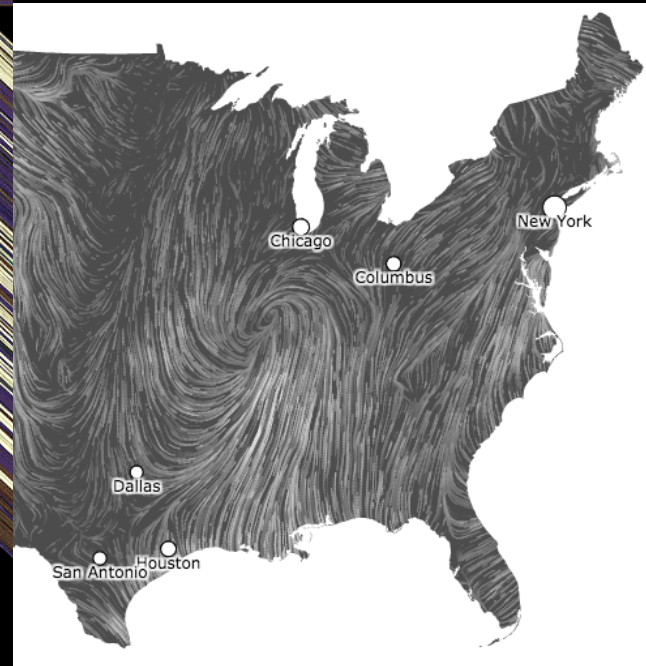
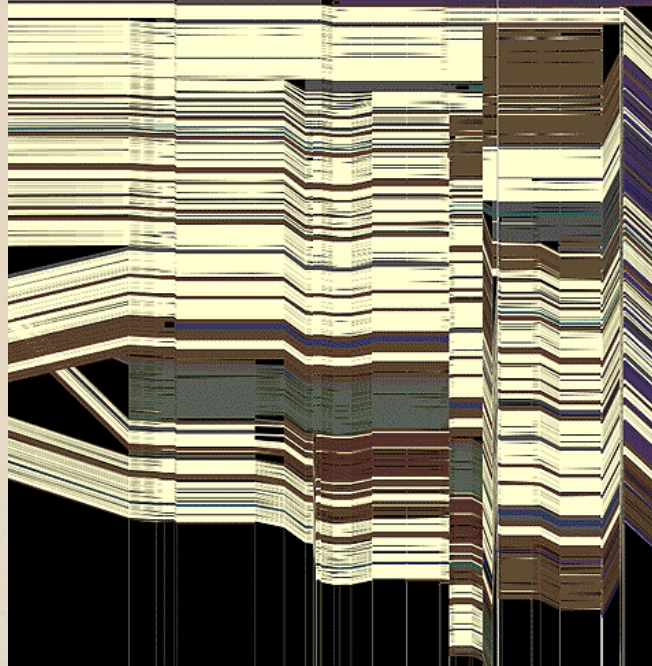
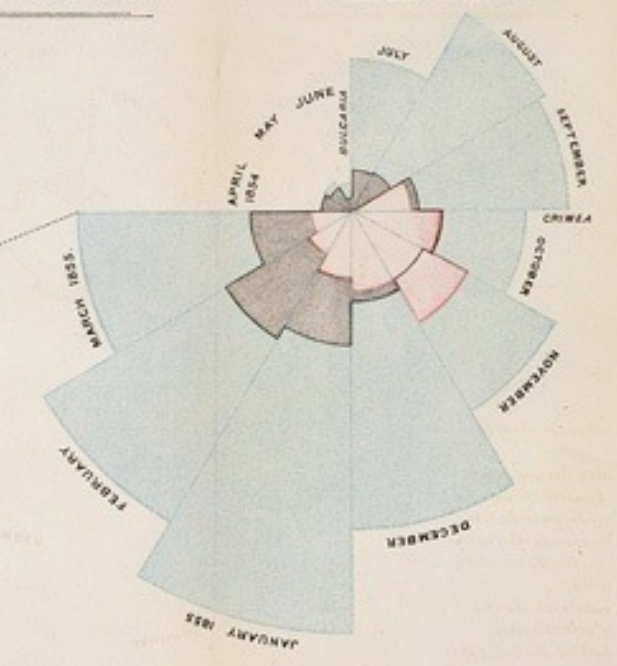


CSE 412 - Intro to Data Visualization

Visualization Tools



Jane Hoffswell University of Washington

How do people create visualizations?

