiang** [EuroVis '13]

imMens - Real-time Querying of Big Data



with Z. Liu, B. Jiang [EuroVis '13]

imMens - Real-time Querying of Big Data



50 fps interactive querying of summary visualizations of **billions of data points**.

with **Z. Liu, B. Jiang** [EuroVis '13]

WebGL!

1:30 - 2:15



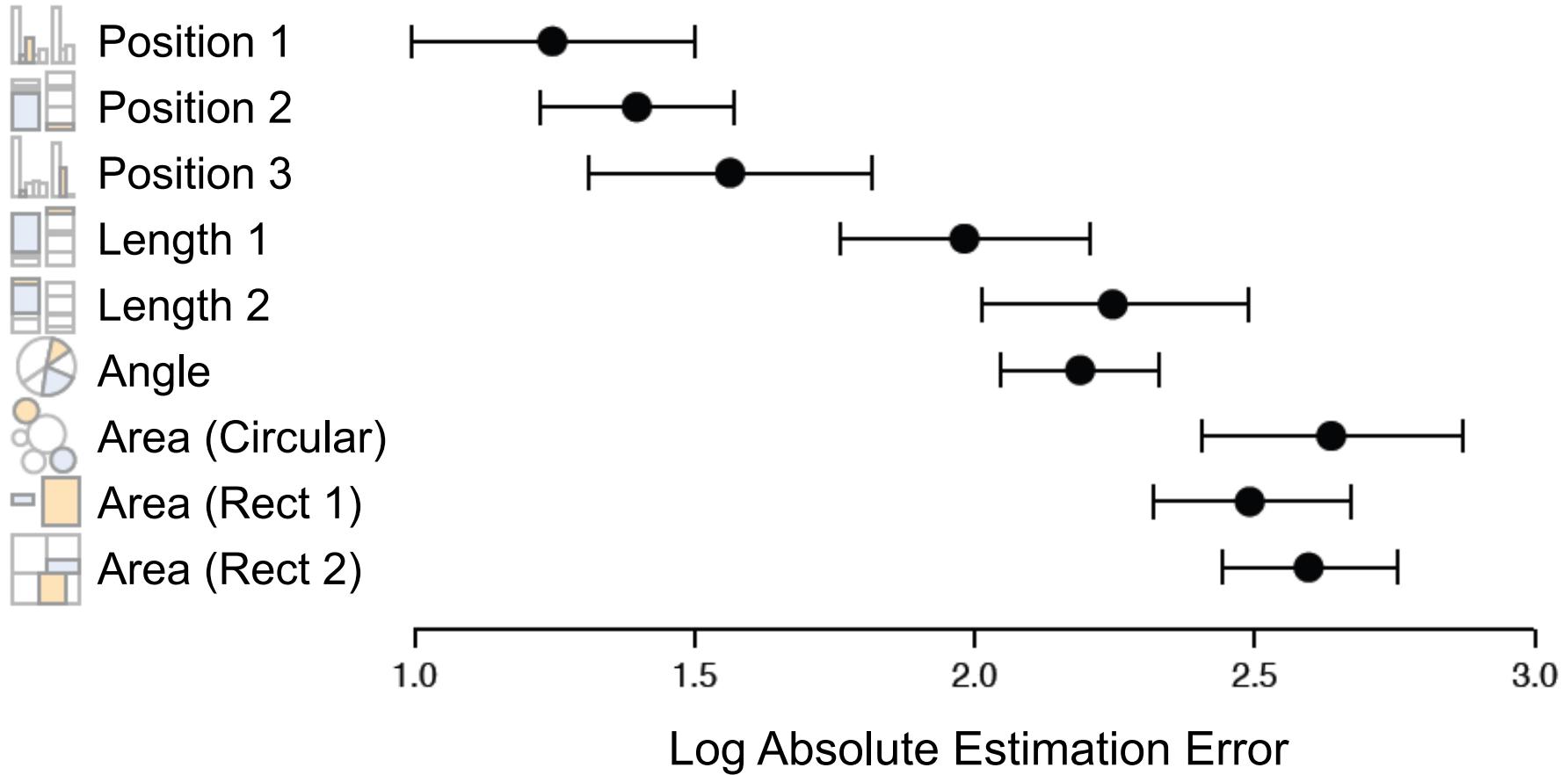
NICOLAS GARCIA BELMONTE
WEBGL FOR GRAPHICS AND
DATA VISUALIZATION

1:30 - 2:15



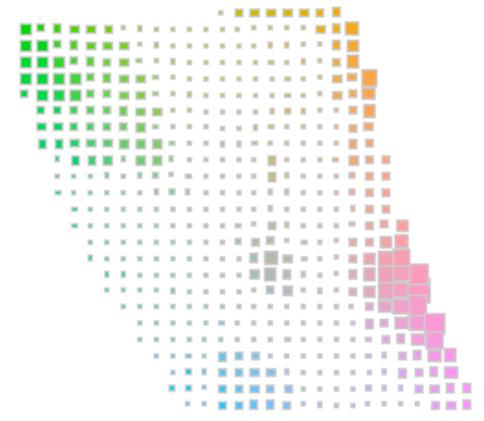
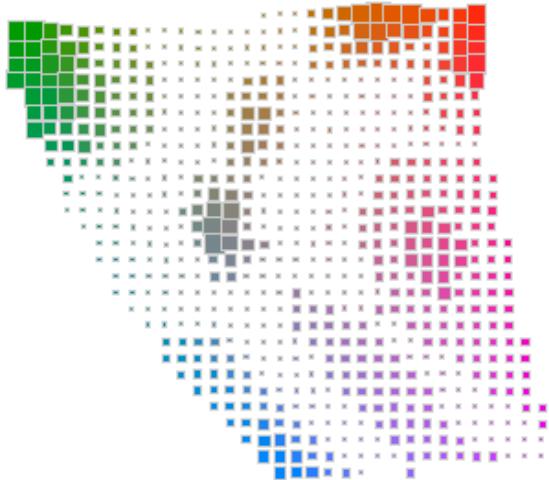
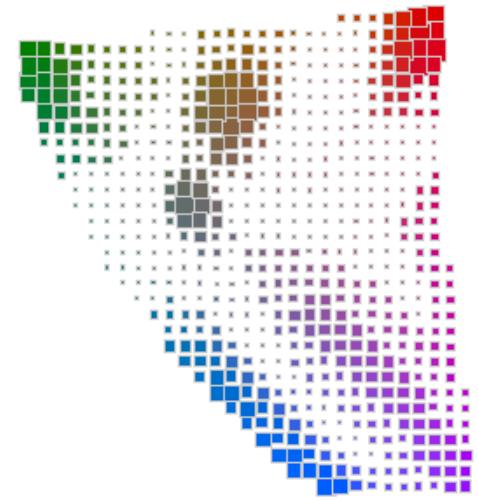
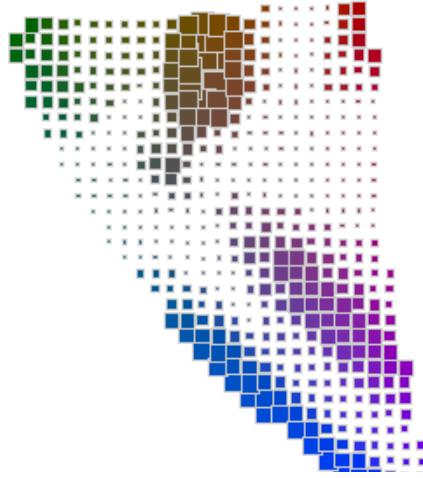
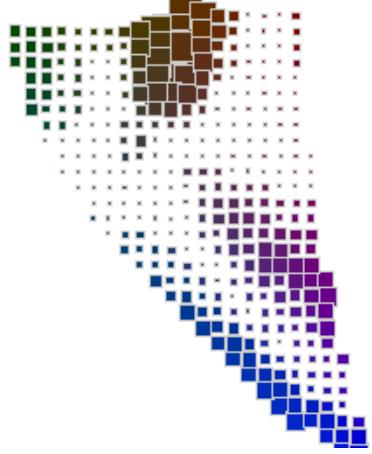
DOMINIKUS BAUR
WEIGHING PERFORMANCE
AGAINST PAIN: SVG, CANVAS
AND WEBGL

Perception!



Graphical Perception [CHI'09, CHI'10, InfoVis'10...]

Experiments to assess the effectiveness of visual encodings.



Cognitive Color Models [CHI '12, EuroVis '13]

Novel measures informing color-concept maps & palette design.

12:00 - 12:30



LANE HARRISON
USER-CENTERED
VISUALIZATION RESEARCH

HOME

ABOUT

SCHEDULE

TICKETS

VENUE

NEWS



[TICKETS]

3:00 - 3:30



NIGEL HOLMES USING HUMOR TO INFORM

HOME

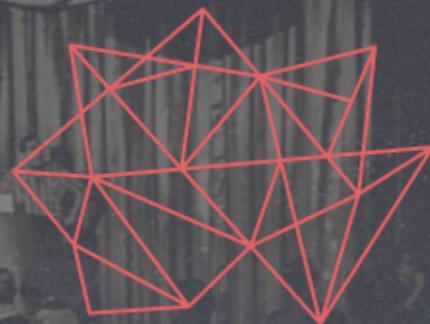
ABOUT

SCHEDULE

TICKETS

VENUE

NEWS



OPENVIS

20 CONFERENCE 15

APR 6-7

BOSTON, MA

[TICKETS]

Visualization Tools

prefuse

Protopvis



vega

 Data-Driven Documents

Raising the Bar (Chart)

THE NEXT GENERATION OF VISUALIZATION TOOLS

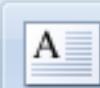
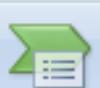
Jeffrey Heer @jeffrey_heer

Univ. of Washington + Trifacta



How might we
create visualizations?

Layout Formulas Data Review View Add-Ins



Column

Line

Pie

Bar

Area

Scatter

Other Charts

Hyperlink

Text Box

Header & Footer

WordArt

Signature Line

Object

Symbol

Charts

Links

Text

fx

L1

B	C	D	E	F	G	H	I	J
	task	row	col	d	d1	d2	L1	L2
-color	SA		1	1	0	0	0	0
-color	SA		1	2	0.63512	0.73568	0	0.73568
-color	SA		1	3	0.59295	0.73119	0	0.73119
-color	SA		1	4	0.45864	0.24612	0	0.24612
-color	SA		1	5	0.49399	0	0.8119	0.8119
-color	SA		1	6	0.85906	0.73568	0.8119	1.54758
-color	SA		1	7	0.81534	0.73119	0.8119	1.54309
-color	SA		1	8	0.71854	0.24612	0.8119	1.05802
-color	SA		1	9	0.57947	0	0.8417	0.8417
-color	SA		1	10	0.91546	0.73568	0.8417	1.57738
-color	SA		1	11	0.91988	0.73119	0.8417	1.57289
-color	SA		1	12	0.77741	0.24612	0.8417	1.08782
-color	SA		1	13	0.29483	0	0.2825	0.2825
-color	SA		1	14	0.75357	0.73568	0.2825	1.01818
-color	SA		1	15	0.73206	0.73119	0.2825	1.01369
-color	SA		1	16	0.53588	0.24612	0.2825	0.52862

Screenshot of Microsoft Excel showing the ribbon menu and a chart creation dialog box.

The ribbon menu includes tabs: Layout, Formulas, Data, Review, View, Add-Ins, and a chart-specific tab.

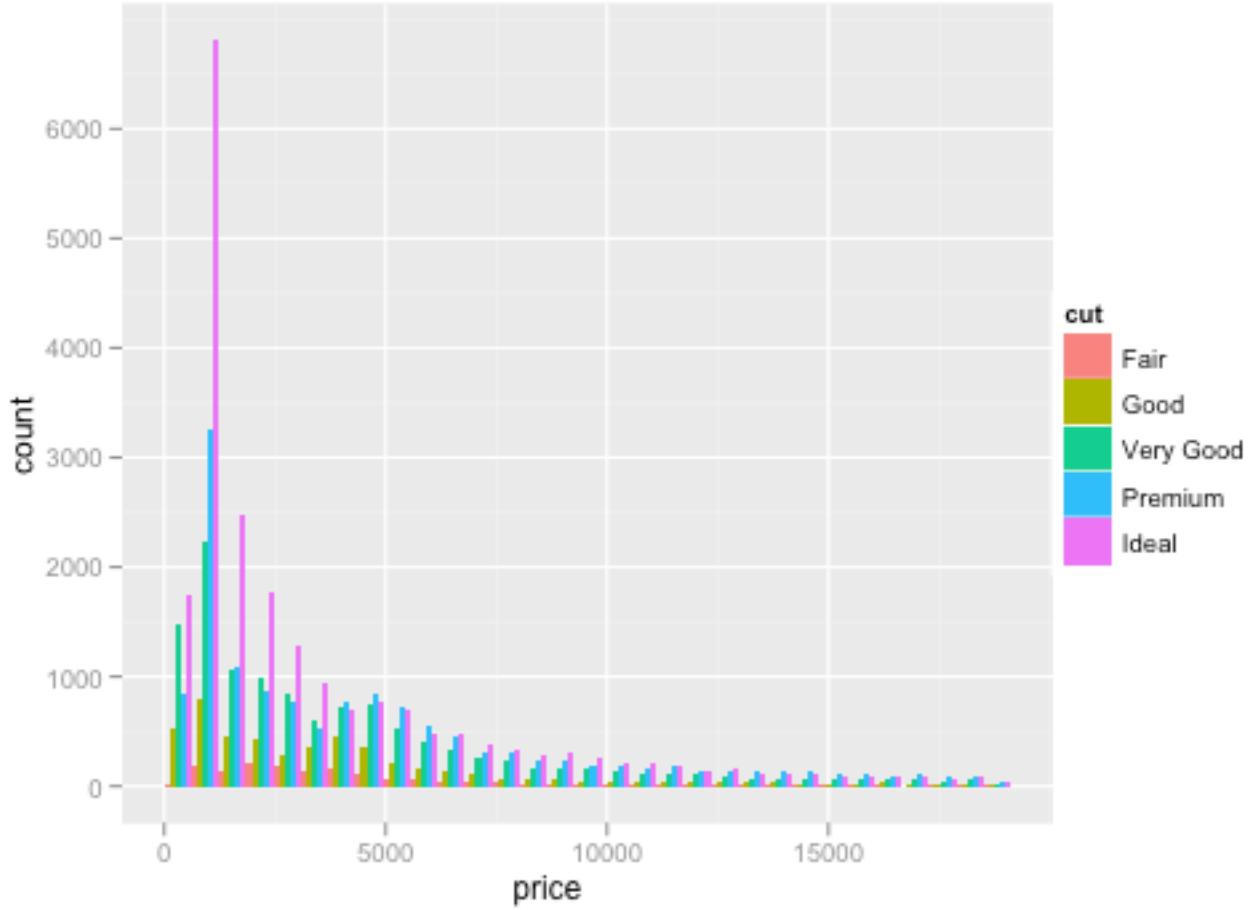
The chart creation dialog box is open, showing categories: 2-D Bar, 3-D Bar, Cylinder, Cone, and Pyramid. Each category has three preview icons.

The main Excel window shows a table with columns B, C, D, and L1. The L1 column contains formulas like =color and =SA.