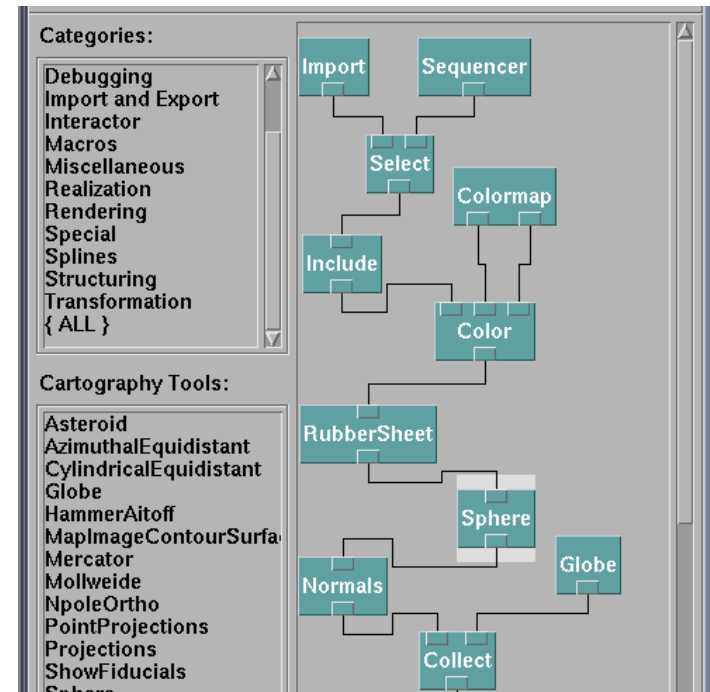


Chart Typology

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



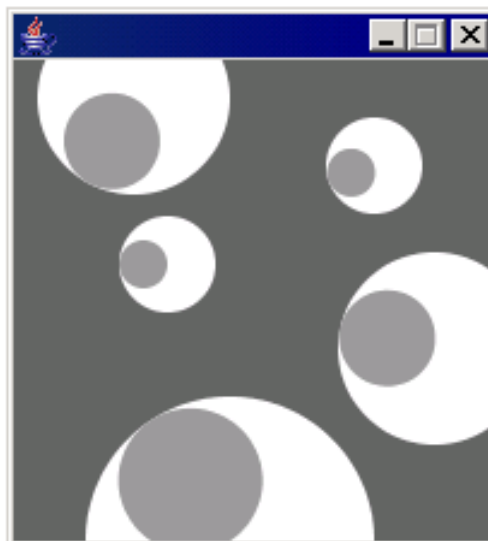
Graphics APIs

Processing, OpenGL, Java2D



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();  
  }  
}
```





US Air Traffic, Aaron Koblin

Graphics APIs

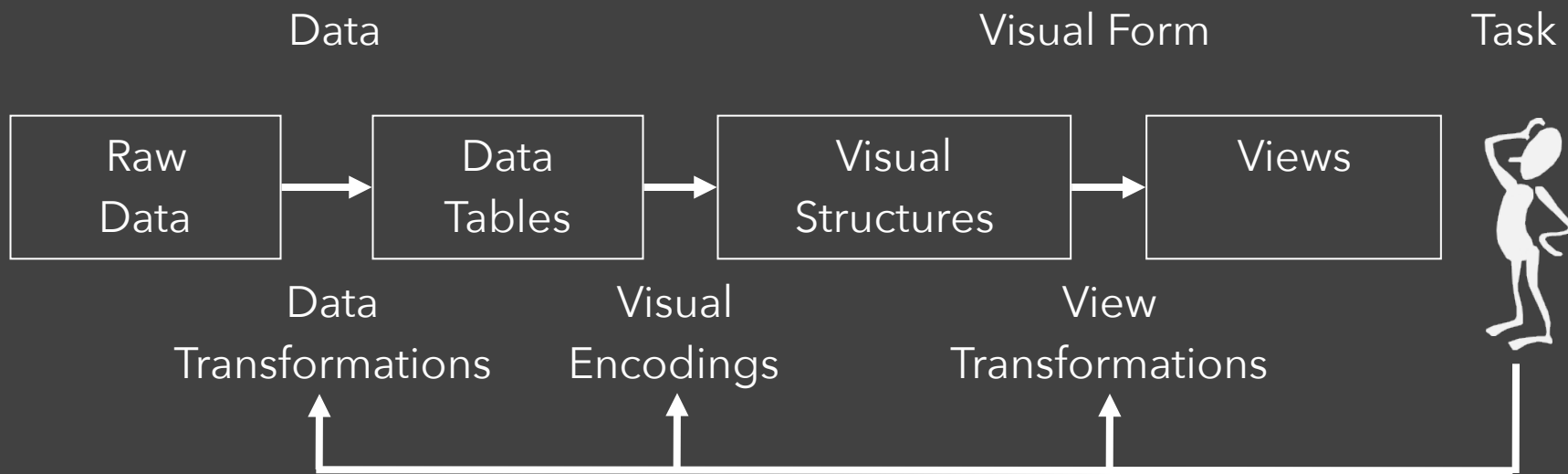
Processing, OpenGL, Java2D

Component Architectures

Prefuse, Flare, Improvise, VTK

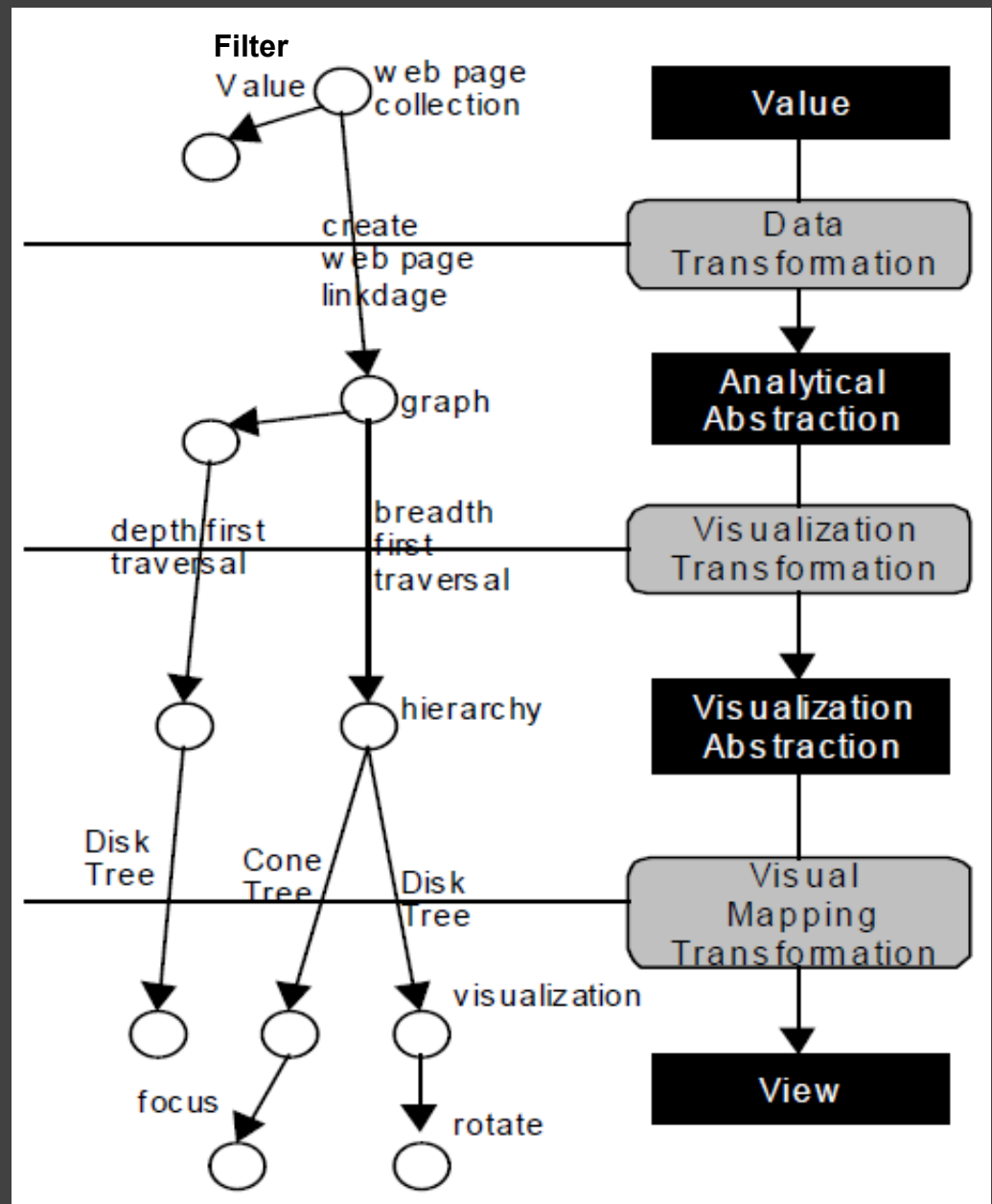
Graphics APIs

Processing, OpenGL, Java2D



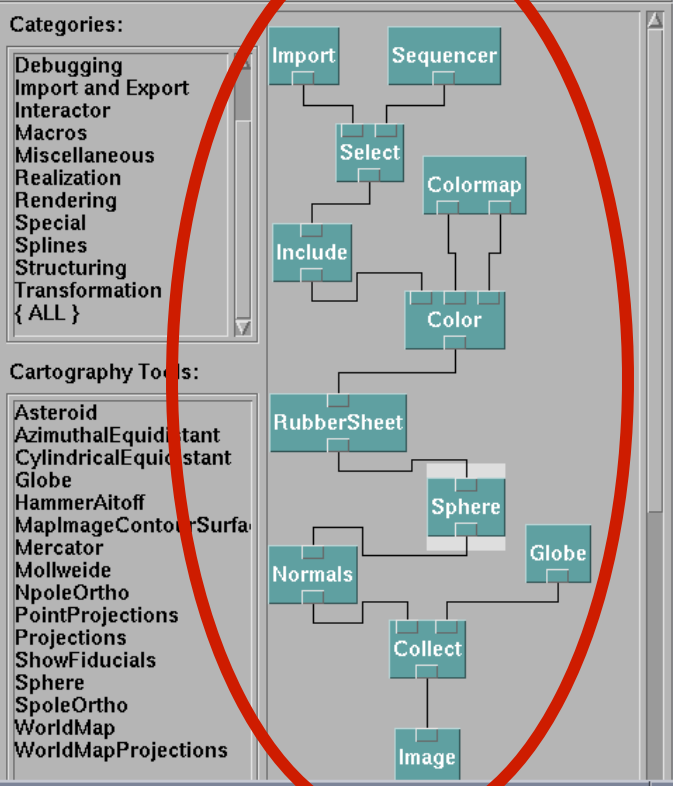
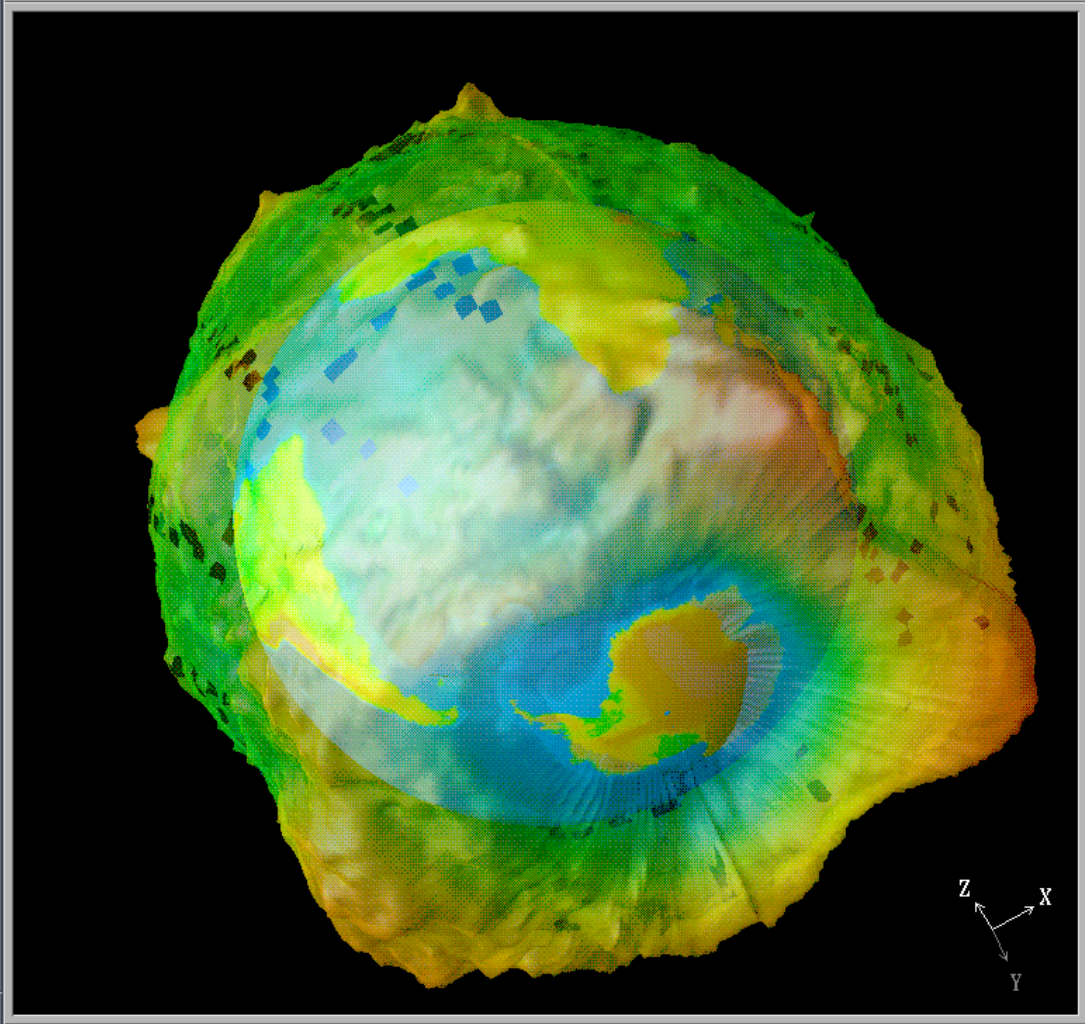
Data State Model

[Chi 98]



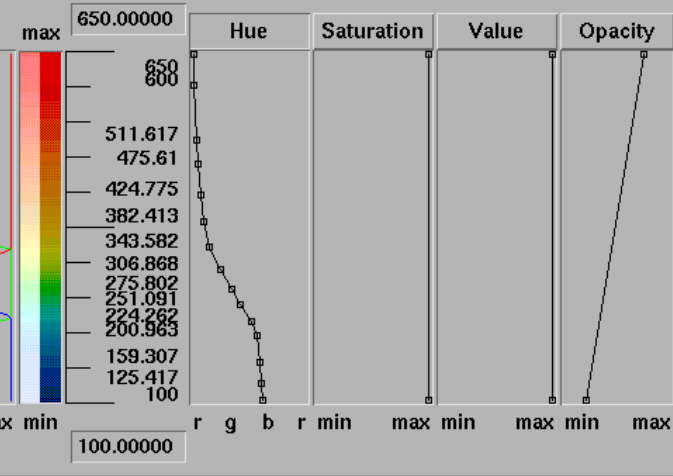
File Execute Windows Connection Options Help

File Edit Execute Windows Connection Options Help



Colormap Editor

File Execute Options Help



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

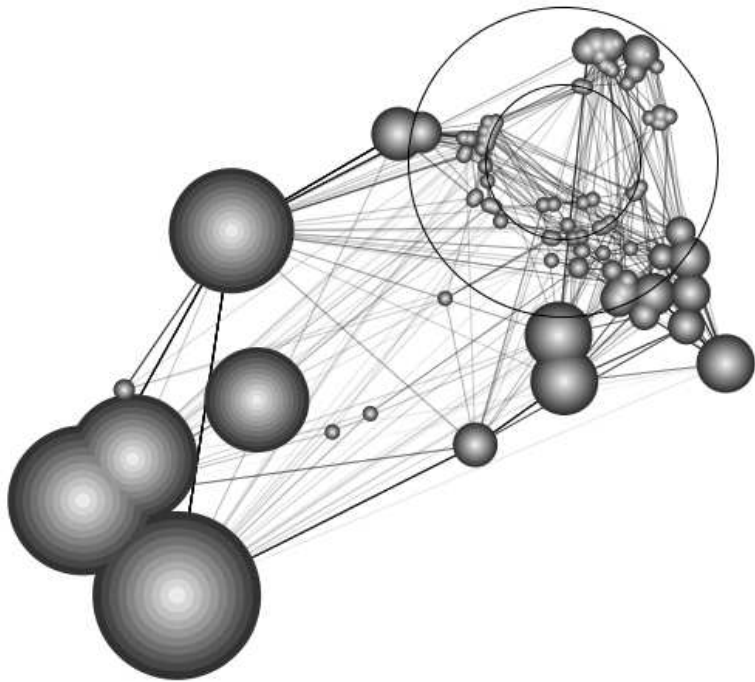
⏮ ⏪ ⏩ ⏭

⏮ ⏪ ⏩ ⏭

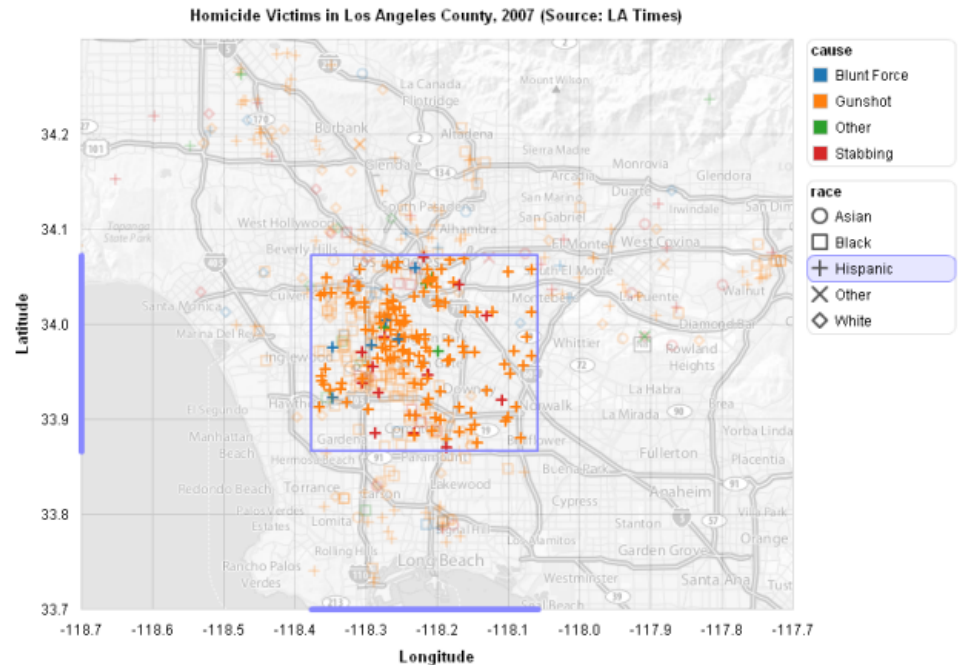
Prefuse & Flare

Operator-based toolkits for visualization design

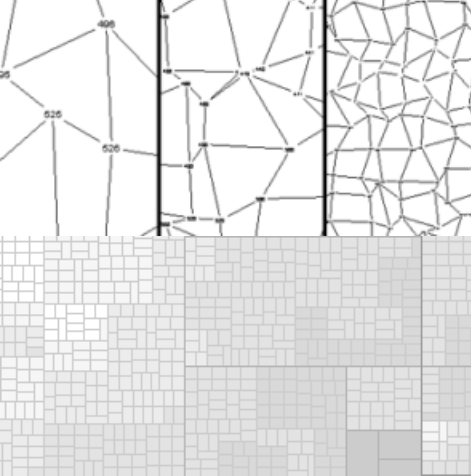
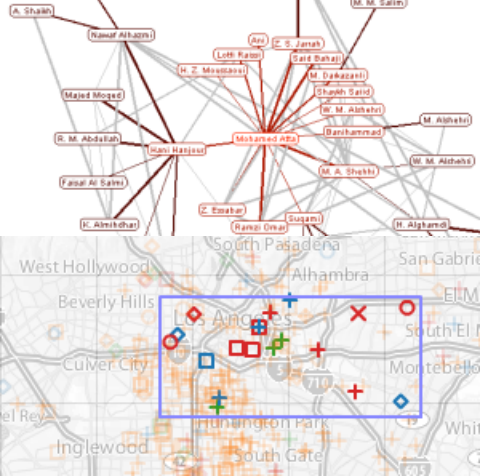
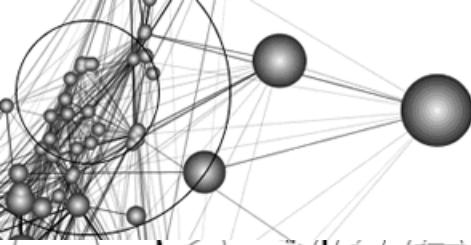
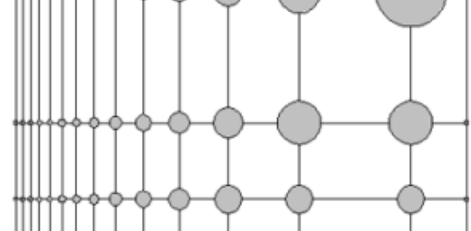