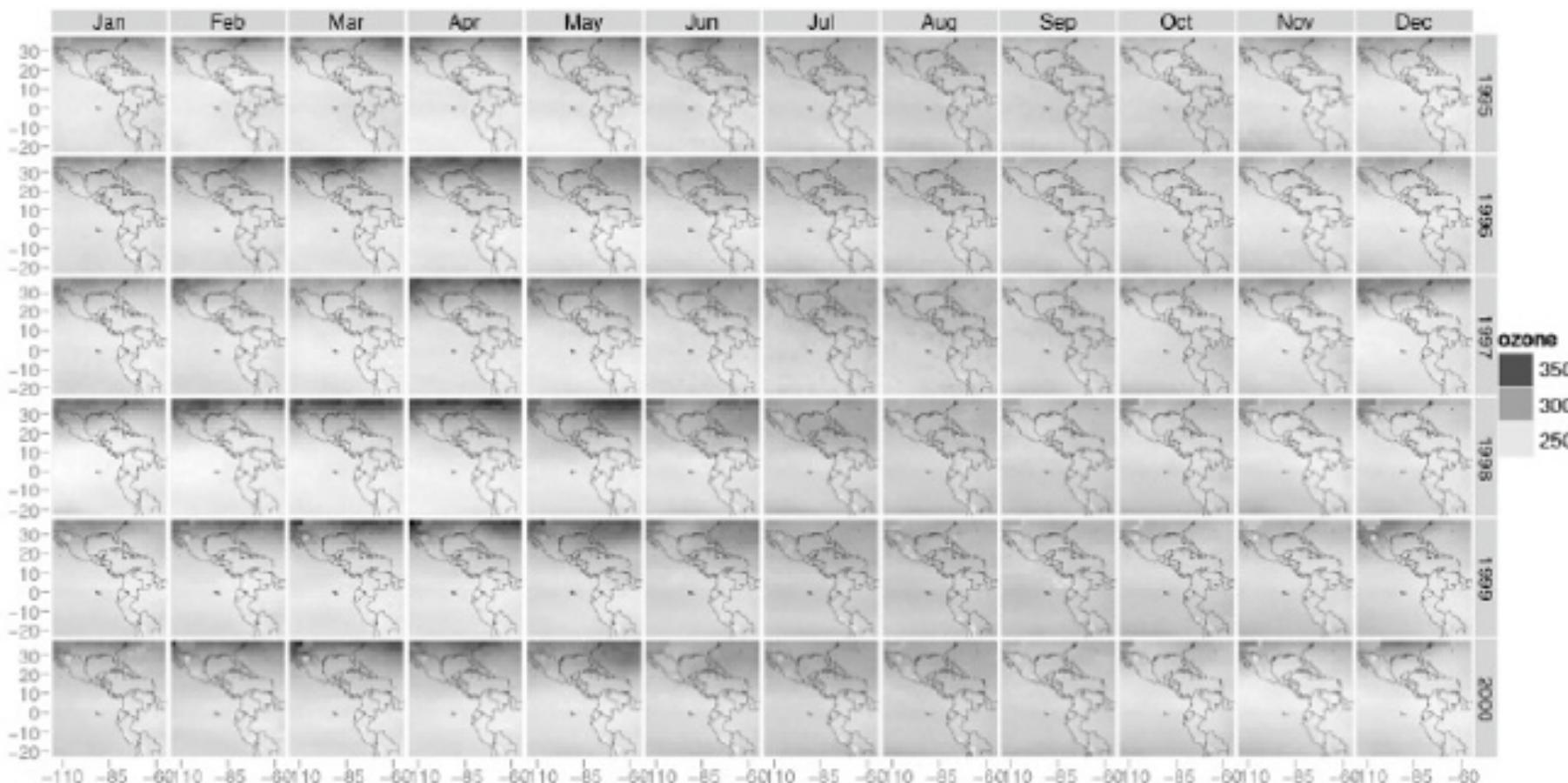
	B	C	D	L1
1	task	row	col	=color
2	SA		1	=color
3	SA		1	=color
4	SA		1	=color
5	SA		1	=color
6	SA		1	=color
7	SA		1	=color
8	SA		1	=color
9	SA		1	=color
10	SA		1	=color
11	SA		1	=color
12	SA		1	=color
13	SA		1	=color
14	SA		1	=color
15	SA		1	=color
16	SA		1	=color
17	SA		1	=color
18	SA		1	=color
19	SA		1	=color
20	SA		1	=color
21	SA		1	=color
22	SA		1	=color
23	SA		1	=color
24	SA		1	=color
25	SA		1	=color
26	SA		1	=color
27	SA		1	=color
28	SA		1	=color
29	SA		1	=color
30	SA		1	=color
31	SA		1	=color
32	SA		1	=color
33	SA		1	=color
34	SA		1	=color
35	SA		1	=color
36	SA		1	=color
37	SA		1	=color
38	SA		1	=color
39	SA		1	=color
40	SA		1	=color
41	SA		1	=color
42	SA		1	=color
43	SA		1	=color
44	SA		1	=color
45	SA		1	=color
46	SA		1	=color
47	SA		1	=color
48	SA		1	=color
49	SA		1	=color
50	SA		1	=color
51	SA		1	=color
52	SA		1	=color
53	SA		1	=color
54	SA		1	=color
55	SA		1	=color
56	SA		1	=color
57	SA		1	=color
58	SA		1	=color
59	SA		1	=color
60	SA		1	=color
61	SA		1	=color
62	SA		1	=color
63	SA		1	=color
64	SA		1	=color
65	SA		1	=color
66	SA		1	=color
67	SA		1	=color
68	SA		1	=color
69	SA		1	=color
70	SA		1	=color
71	SA		1	=color
72	SA		1	=color
73	SA		1	=color
74	SA		1	=color
75	SA		1	=color
76	SA		1	=color
77	SA		1	=color
78	SA		1	=color
79	SA		1	=color
80	SA		1	=color
81	SA		1	=color
82	SA		1	=color
83	SA		1	=color
84	SA		1	=color
85	SA		1	=color
86	SA		1	=color
87	SA		1	=color
88	SA		1	=color
89	SA		1	=color
90	SA		1	=color
91	SA		1	=color
92	SA		1	=color
93	SA		1	=color
94	SA		1	=color
95	SA		1	=color
96	SA		1	=color
97	SA		1	=color
98	SA		1	=color
99	SA		1	=color
100	SA		1	=color
101	SA		1	=color
102	SA		1	=color
103	SA		1	=color
104	SA		1	=color
105	SA		1	=color
106	SA		1	=color
107	SA		1	=color
108	SA		1	=color
109	SA		1	=color
110	SA		1	=color
111	SA		1	=color
112	SA		1	=color
113	SA		1	=color
114	SA		1	=color
115	SA		1	=color
116	SA		1	=color
117	SA		1	=color
118	SA		1	=color
119	SA		1	=color
120	SA		1	=color
121	SA		1	=color
122	SA		1	=color
123	SA		1	=color
124	SA		1	=color
125	SA		1	=color
126	SA		1	=color
127	SA		1	=color
128	SA		1	=color
129	SA		1	=color
130	SA		1	=color
131	SA		1	=color
132	SA		1	=color
133	SA		1	=color
134	SA		1	=color
135	SA		1	=color
136	SA		1	=color
137	SA		1	=color
138	SA		1	=color
139	SA		1	=color
140	SA		1	=color
141	SA		1	=color
142	SA		1	=color
143	SA		1	=color
144	SA		1	=color
145	SA		1	=color
146	SA		1	=color
147	SA		1	=color
148	SA		1	=color
149	SA		1	=color
150	SA		1	=color
151	SA		1	=color
152	SA		1	=color
153	SA		1	=color
154	SA		1	=color
155	SA		1	=color
156	SA		1	=color
157	SA		1	=color
158	SA		1	=color
159	SA		1	=color
160	SA		1	=color
161	SA		1	=color
162	SA		1	=color
163	SA		1	=color
164	SA		1	=color
165	SA		1	=color
166	SA		1	=color
167	SA		1	=color
168	SA		1	=color
169	SA		1	=color
170	SA		1	=color
171	SA		1	=color
172	SA		1	=color
173	SA		1	=color
174	SA		1	=color
175	SA		1	=color
176	SA		1	=color
177	SA		1	=color
178	SA		1	=color
179	SA		1	=color
180	SA		1	=color
181	SA		1	=color
182	SA		1	=color
183	SA		1	=color
184	SA		1	=color
185	SA		1	=color
186	SA		1	=color
187	SA		1	=color
188	SA		1	=color
189	SA		1	=color
190	SA		1	=color
191	SA		1	=color
192	SA		1	=color
193	SA		1	=color
194	SA		1	=color
195	SA		1	=color
196	SA		1	=color
197	SA		1	=color
198	SA		1	=color
199	SA		1	=color
200	SA		1	=color
201	SA		1	=color
202	SA		1	=color
203	SA		1	=color
204	SA		1	=color
205	SA		1	=color
206	SA		1	=color
207	SA		1	=color
208	SA		1	=color
209	SA		1	=color
210	SA		1	=color
211	SA		1	=color
212	SA		1	=color
213	SA		1	=color
214	SA		1	=color
215	SA		1	=color
216	SA		1	=color
217	SA		1	=color
218	SA		1	=color
219	SA		1	=color
220	SA		1	=color
221	SA		1	=color
222	SA		1	=color
223	SA		1	=color
224	SA		1	=color
225	SA		1	=color
226	SA		1	=color
227	SA		1	=color
228	SA		1	=color
229	SA		1	=color
230	SA		1	=color
231	SA		1	=color
232	SA		1	=color
233	SA		1	=color
234	SA		1	=color
235	SA		1	=color
236	SA		1	=color
237	SA		1	=color
238	SA		1	=color
239	SA		1	=color
240	SA		1	=color
241	SA		1	=color
242	SA		1	=color
243	SA		1	=color
244	SA		1	=color
245	SA		1	=color
246	SA		1	=color
247	SA		1	=color
248	SA		1	=color
249	SA		1	=color
250	SA		1	=color
251	SA		1	=color
252	SA		1	=color
253	SA		1	=color
254	SA		1	=color
255	SA		1	=color
256	SA		1	=color
257	SA		1	=color
258	SA		1	=color
259	SA		1	=color
260	SA		1	=color
261	SA		1	=color
262	SA		1	=color
263	SA		1	=color
264	SA		1	=color
265	SA		1	=color
266	SA		1	=color
267	SA		1	=color
268	SA		1	=color
269	SA		1	=color
270	SA		1	=color
271	SA		1	=color
272	SA		1	=color
273	SA		1	=color
274	SA		1	=color
275	SA		1	=color
276	SA		1	=color
277	SA		1	=color
278	SA		1	=color
279	SA		1	=color
280	SA		1	=color
281	SA		1	=color
282	SA		1	=color
283	SA		1	=color
284	SA		1	=color
285	SA		1	=color
286	SA		1	=color
287	SA		1	=color
288	SA		1	=color
289	SA		1	=color
290	SA		1	=color
291	SA		1	=color
292	SA		1	=color
293	SA		1	=color
294	SA		1	=color
295	SA		1	=color
296	SA		1	=color
297	SA		1	=color
298	SA		1	=color
299	SA		1	=color
300	SA		1	=color
301	SA		1	=color
302	SA		1	=color
303	SA		1	=color
304	SA		1	=color
305	SA		1	=color
306	SA		1	=color
307	SA		1	=color
308	SA		1	=color
309	SA		1	=color
310	SA		1	=color
311	SA		1	=color
312	SA		1	=color
313	SA		1	=color
314	SA		1	=color
315	SA		1	=color
316	SA		1	=color
317	SA		1	=color
318	SA		1	=color
319	SA		1	=color
320	SA		1	=color
321	SA		1	=color
322	SA		1	=color
323	SA		1	=color
324	SA		1	=color
325	SA		1	=color
326	SA		1	=color
327	SA		1	=color
328	SA		1	=color
329	SA		1	=color
330	SA		1	=color
331	SA		1	=color
332	SA		1	=color
333	SA		1	=color
334	SA		1	=color
335	SA		1	=color
336	SA		1	=color
337	SA		1	=color
338	SA		1	=color
339	SA		1	=color
340	SA		1	=color
341	SA		1	=color
342	SA		1	=color
343	SA		1	=color
344	SA		1	=color
345	SA		1	=color
346	SA		1	=color
347	SA		1	=color
348	SA		1	=color
349	SA		1	=color
350	SA		1	=color
351	SA		1	=color
352	SA		1	=color
353	SA		1	=color
354	SA		1	=color
355	SA		1	=color
356	SA		1	=color
357	SA		1	=color
358	SA		1	=color
359	SA		1	=color
360	SA		1	=color
361	SA		1	=color
362	SA		1	=color</



```
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
```

```
var svg = div.append("svg:svg")
    .attr("width", w)
    .attr("height", h)
    .append("svg:g")
    .attr("transform", "translate(" + rx + "," + ry + ")");

svg.append("svg:path")
    .attr("class", "arc")
    .attr("d", d3.svg.arc().outerRadius(ry - 120).innerRadius(0).startAngle(0).endAngle(2 * Math.PI))
    .on("mousedown", mousedown);

d3.json("data/flare-imports.json", function(classes) {
  var nodes = cluster.nodes(packages.root(classes)),
      links = packages.imports(nodes),
      splines = bundle(links);

  var path = svg.selectAll("path.link")
    .data(links)
    .enter().append("svg:path")
    .attr("class", function(d) { return "link source-" + d.source.key + " target-" + d.target.key; })
    .attr("d", function(d, i) { return line(splines[i]); });

  svg.selectAll("g.node")
    .data(nodes.filter(function(n) { return !n.children; }))
    .enter().append("svg:g")
    .attr("class", "node")
    .attr("id", function(d) { return "node-" + d.key; })
    .attr("transform", function(d) { return "rotate(" + (d.x - 90) + ")translate(" + d.y + ")"; })
    .append("svg:text")
    .attr("dx", function(d) { return d.x < 180 ? 8 : -8; })
    .attr("dy", ".31em")
    .attr("text-anchor", function(d) { return d.x < 180 ? "start" : "end"; })
    .attr("transform", function(d) { return d.x < 180 ? null : "rotate(180)"; })
    .text(function(d) { return d.key; })
    .on("mouseover", mouseover)
    .on("mouseout", mouseout);

  d3.select("input[type=range]").on("change", function() {
    line.tension(this.value / 100);
    path.attr("d", function(d, i) { return line(splines[i]); });
  });
});
```