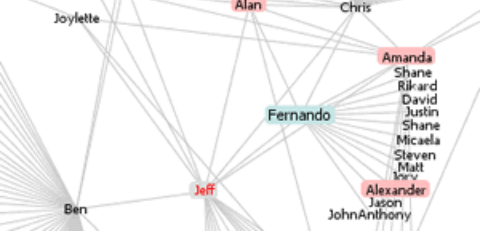
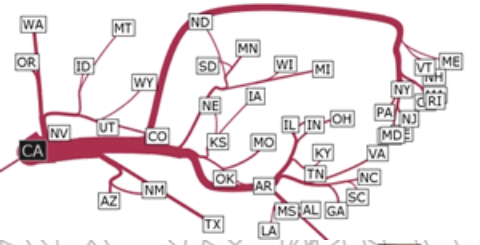
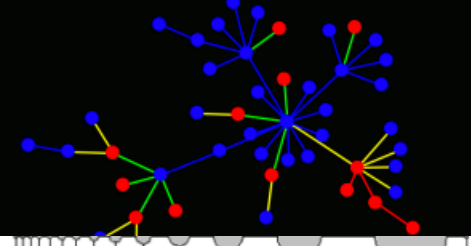
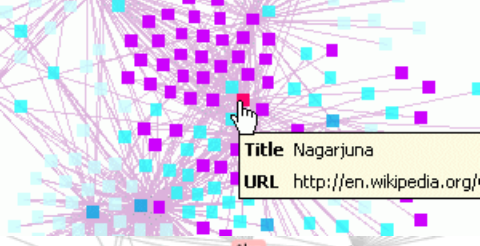
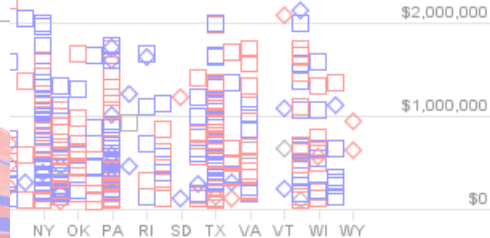
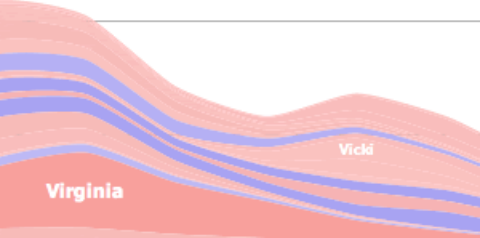
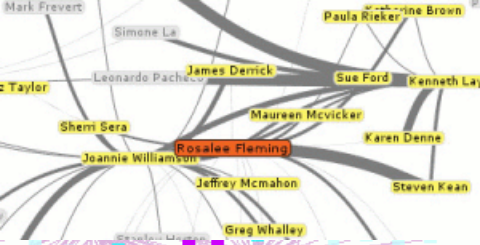
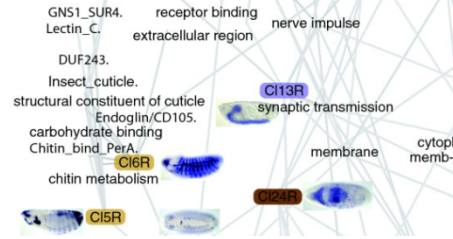
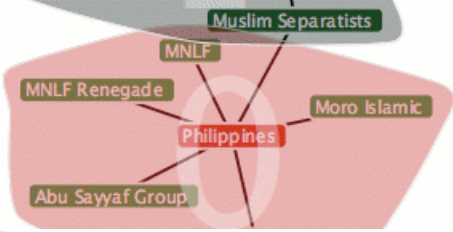
Vis = (Input Data -> Visual Objects) + Operators

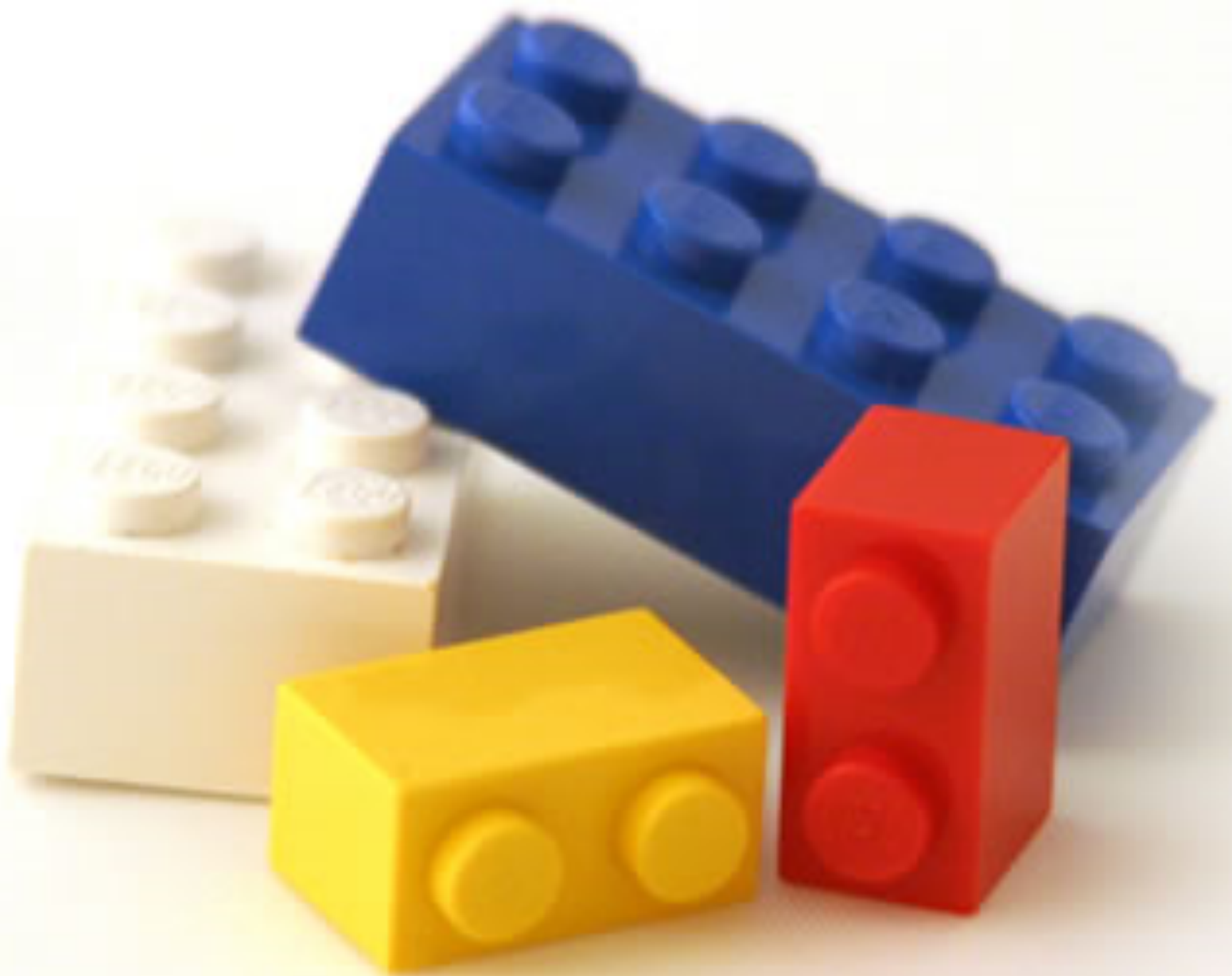


Prefuse (<http://prefuse.org>)



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







?

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](#)