Graphics APIs

Processing, OpenGL, Java2D

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

Excel, Many Eyes, Google Charts

Visual Analysis Grammars

VizQL, ggplot2

Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Processing, OpenGL, Java2D

Ease-of-Use



Chart Typologies

Excel, Many Eyes, Google Charts

Visual Analysis Grammars

VizQL, ggplot2



Expressiveness

Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Processing, OpenGL, Java2D

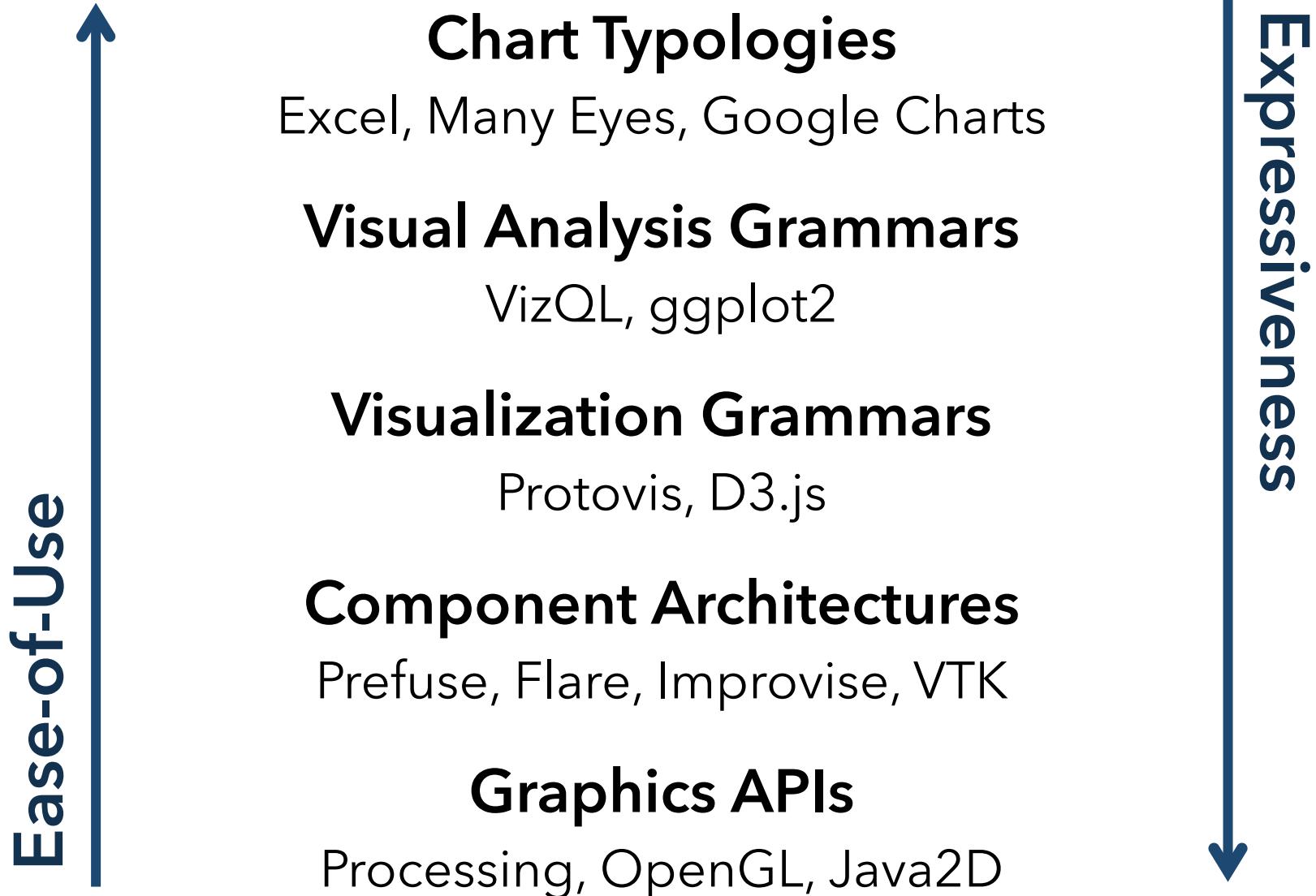


Chart Typologies

Excel, Many Eyes, Google Charts

Charting
Tools

Visual Analysis Grammars

VizQL, ggplot2

Declarative
Languages

Visualization Grammars

Protopis, 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

Declarative
Languages

Visualization Grammars

Protopis, D3.js

Programming
Toolkits

Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Processing, OpenGL, Java2D

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); })
```



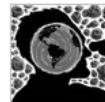
The New York Times

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

ING DIRECT



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

Painting at 99, With No Compromises

By ROBIN FINN

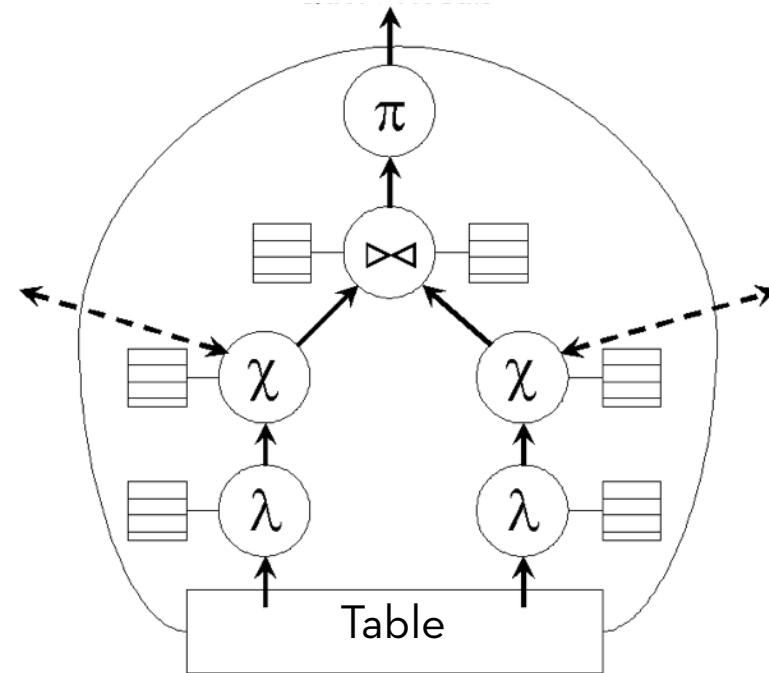
An exhibition celebrating Will Barnet's centennial year traces his evolution as a modern American artist.

Glaxo Pays \$750 Million Fine for Tainted Products

By GARDNER HARRIS and DUFF

```
<!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"></script>
    <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>
    
    <span id="toScript"></span>
    <script type="text/javascript">...</script>
    
    <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

Chart Typologies

Excel, Many Eyes, Google Charts

Charting
Tools

Visual Analysis Grammars

VizQL, ggplot2

Declarative
Languages

Visualization Grammars

Protopis, 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, *Vegalite*

Declarative
Languages

Visualization Grammars

Protopis, D3.js, *Vega*

Programming
Toolkits

Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Processing, OpenGL, Java2D

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, *Vegalite*

Declarative
Languages

Visualization Grammars

Protopis, D3.js, *Vega*

Programming
Toolkits

Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Processing, OpenGL, Java2D

?! Charting
Tools

Chart Typologies

Excel, Many Eyes, Google Charts

Declarative
Languages

Visual Analysis Grammars

VizQL, ggplot2, *Vegalite*

Programming
Toolkits

Visualization Grammars

Protopis, D3.js, *Vega*

Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Processing, OpenGL, Java2D

Visual Analysis Grammars

VizQL, ggplot2, *Vegalite*

Declarative
Languages

Visualization Grammars

Protopis, D3.js, *Vega*

Component Architectures

Prefuse, Flare, Improvise, VTK

Programming
Toolkits

Graphics APIs

Processing, OpenGL, Java2D

Interactive Data Exploration

Tableau, *Lyra, Polestar, Voyager*

Graphical
Interfaces

Visual Analysis Grammars

VizQL, ggplot2, *Vegalite*

Declarative
Languages

Visualization Grammars

Protopis, D3.js, *Vega*

Programming
Toolkits

Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Processing, OpenGL, Java2D



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

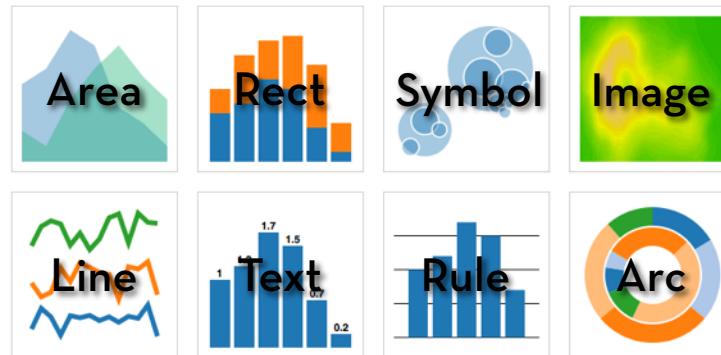
Data	Input data to visualize
Transforms	Grouping, stats, projection, layout
Scales	Map data values to visual values

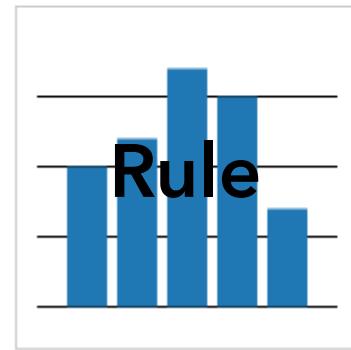
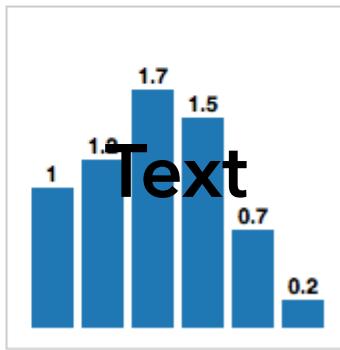
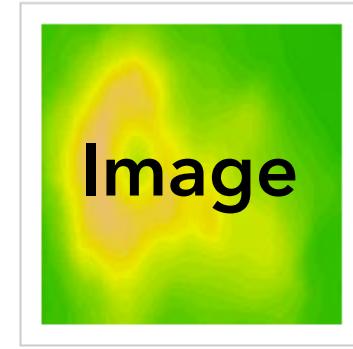
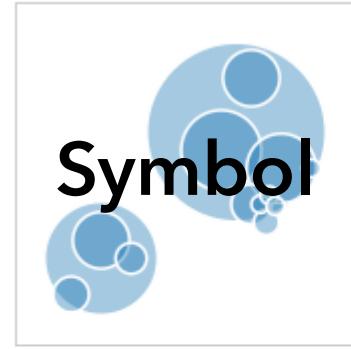
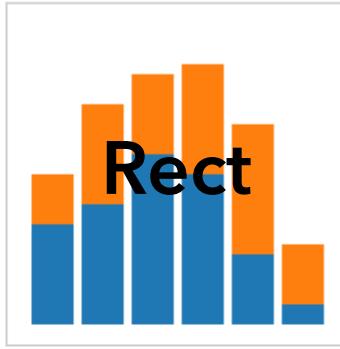
Visualization Grammar

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

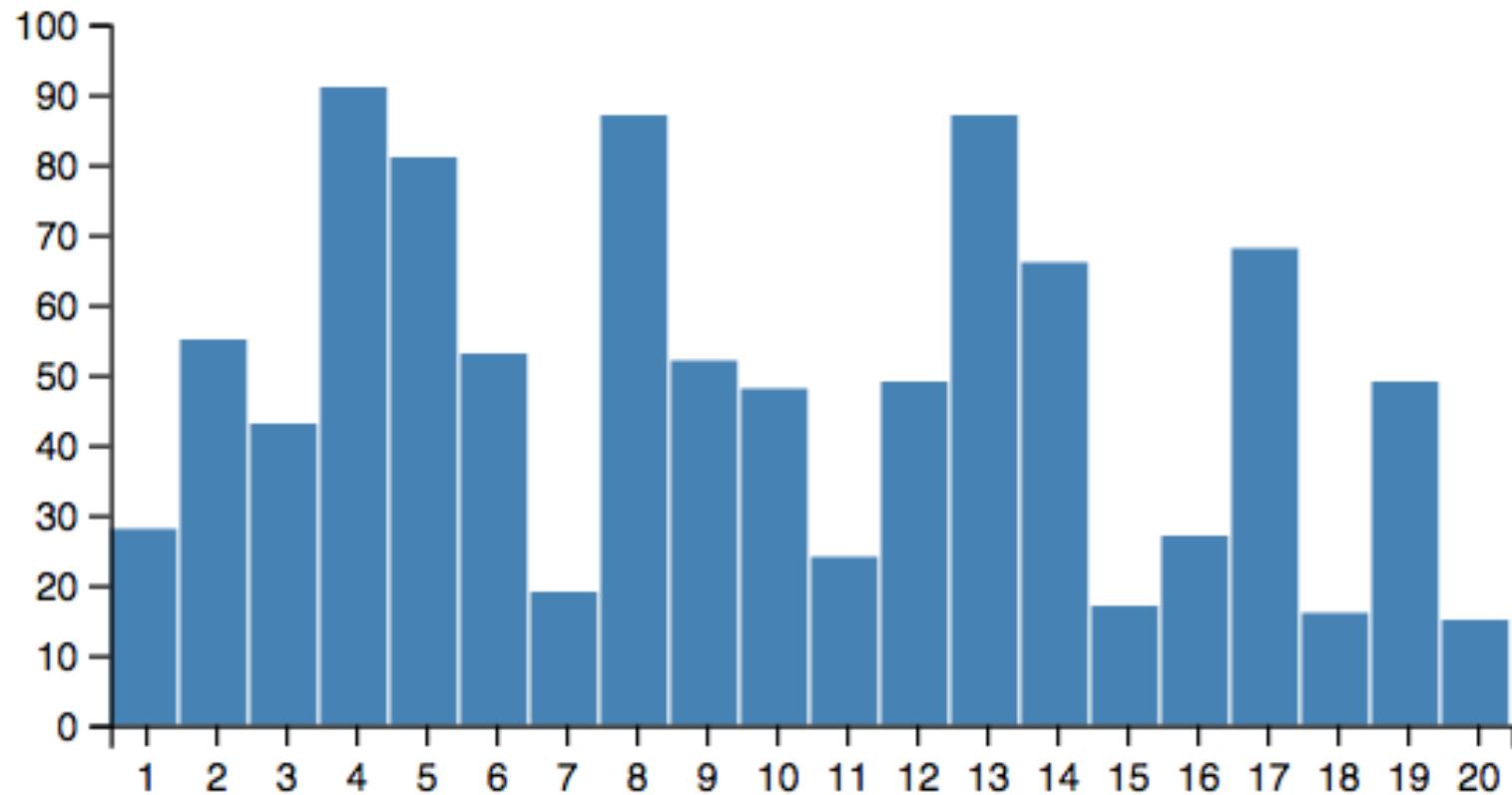
Visualization Grammar

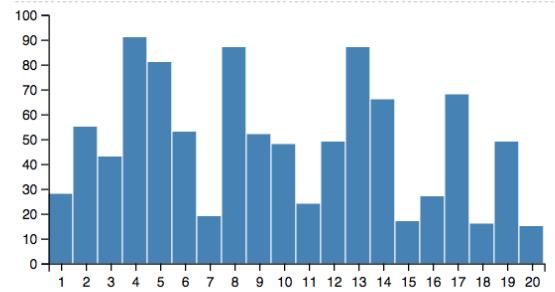
Data	Input data to visualize
Transforms	Grouping, stats, projection, layout
Scales	Map data values to visual values
Guides	Axes & legends visualize scales
Marks	Data-representative graphics



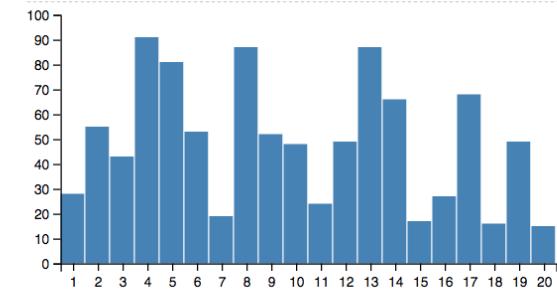


MARKS: Graphical Primitives



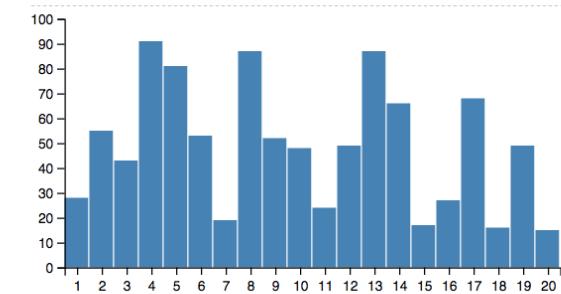


```
{
  "width": 400, "height": 200,
  "data": [
    {"name": "table", "url": "/data/sample.json"}
  ],
  "scales": [
    {
      "name": "x", "type": "ordinal",
      "range": "width",
      "domain": {"data": "table", "field": "x"}
    },
    {
      "name": "y",
      "range": "height", "nice": true,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



Data + Transforms

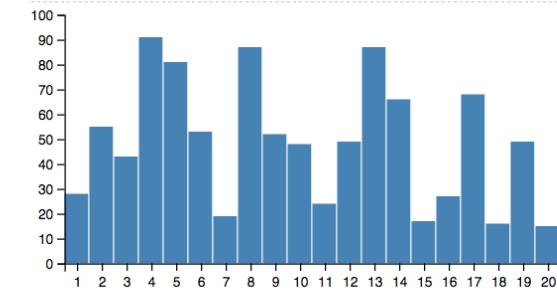
```
{  
  "width": 400, "height": 200,  
  "data": [  
    {"name": "table", "url": "/data/sample.json"}  
,  
  "scales": [  
    {  
      "name": "x", "type": "ordinal",  
      "range": "width",  
      "domain": {"data": "table", "field": "x"}  
    },  
    {  
      "name": "y",  
      "range":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