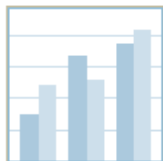


Word Tree

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

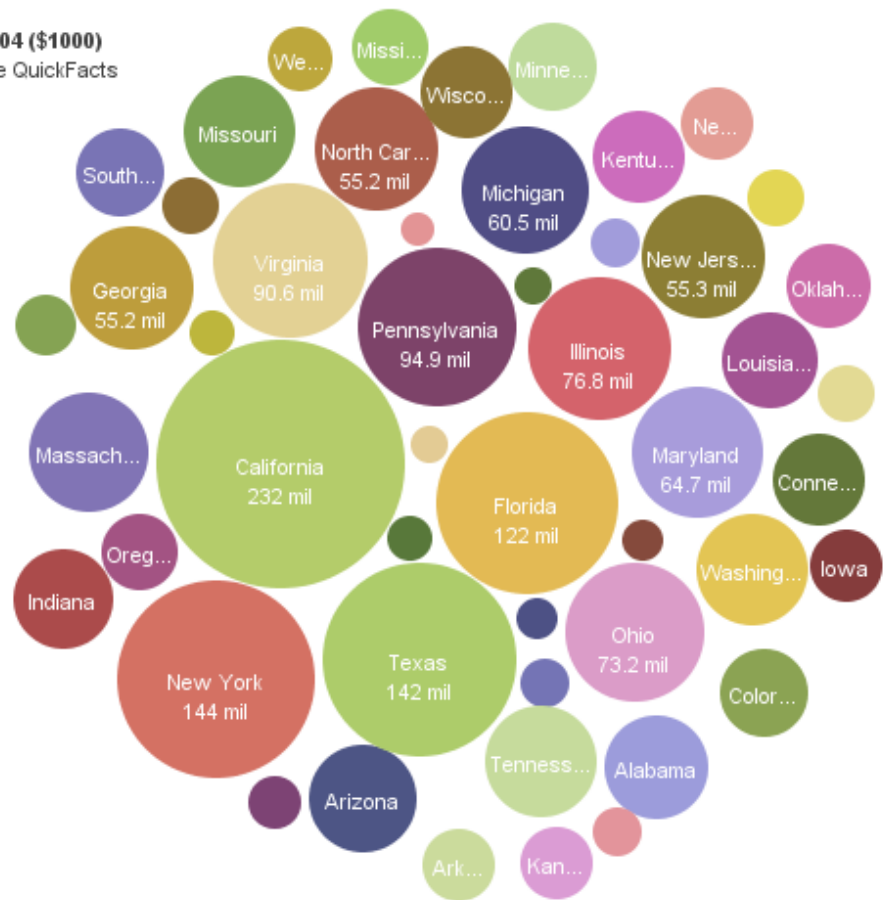
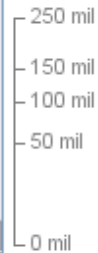
Tags: census people

People QuickFac...

Click to select,
Ctrl-Click: multiple
Shift-Click: range

Federal spending 2004 (\$1000)
Disks colored by People QuickFacts

- Alabama
- Alaska
- Arizona
- Arkansas
- California
- Colorado
- Connecticut
- Delaware
- Florida
- Georgia
- Hawaii
- Idaho
- Illinois
- Indiana
- Iowa
- Kansas
- Kentucky
- Louisiana
- Maine
- Maryland



Search>>

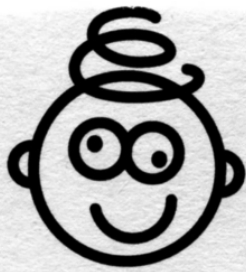
To highlight or find totals
click or ctrl-click.

Bubble Size Label Color

- Retail sales per capita 2002
- Minority-owned firms percent of total 1997
- Women-owned firms percent of total 1997
- Housing units authorized by building permits 2004
- Federal spending 2004 (\$1000)
- Land area 2000 (square miles)
- Persons per square mile 2000
- FIPS Code

Census Bureau

Comments (1)



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 ~~keys~~ 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 minutes
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



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

Columns: Party Year

Rows: SUM(Total..)

Filters:

Level of Detail:

Mark: Automatic

Text:

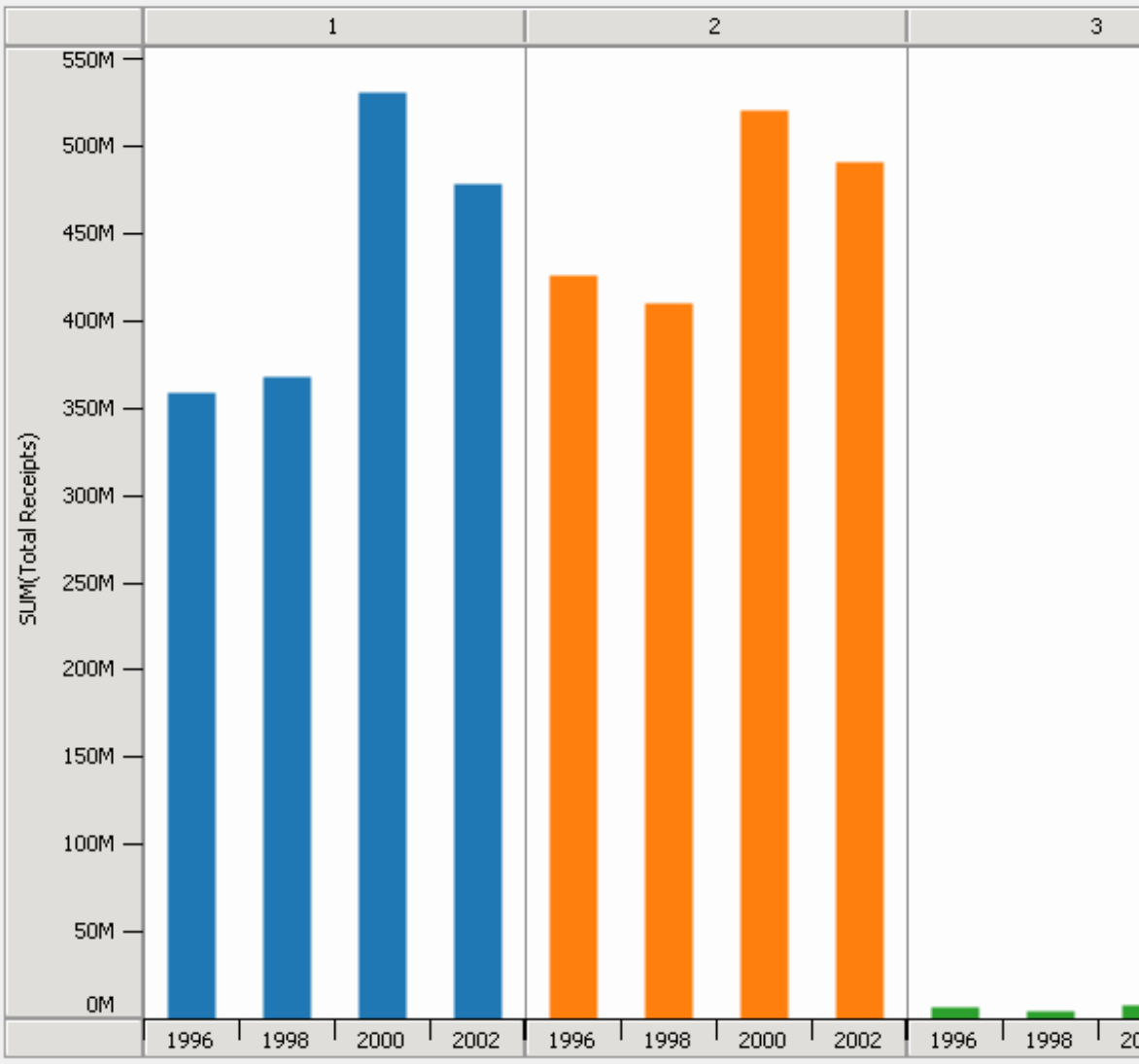
Color: Party

Size:

Legend:

- 1
- 2
- 3

Size:



Statistics and Computing

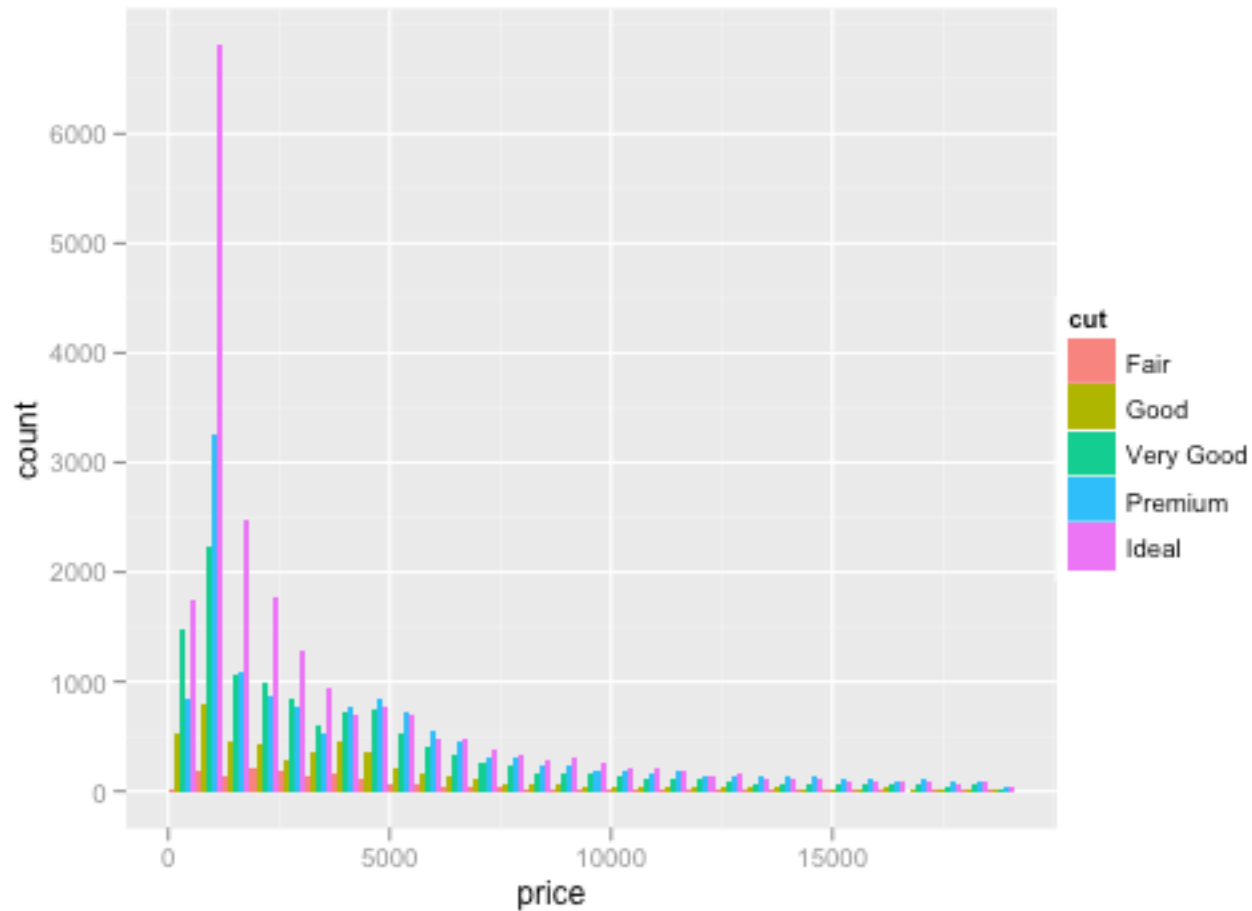
Leland Wilkinson

**The Grammar
of Graphics**

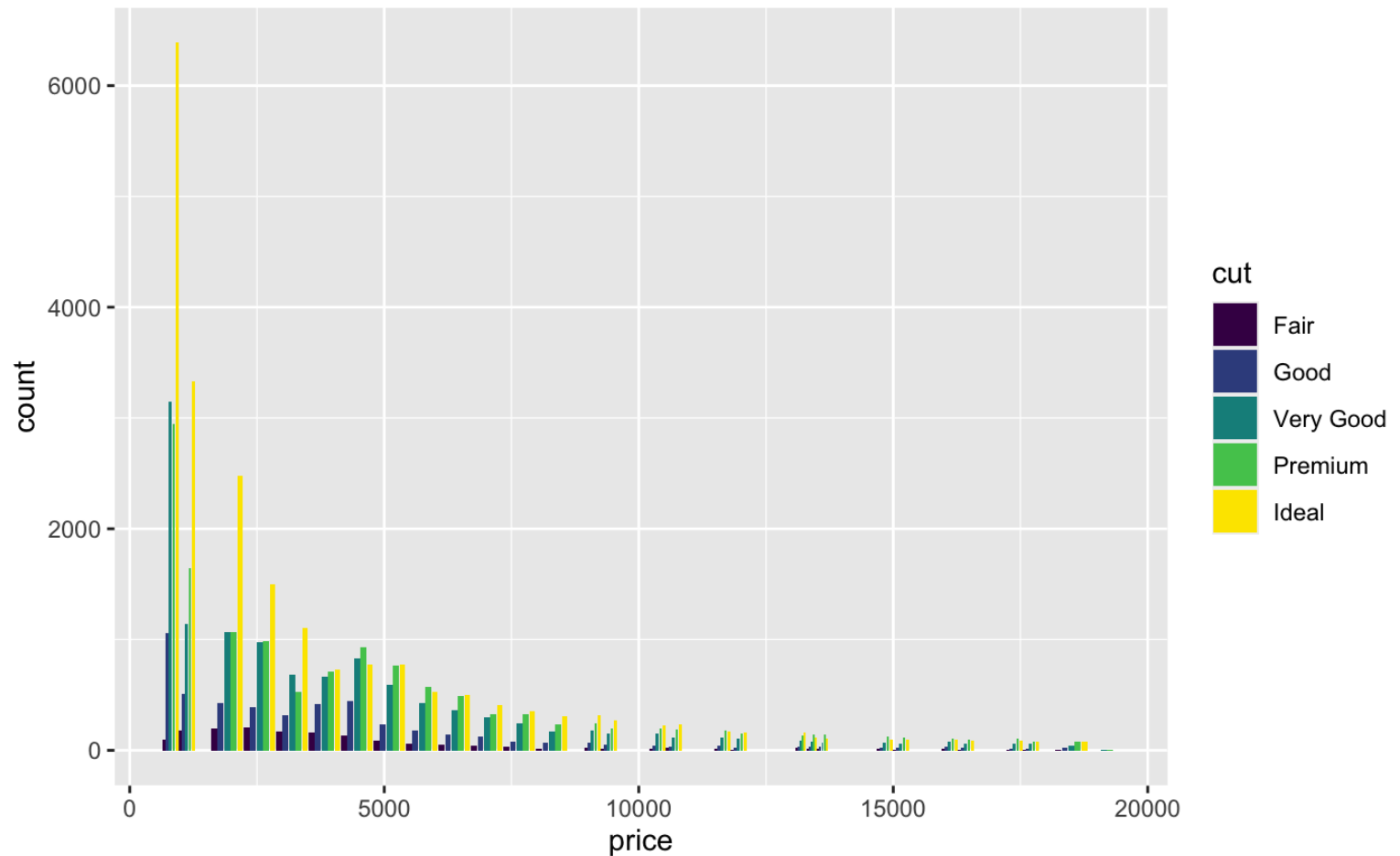
Second Edition

 Springer

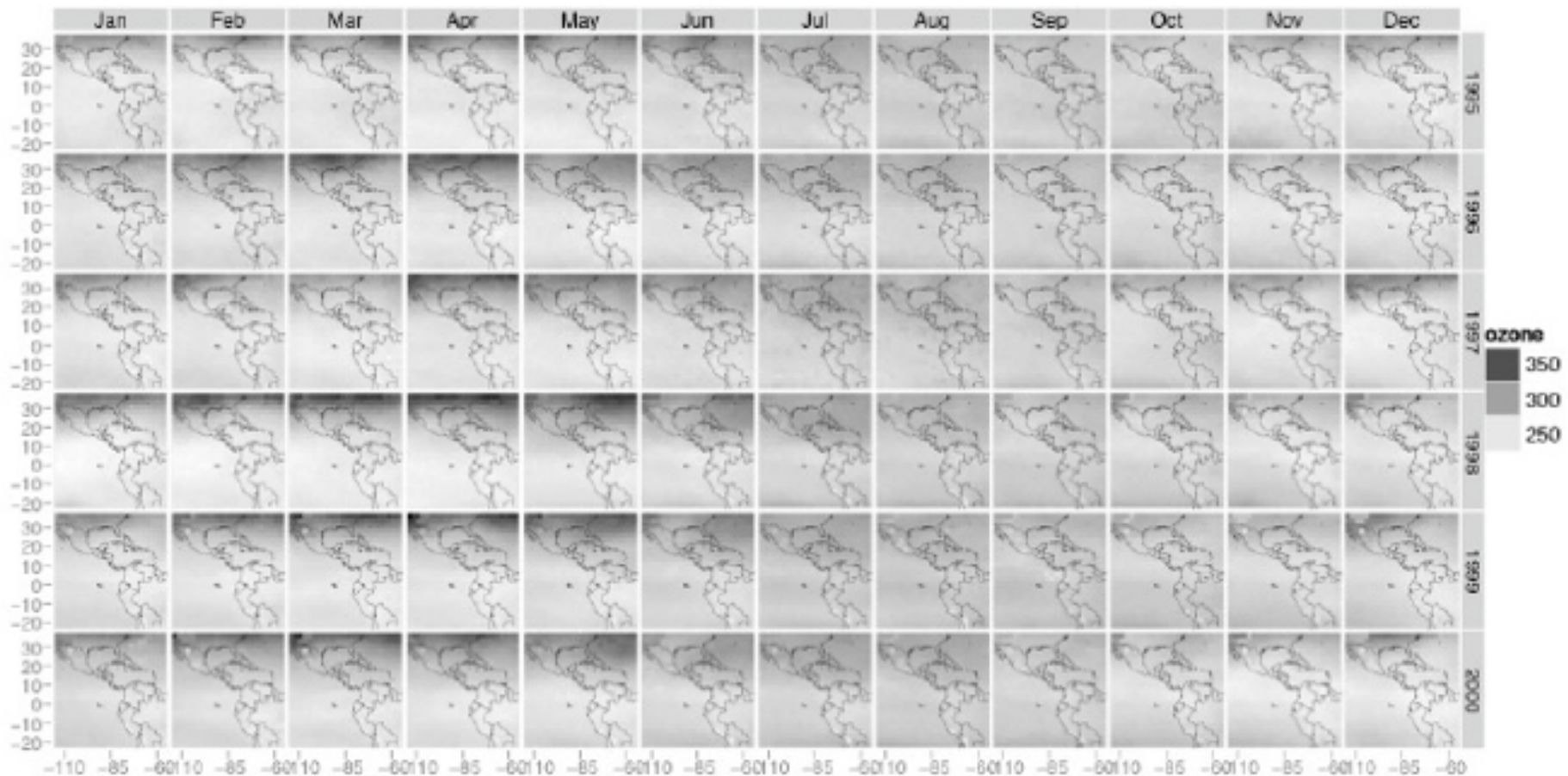
```
ggplot(diamonds, aes(x=price, fill=cut))  
+ geom_bar(position="dodge")
```

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