Data + Transforms

Scales

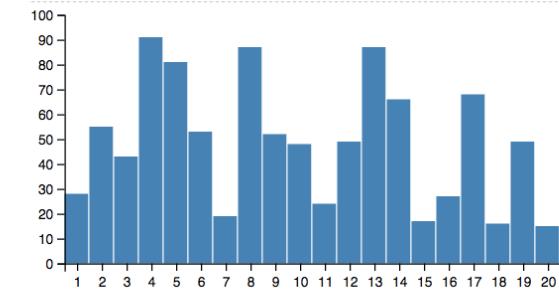


```
{
  "width": 400, "height": 200,
  "data": [
    {"name": "table", "url": "/data/sample.json"}
  ],
  "scales": [
    {
      "name": "x", "type": "ordinal",
      "range": "width",
      "domain": {"data": "table", "field": "x"}
    },
    {
      "name": "y",
      "range": "height", "nice": true,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

Data + Transforms

Scales

Guides



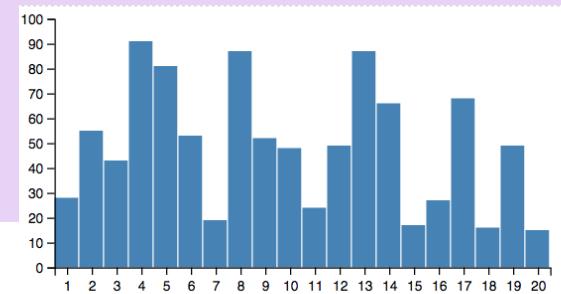
```
{
  "width": 400, "height": 200,
  "data": [
    {"name": "table", "url": "/data/sample.json"}
  ],
  "scales": [
    {
      "name": "x", "type": "ordinal",
      "range": "width",
      "domain": {"data": "table", "field": "x"}
    },
    {
      "name": "y",
      "range": "height", "nice": true,
      "domain": {"data": "table", "field": "y"}
    }
  ],
  "axes": [
    {"type": "x", "scale": "x"}, {"type": "y", "scale": "y"}
  ],
  "marks": [
    {
      "type": "rect",
      "from": {"data": "table"}, (Data + Transforms)
      "properties": {
        "enter": {
          "x": {"scale": "x", "field": "x"}, "width": {"scale": "x", "band": true, "offset": -1},
          "y": {"scale": "y", "field": "y"}, "y2": {"scale": "y", "value": 0}, "fill": {"value": "steelblue"}
        }
      }
    }
  ]
}
```

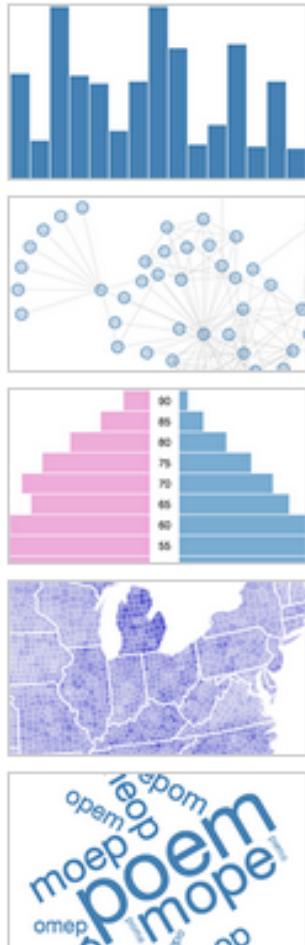
Data + Transforms

Scales

Guides

Marks





vega

[vega.min.js \(188k\)](#)
[Source \(GitHub\)](#)

Vega is a visualization grammar, a declarative format for creating, saving and sharing visualization designs.

With Vega you can describe data visualizations in a JSON format, and generate interactive views using either HTML5 Canvas or SVG.

Read the [tutorial](#), browse the [documentation](#), join the [discussion](#), and explore visualizations using the web-based [Vega Editor](#).



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

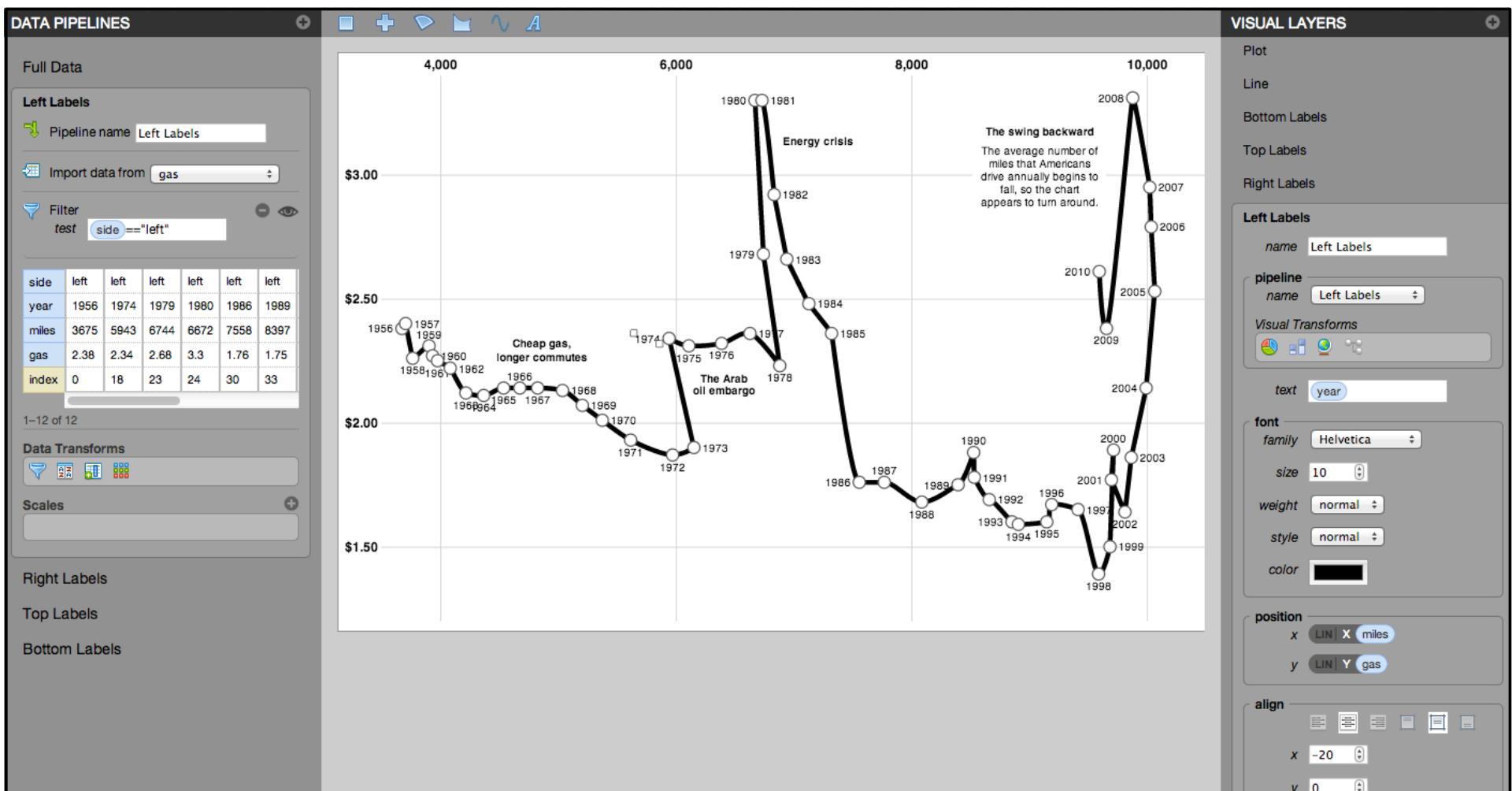


William Playfair's classic chart comparing the price of wheat and wages in England recreated in the Lyra VDE.

ABSTRACT

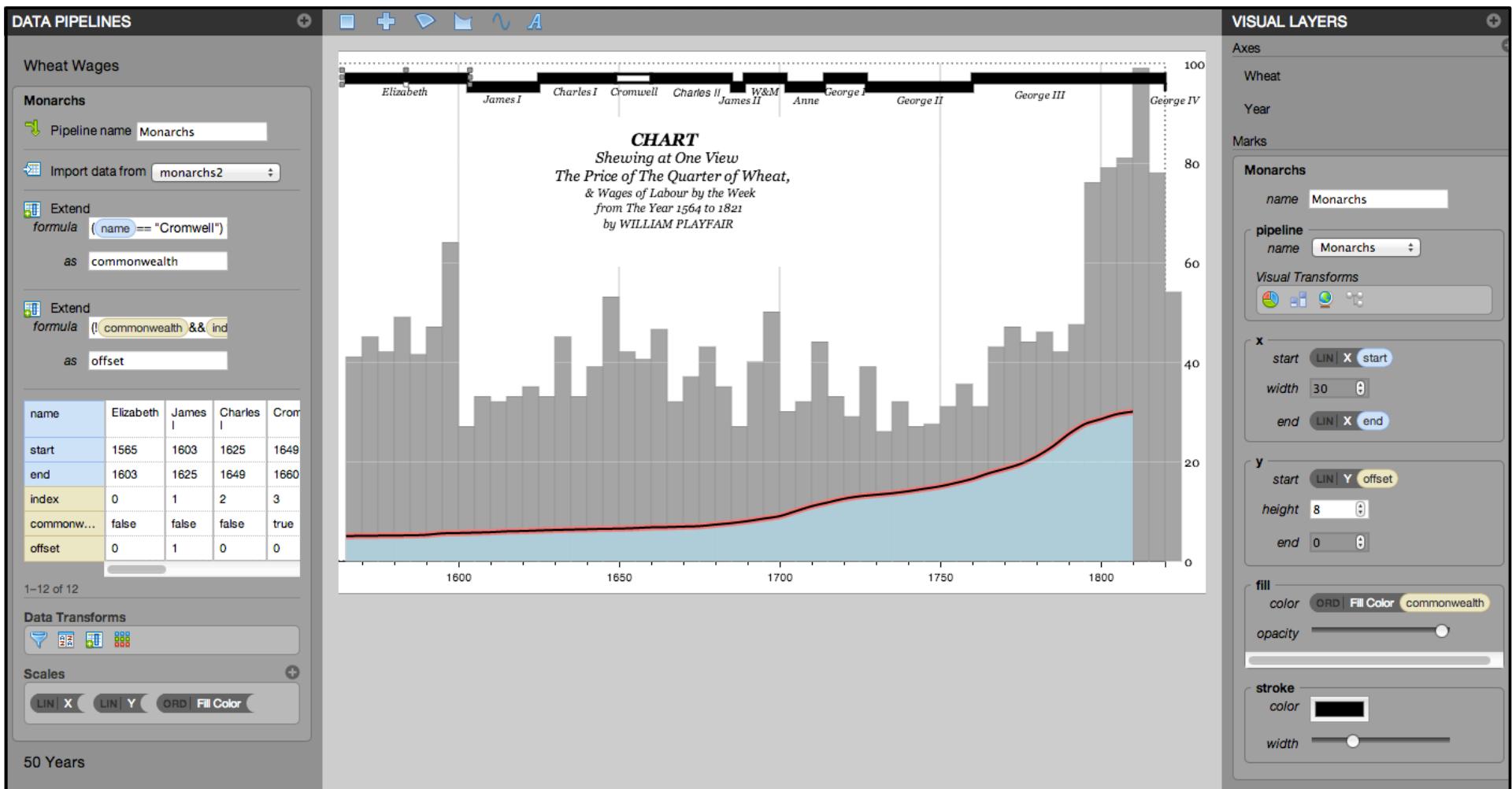
Lyra is an interactive environment that enables custom visualization design without writing any code. Graphical "marks" can be bound to data fields using property drop zones; dynamically positioned using connectors; and directly moved, rotated, and resized using handles. Lyra also provides a data pipeline 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 A Visualization Design Environment



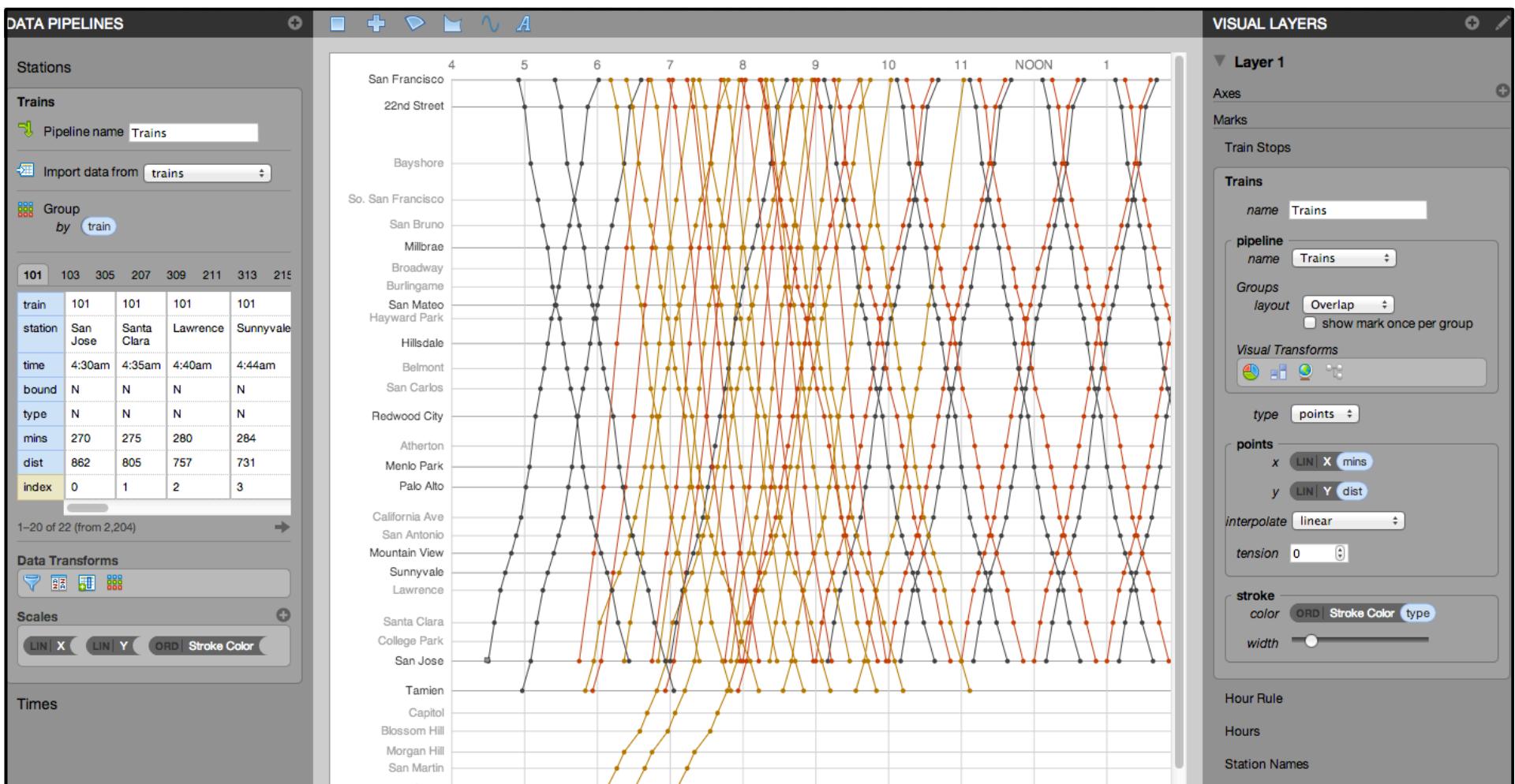
Driving Shifts into Reverse by Hannah Fairfield, NYTimes

Lyra A Visualization Design Environment



by William Playfair

Lyra A Visualization Design Environment



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

Lyra

A diagram illustrating the stack of data visualization libraries. It consists of four horizontal bars of increasing height from left to right. The first bar is light gray and labeled "JavaScript". The second bar is medium gray and labeled "SVG". The third bar is white and labeled "D3.js". The fourth bar is teal and labeled "Vega". A vertical line connects the "D3.js" and "Vega" bars. A dashed horizontal line connects the top of the "D3.js" bar to the top of the "Vega" bar.

Vega

D3.js

JavaScript

SVG

Canvas

```
graph TD; JS[JavaScript] --- SVG[SVG]; JS --- CANVAS[Canvas]; Vega[Vega] --- VEGALITE[Vegalite]; VEGALITE --- LYRA[Lyra]
```

Lyra

Vegalite

Vega

D3.js

JavaScript

SVG

Canvas

Vegalite

A formal model for statistical graphics

Inspired by *Grammar of Graphics* & *Tableau*

Includes **data transformation & encoding**

Vegalite

A formal model for statistical graphics

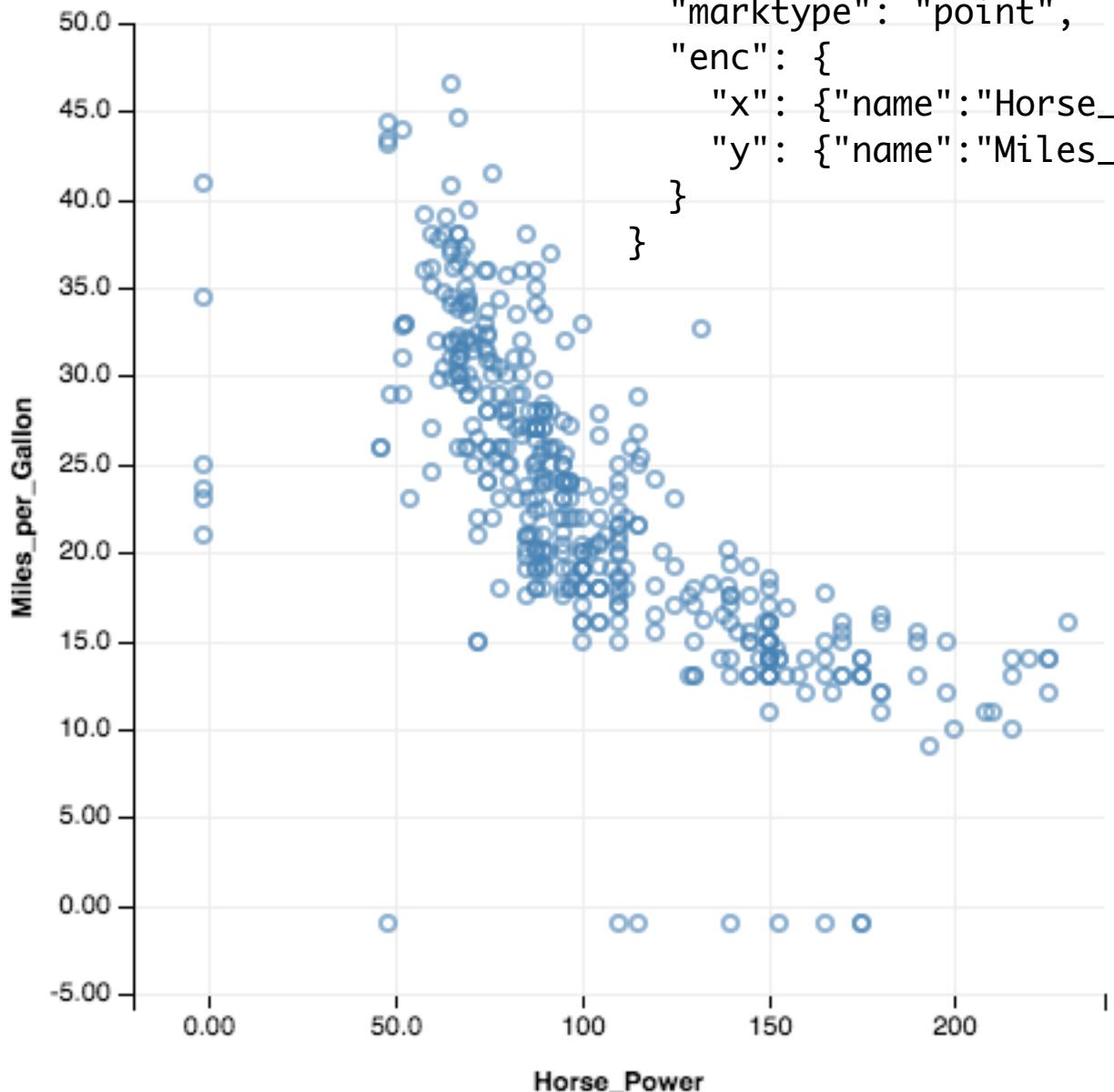
Inspired by *Grammar of Graphics* & *Tableau*

Includes **data transformation & encoding**

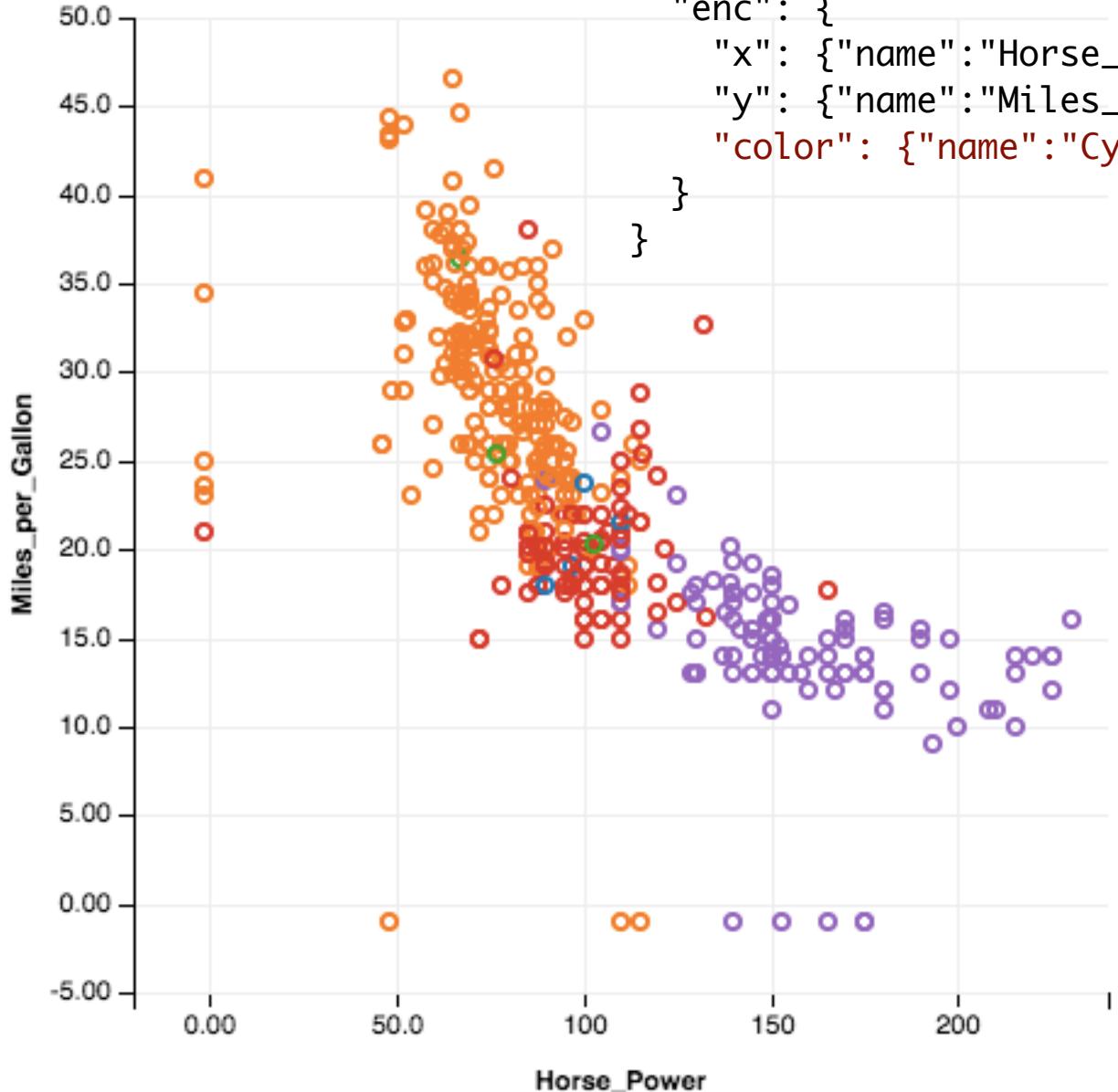
Uses a simple, concise **JSON format** that
compiles to full-blown **Vega specifications**

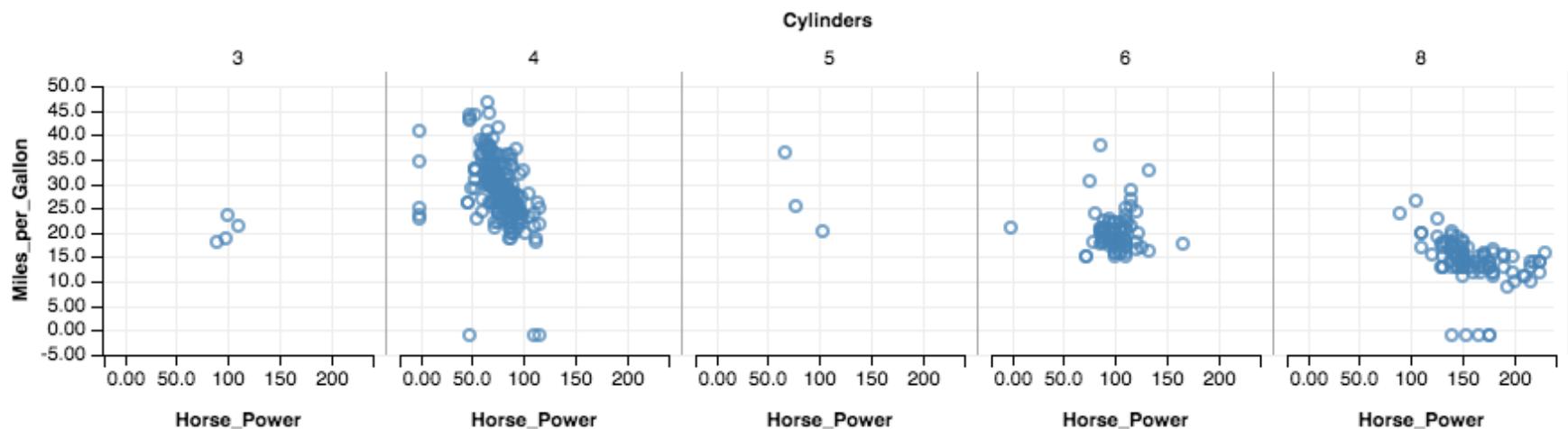
Easy **programmatic generation!**

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



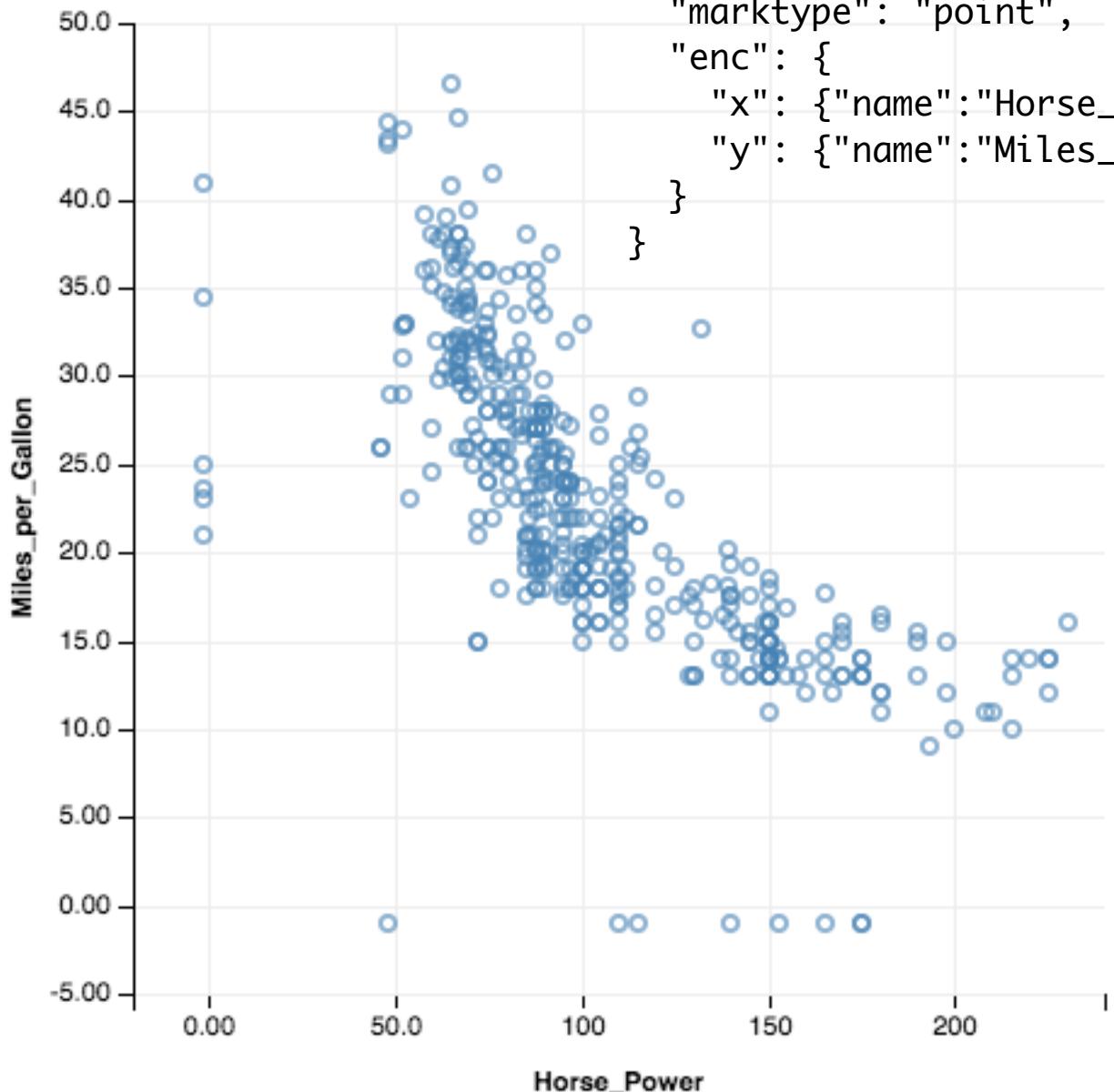
```
{  
  "marktype": "point",  
  "enc": {  
    "x": {"name": "Horse_Power", "type": "Q"},  
    "y": {"name": "Miles_per_Gallon", "type": "Q"},  
    "color": {"name": "Cylinders", "type": "O"}  
  }  
}
```

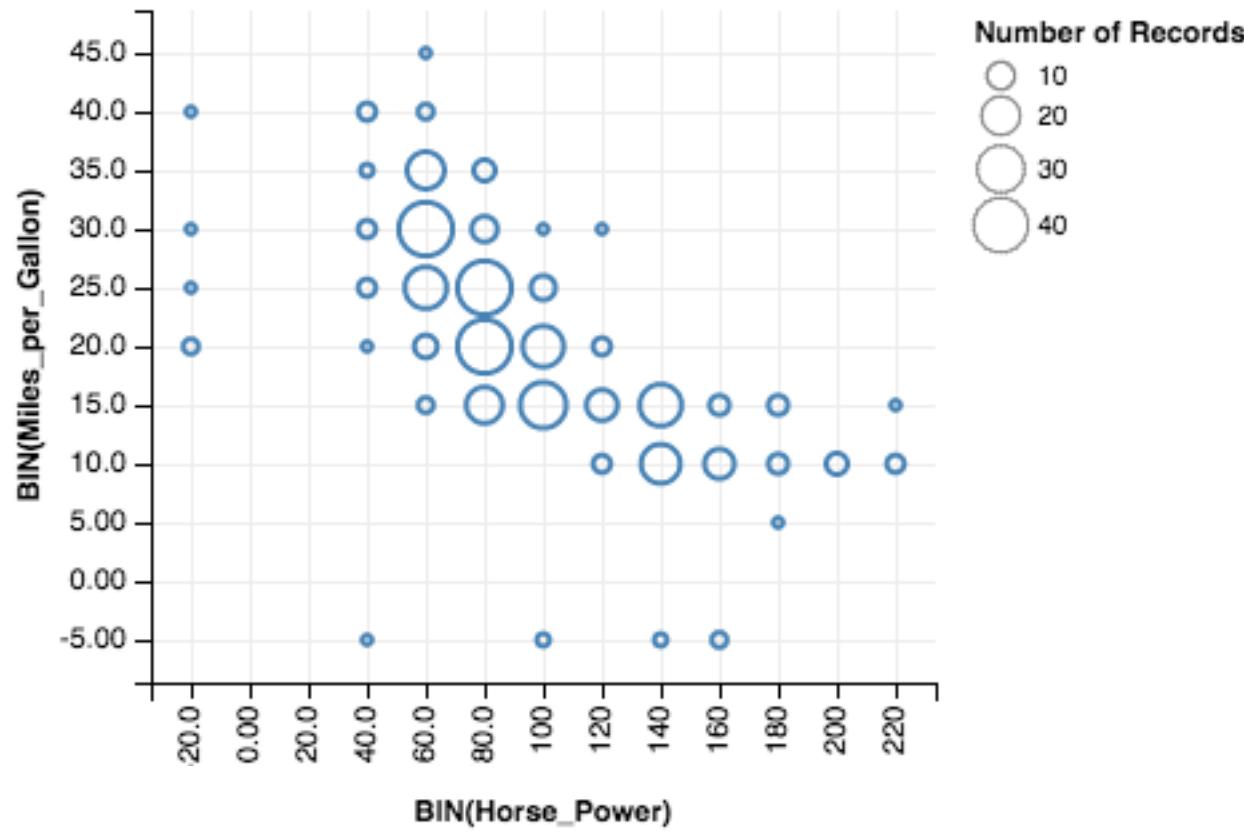




```
{  
  "marktype": "point",  
  "enc": {  
    "x": {"name": "Horse_Power", "type": "Q"},  
    "y": {"name": "Miles_per_Gallon", "type": "Q"},  
    "col": {"name": "Cylinders", "type": "O"}  
  }  
}
```

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





```
{  
  "marktype": "point",  
  "enc": {  
    "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"}  
  }  
}
```

```
graph TD; JS[JavaScript] --- D3[D3.js]; D3 --- SVG[SVG]; SVG --- Lyra[Lyra]; Vg[Vega] --- VL[Vegalite]; VL --- Lyra;
```

Lyra

Vegalite

Vega

D3.js

JavaScript

SVG

Canvas

Polestar

Lyra

Vegalite

Vega

D3.js

JavaScript

SVG

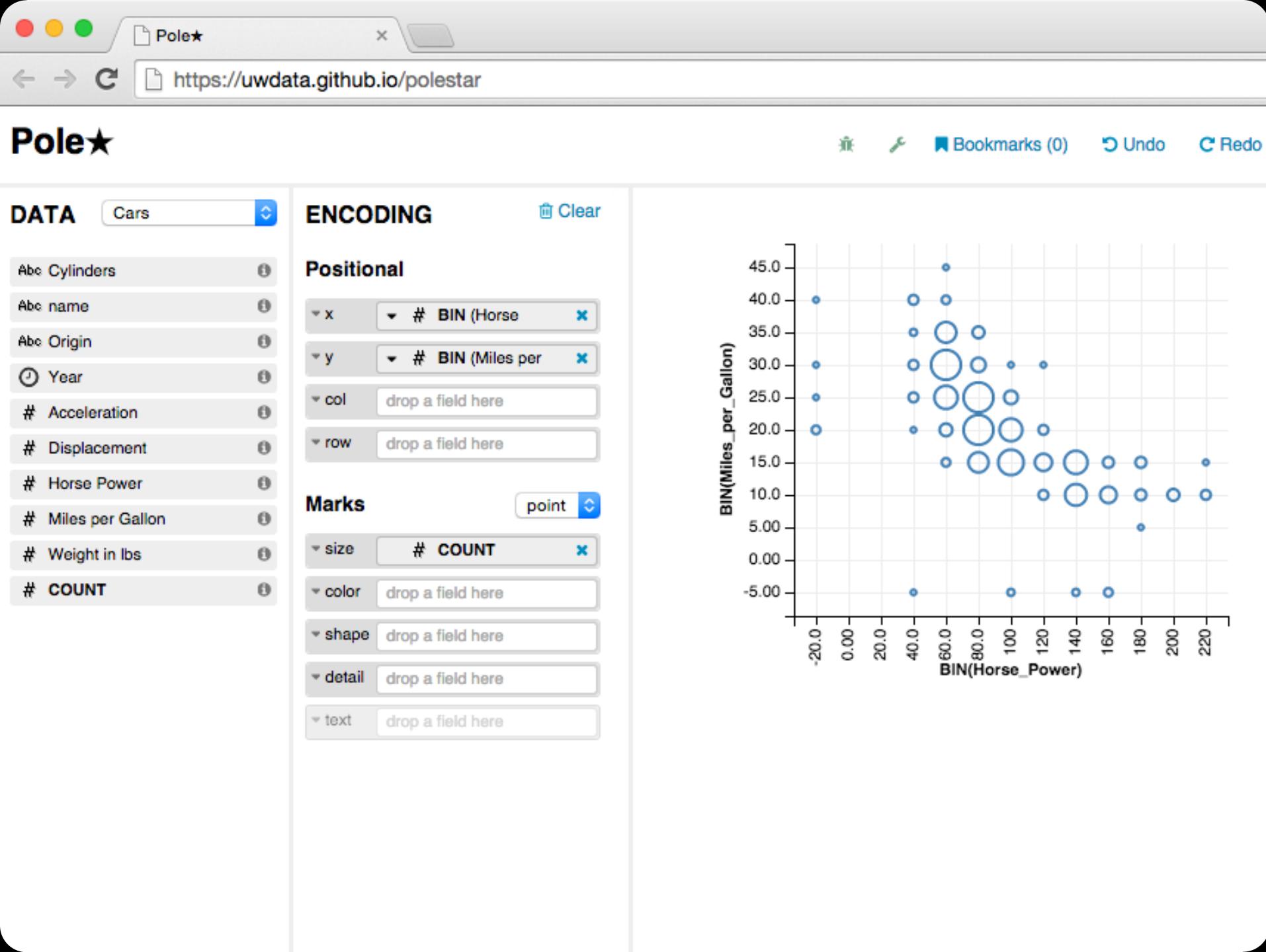
Canvas

Polestar

A graphical interface for **Vegalite**

Rapid visualization via drag-and-drop

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



Polestar

Lyra

Vegalite

Vega

D3.js

JavaScript

SVG

Canvas

Voyager

Polestar

Lyra

Vegalite

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*

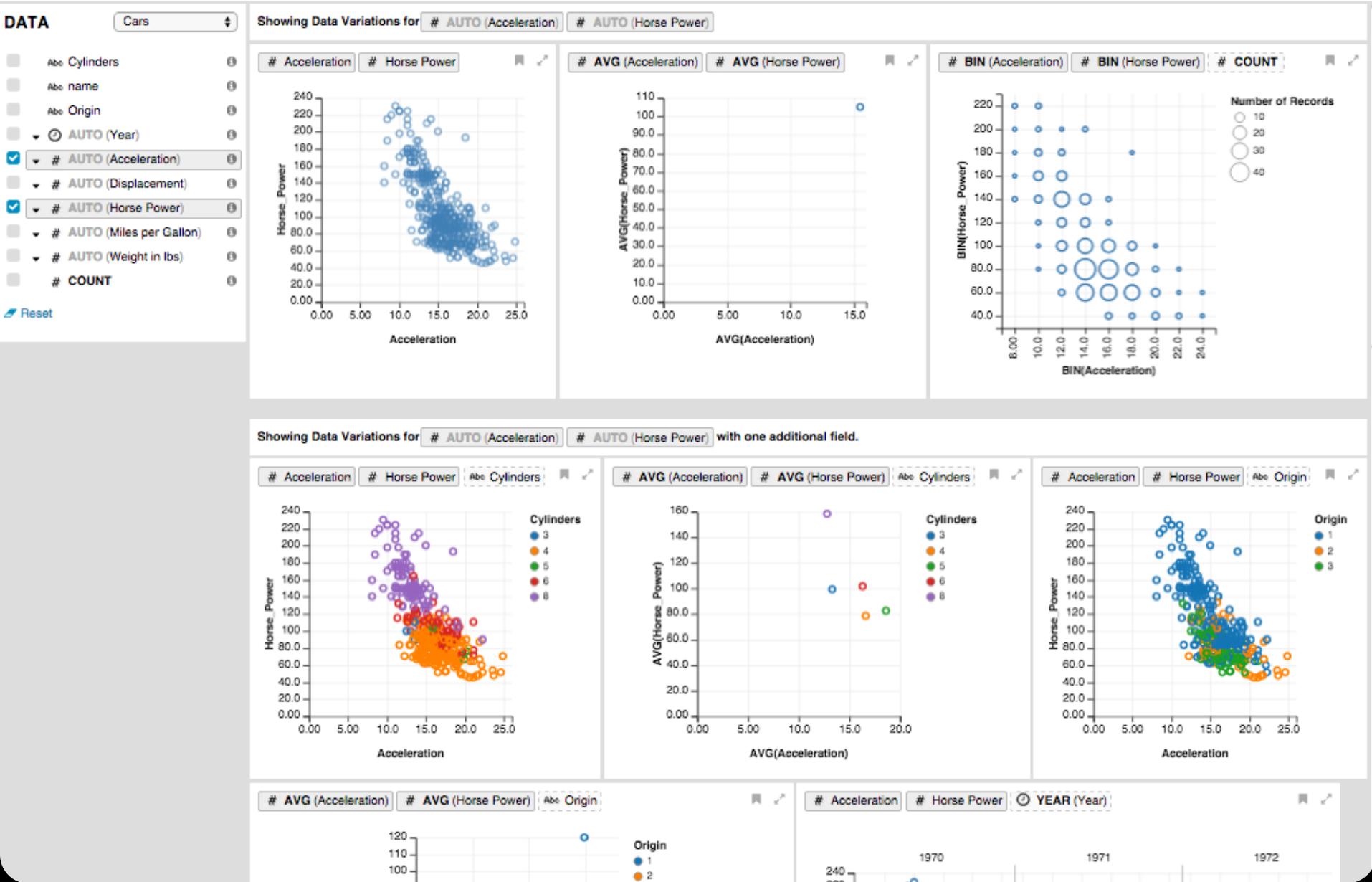
Approach: browse a gallery of visualizations

Challenge - *combinatorial explosion!*

Automatic recommendation of useful views

+ **end-user steering** to focus exploration

Data Voyager

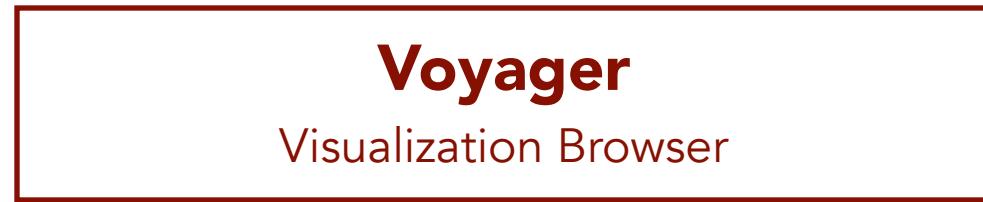




User



Data Set



User

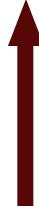


User

Data Schema
& Statistics

Compass

Recommendation Engine



Voyager

Visualization Browser

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



User

Compass
Recommendation Engine

Data Schema
& Statistics



Voyager
Visualization Browser

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



User

Compass

Recommendation Engine

Data Schema
& Statistics

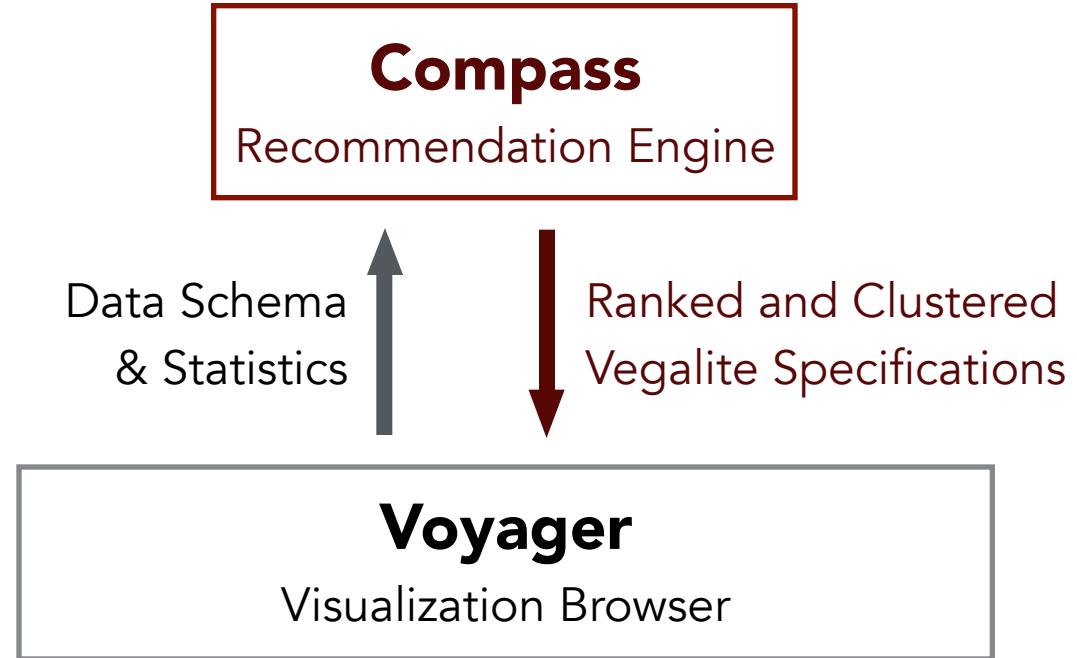


Voyager

Visualization Browser

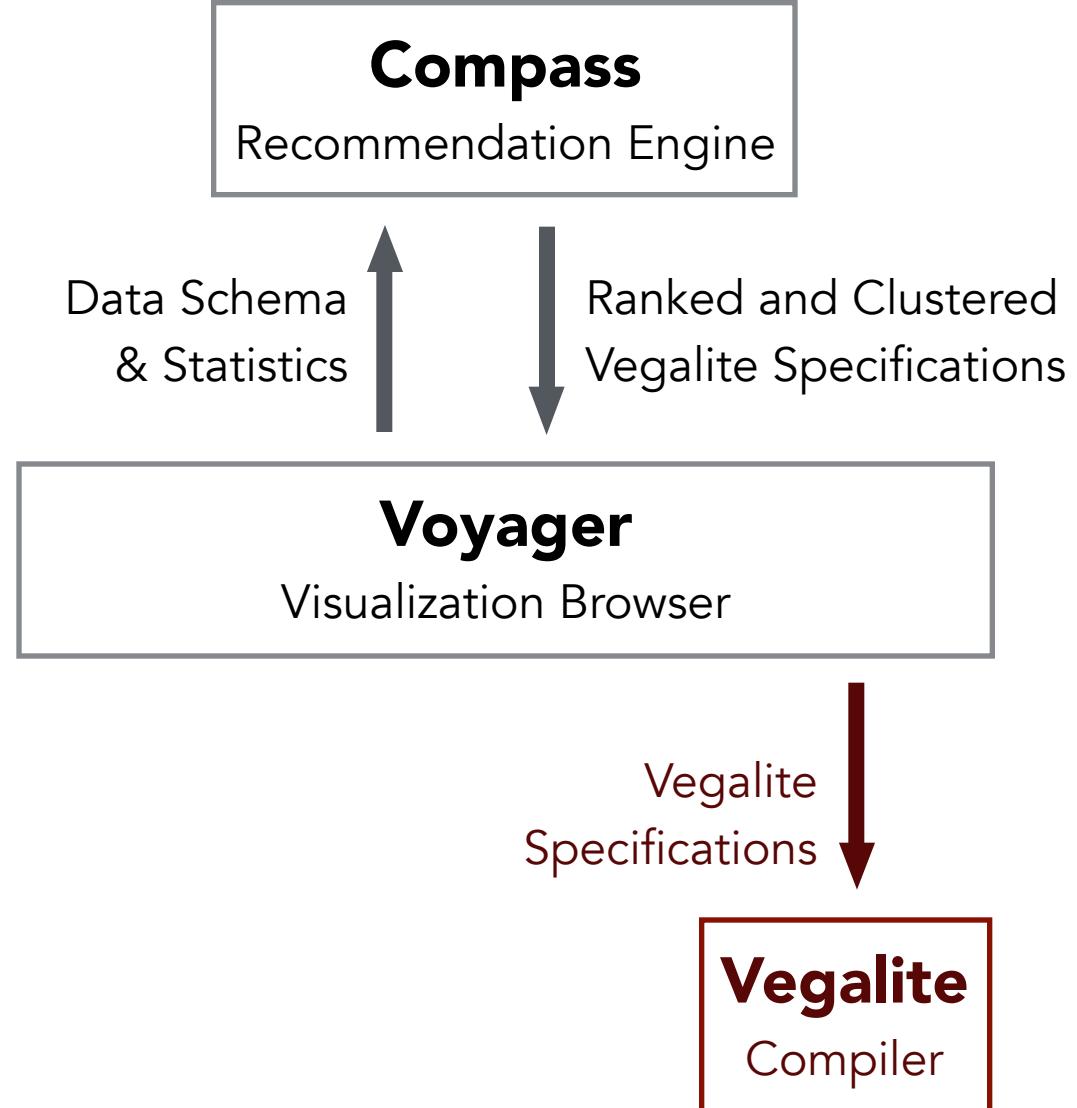


User





User



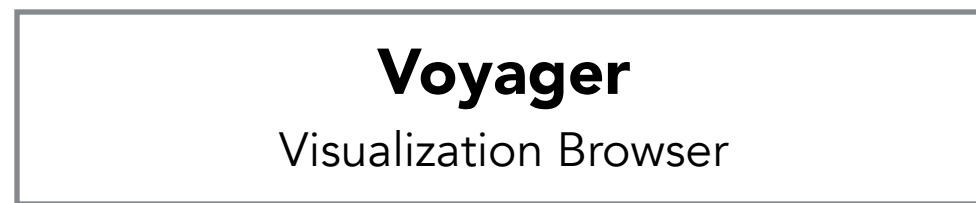


User



Data Schema
& Statistics

Ranked and Clustered
Vegalite Specifications



Vegalite
Specifications



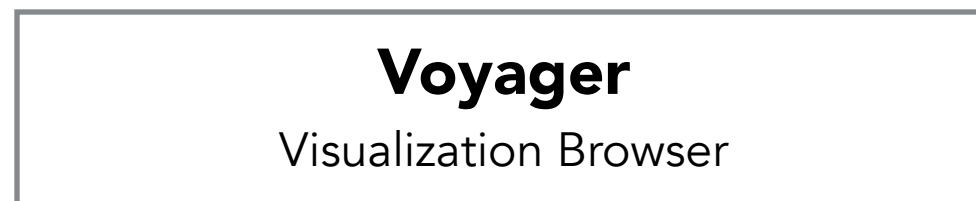


User



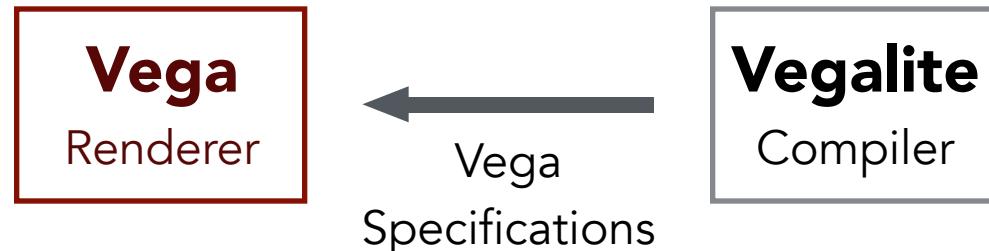
Data Schema
& Statistics

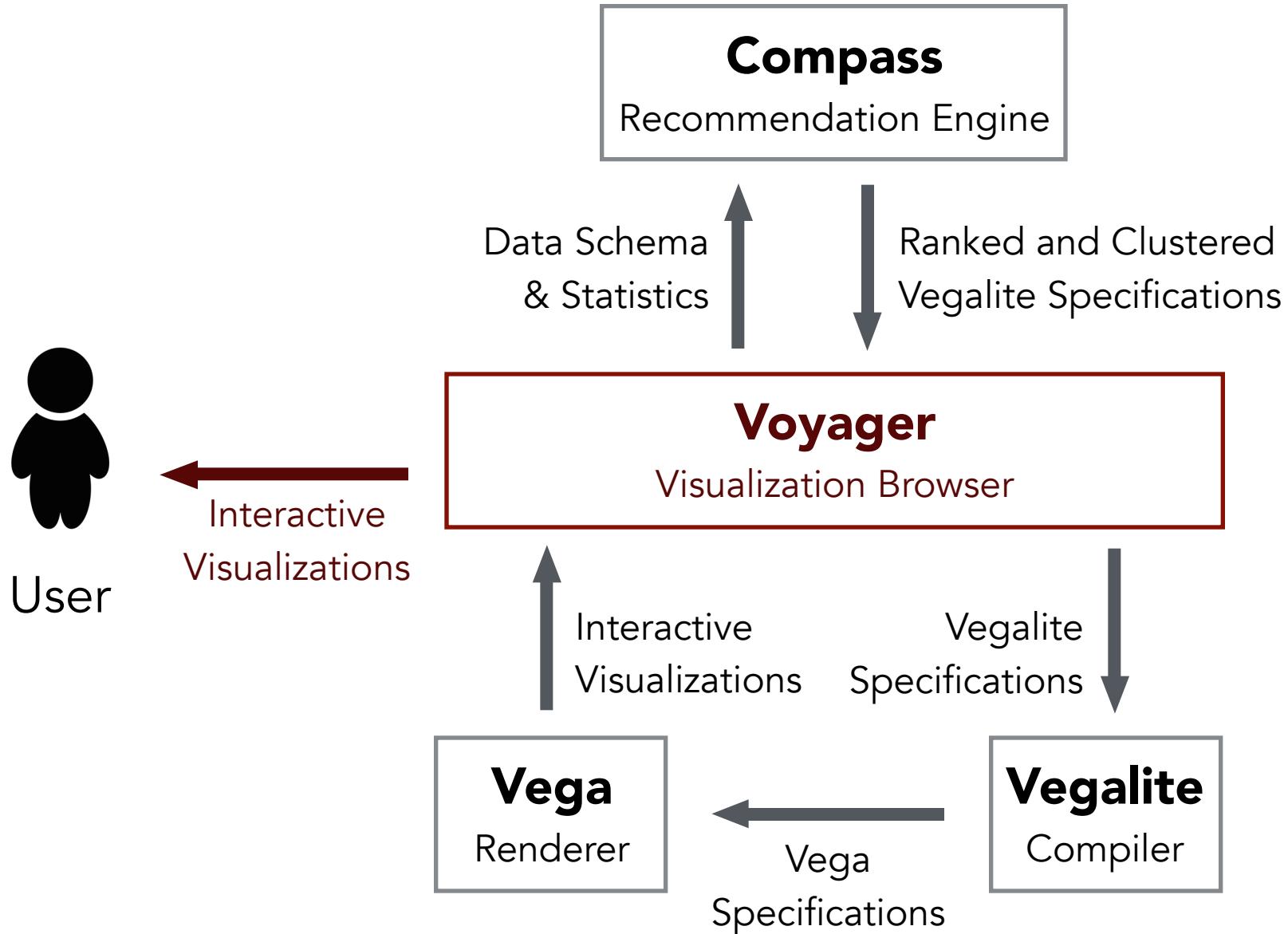
Ranked and Clustered
Vegalite Specifications

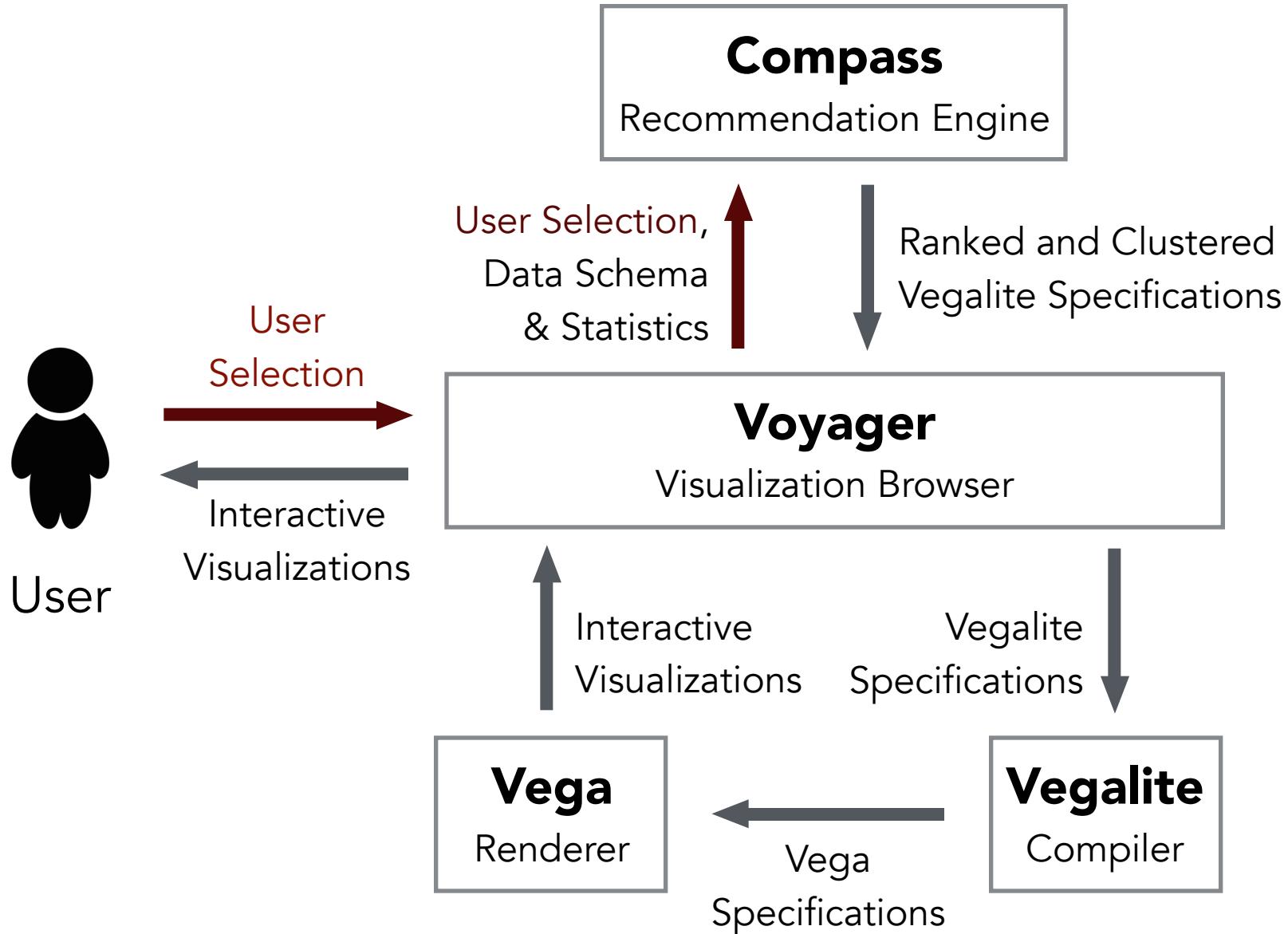


Interactive
Visualizations

Vegalite
Specifications







Improves data coverage!

+3x variable sets shown

+1.5x more interacted with



User

User Selection

Interactive

Visualizations

Compass

Recommendation Engine

User Selection,
Data Schema
& Statistics

Ranked and Clustered
Vegalite Specifications

Voyager

Visualization Browser

Interactive
Visualizations

Vegalite
Specifications

Vega

Renderer

Vega
Specifications

Vegalite

Compiler

Voyager

Polestar

Lyra

Vegalite

Vega

D3.js

JavaScript

SVG

Canvas

Voyager

Polestar

Vegalite

Vega

JavaScript

SVG

Canvas

D3.js

Lyra

One last thing...

What about
interaction?

Voyager

Polestar

Lyra

Vegalite

Vega

D3.js

JavaScript

SVG

Canvas

Voyager

Polestar

Lyra

Vegalite

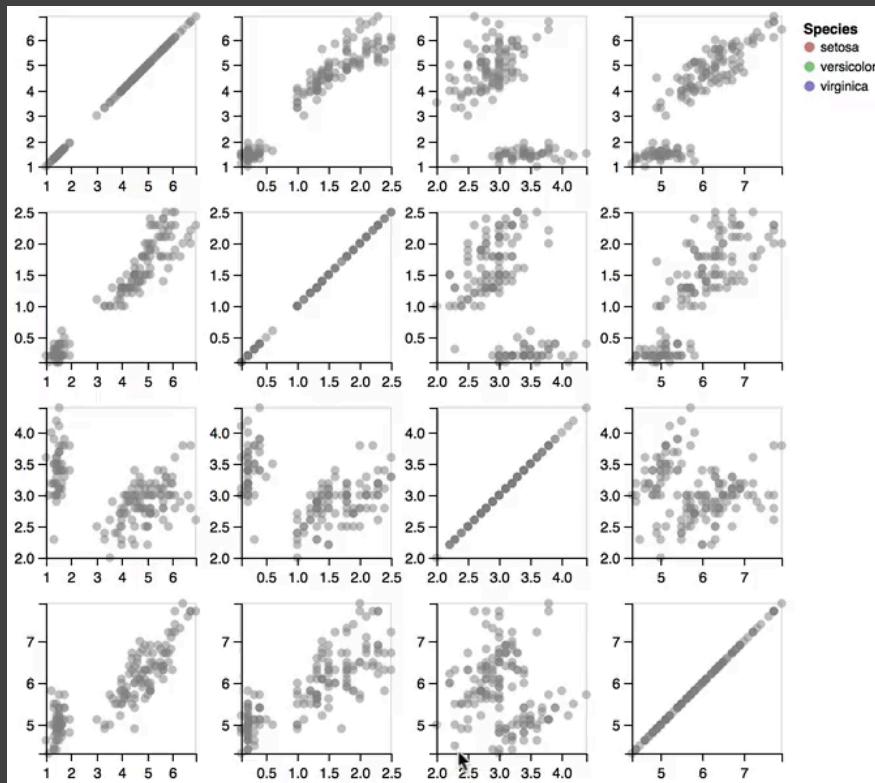
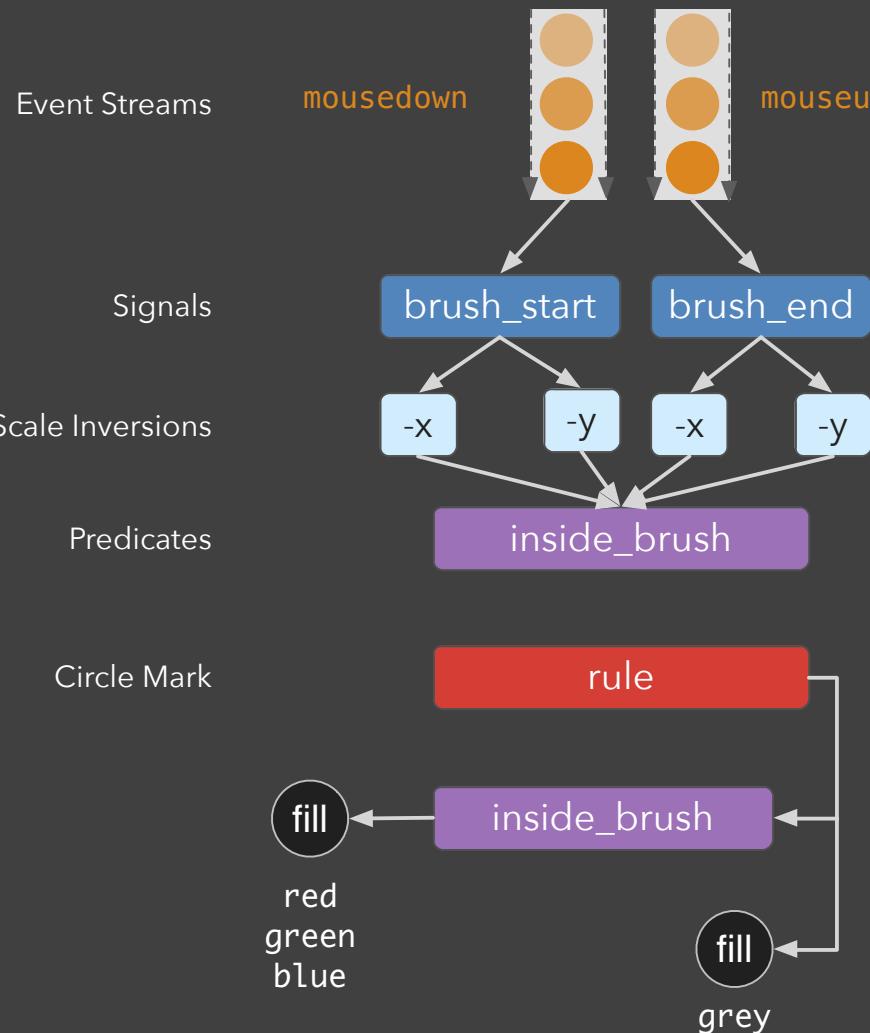
Vega

D3.js

JavaScript

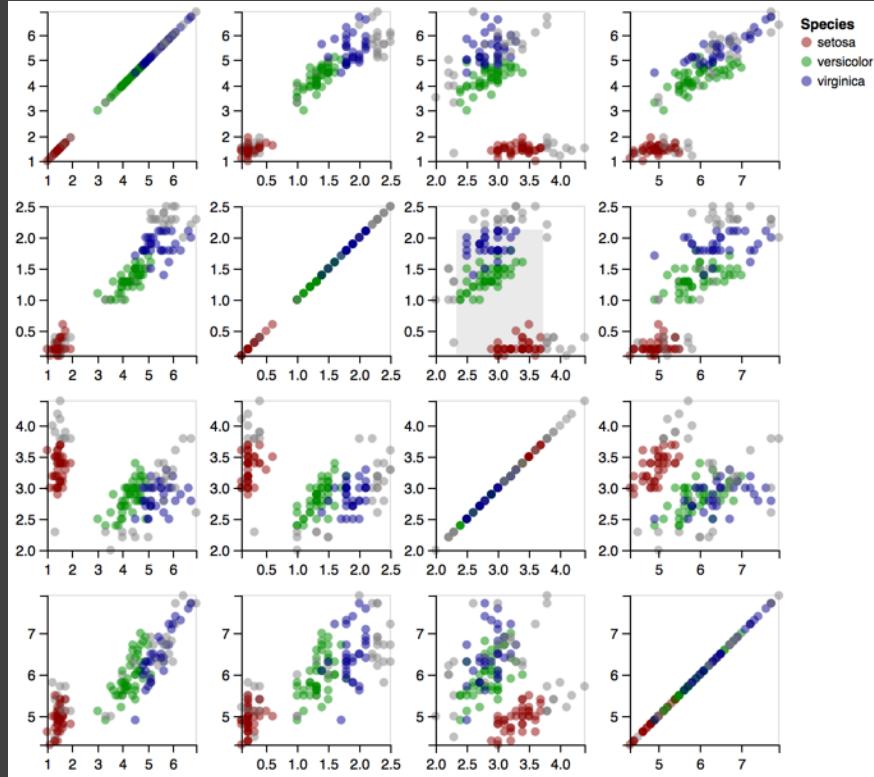
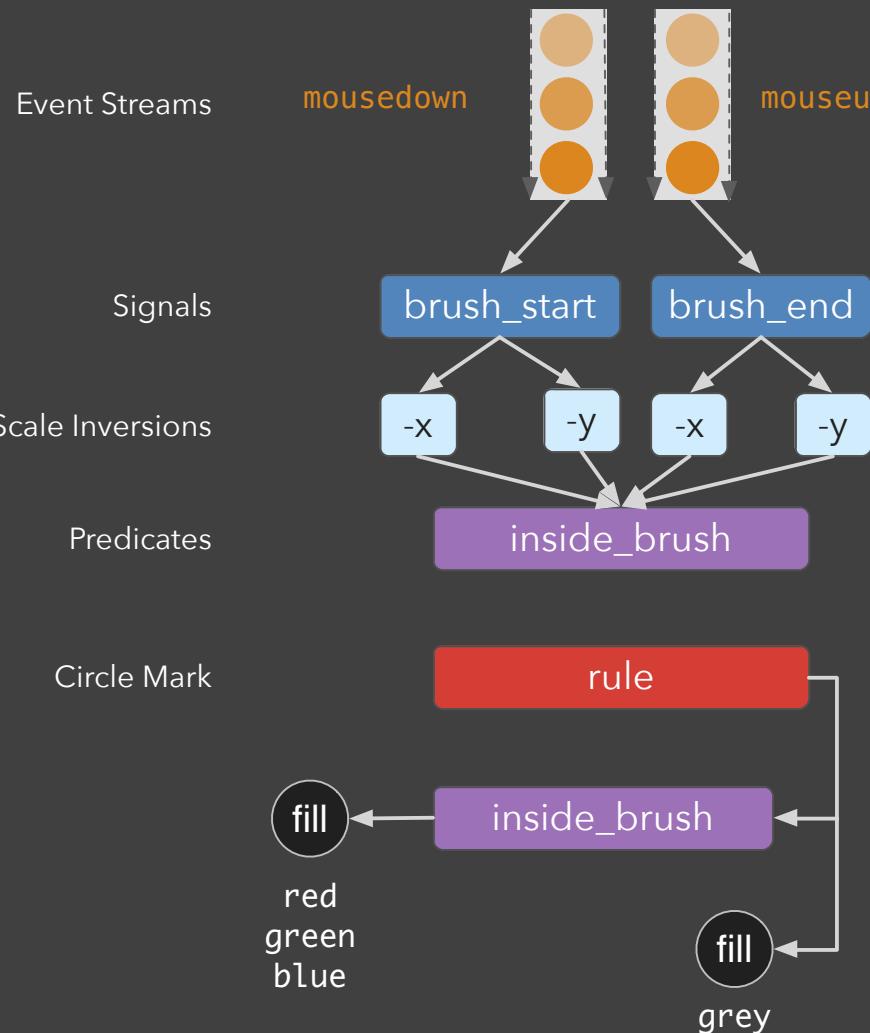
SVG

Canvas



Reactive Vega

Satyanarayan et al. [UIST'14]



Key Insight: Treat user input as first-class streaming data
 Adapt methods from functional reactive programming

Vega 2.0 (“Reactive Vega”)

Single declarative model for specifying
visual encodings + interaction techniques

Vega 2.0 ("Reactive Vega")

Single declarative model for specifying
visual encodings + interaction techniques

JSON → Reactive Dataflow Graph

Designed ground-up for **streaming data**

Performance matches or exceeds D3

Vega 2.0 (“Reactive Vega”)

Single declarative model for specifying
visual encodings + interaction techniques

JSON → Reactive Dataflow Graph

Designed ground-up for **streaming data**

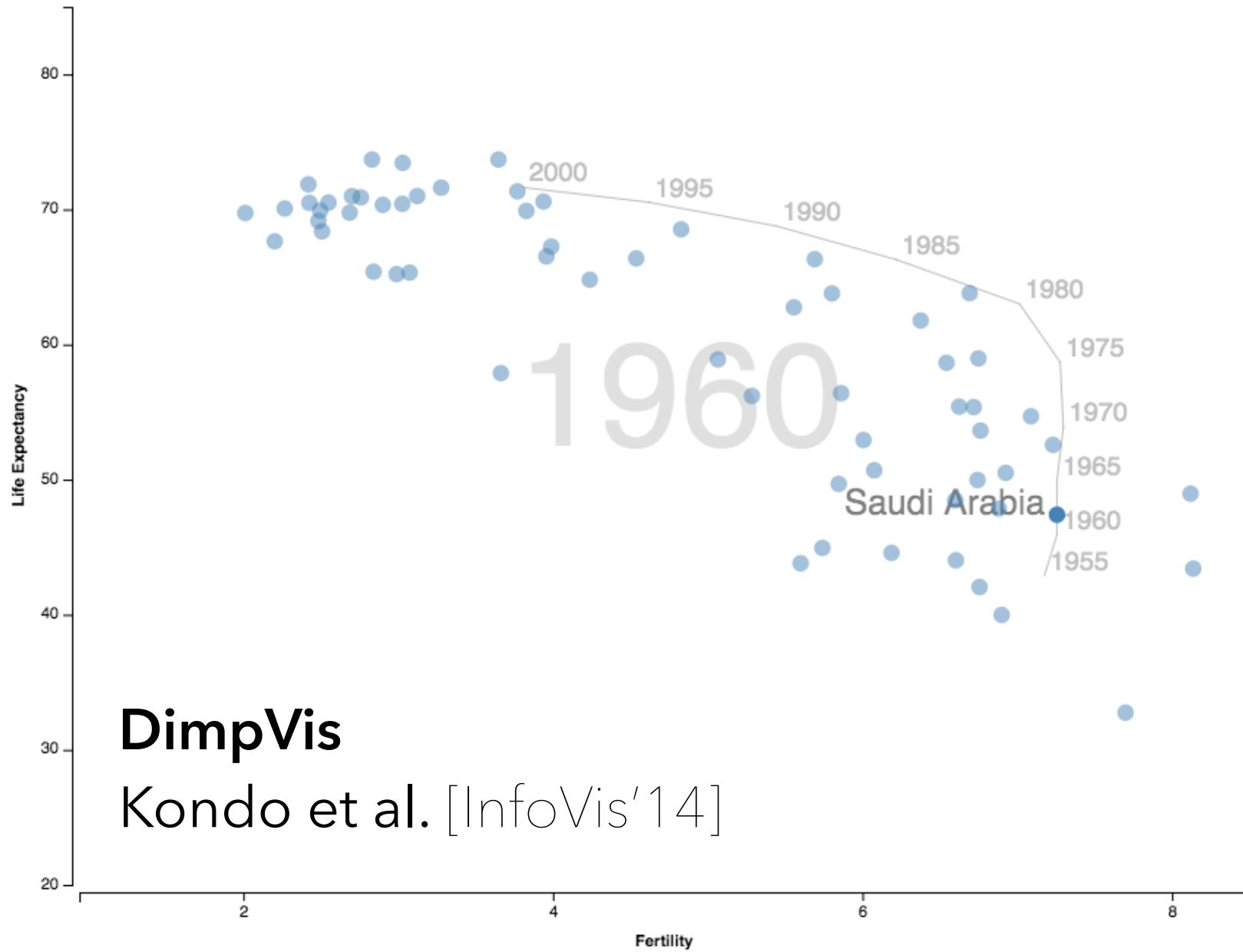
Performance matches or exceeds D3

Portable

Client (*browser*) or server-side (*node.js*)

Pick your renderer: *Canvas, SVG, ...*

Pick your input: *mouse, touch, ...*



Voyager

Polestar

Lyra

Vegalite

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

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, debugging, debugging...

All Open Source...

Vega A VISUALIZATION GRAMMAR



Vega is a declarative format for creating, saving, and sharing visualization designs. With Vega, visualizations are described in JSON, and generate interactive views using either HTML5 Canvas or SVG.

TOOLKITS

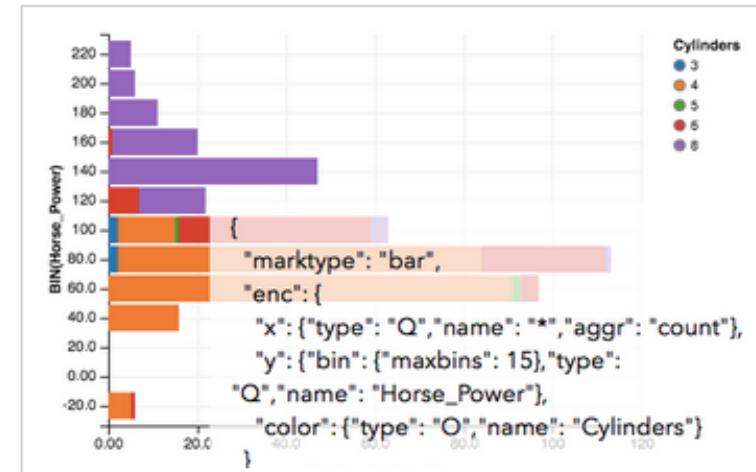


VEGA offers a full declarative visualization grammar, suitable for expressive custom interactive visualization design and programmatic generation.

[Tutorial](#) | [Documentation](#) | [Discussion Forum](#)

v1.5 (stable): [download](#), [examples](#), [github](#)

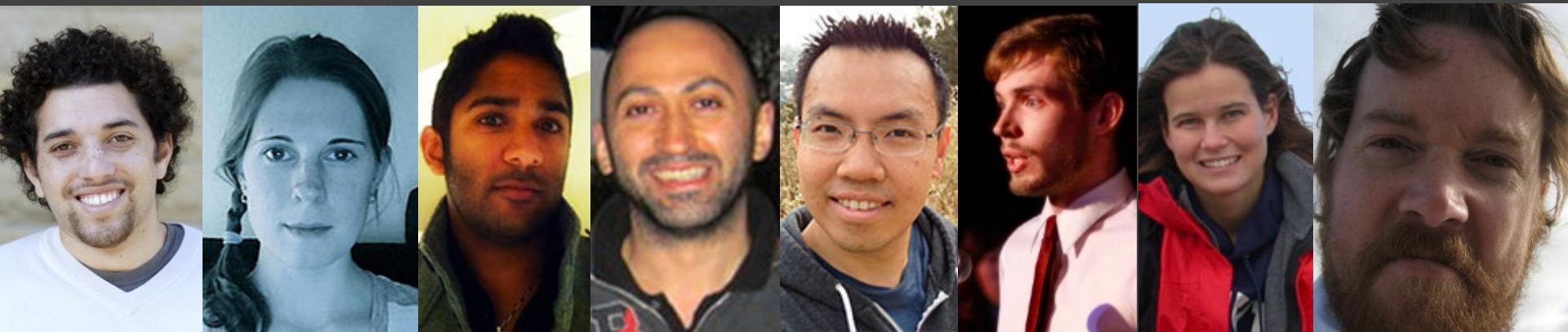
NEW v2.0 (dev): [download](#), [examples](#), [github](#)



NEW VEGALITE provides a higher-level grammar for visual analysis, comparable to ggplot or Tableau, that generates complete Vega specifications.

[Online Editor](#) | [GitHub](#)

vega.github.io



ISTC
BIG DATA



GORDON AND BETTY
MOORE
FOUNDATION



ISTC
BIG DATA



GORDON AND BETTY
MOORE
FOUNDATION

Raising the Bar (Chart)

THE NEXT GENERATION OF VISUALIZATION TOOLS

Jeffrey Heer @jeffrey_heer

<http://vega.github.io/>