```
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

Ease-of-Use



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

Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Processing, OpenGL, Java2D

Expressiveness



Ease-of-Use



Chart Typologies

Excel, Many Eyes, Google Charts

Visual Analysis Grammars

VizQL, ggplot2

?

Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Processing, OpenGL, Java2D

Expressiveness



Ease-of-Use



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

Expressiveness

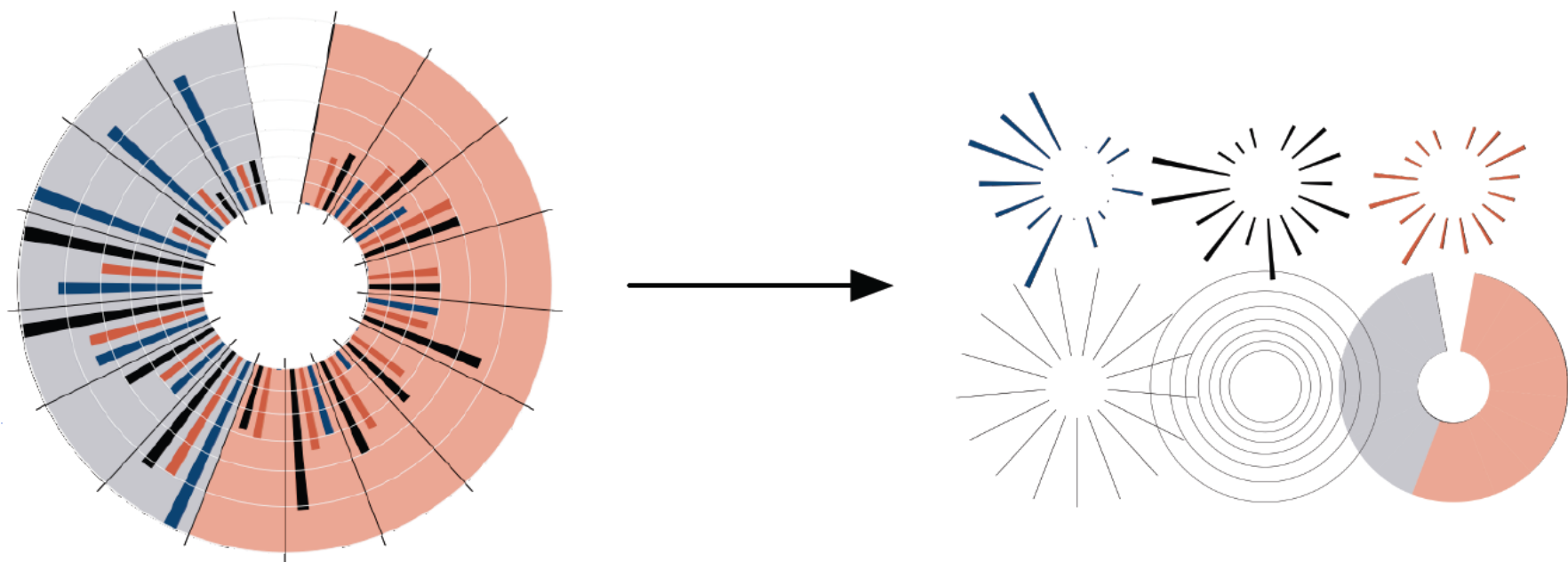


Protovis & D3

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**.

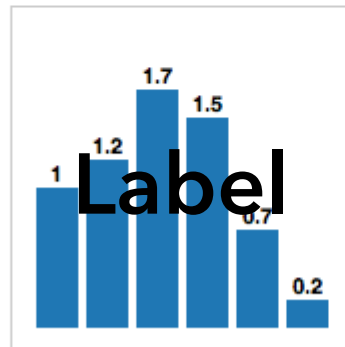
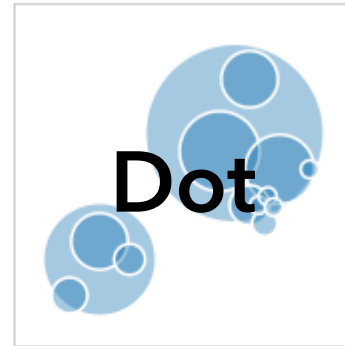
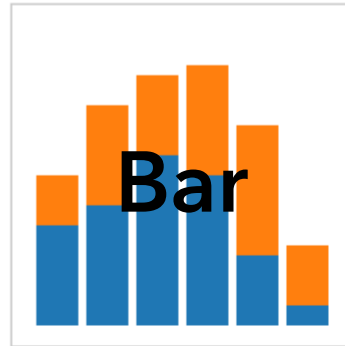
J. W. Tukey, M. B. Wilk
Data Analysis & Statistics, 1965

Protovis: A Grammar for Visualization



A graphic is a composition of data-representative marks.

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

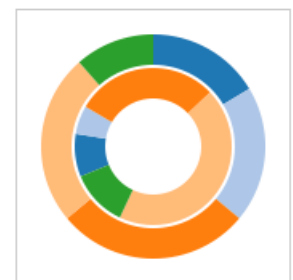
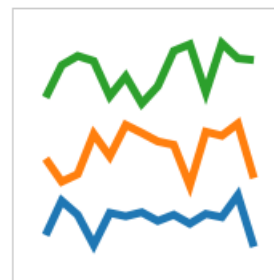
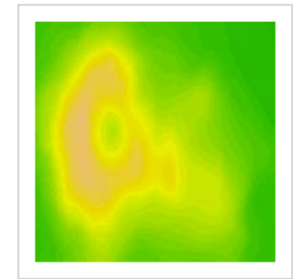
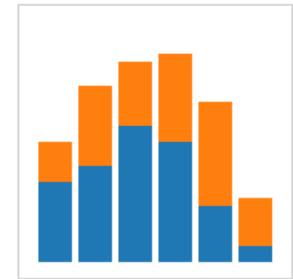


MARKS: Protovis graphical primitives

MARK

$$\lambda : D \rightarrow R$$

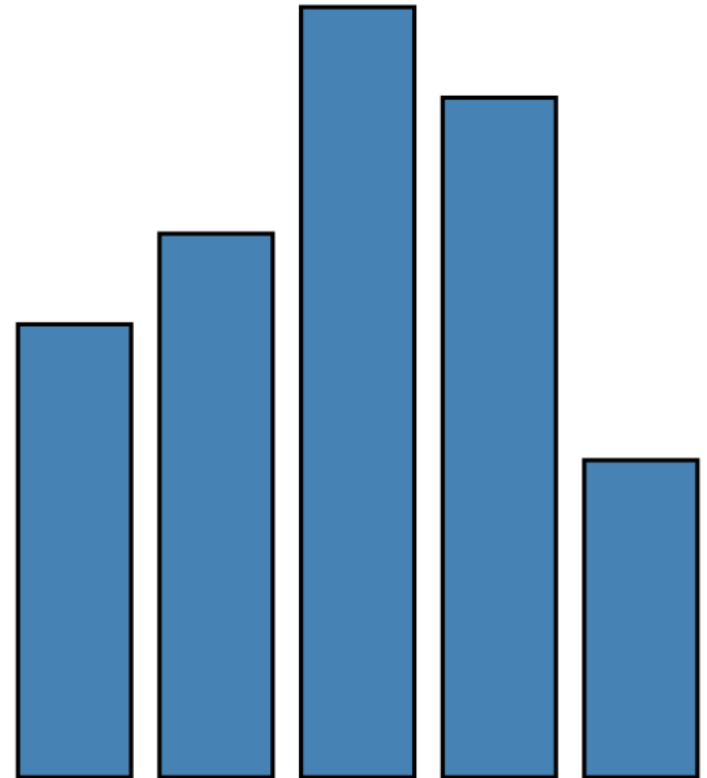
data	λ
visible	λ
left	λ
bottom	λ
width	λ
height	λ
fillStyle	λ
strokeStyle	λ
lineWidth	λ
...	λ



RECT

$\lambda : D \rightarrow R$

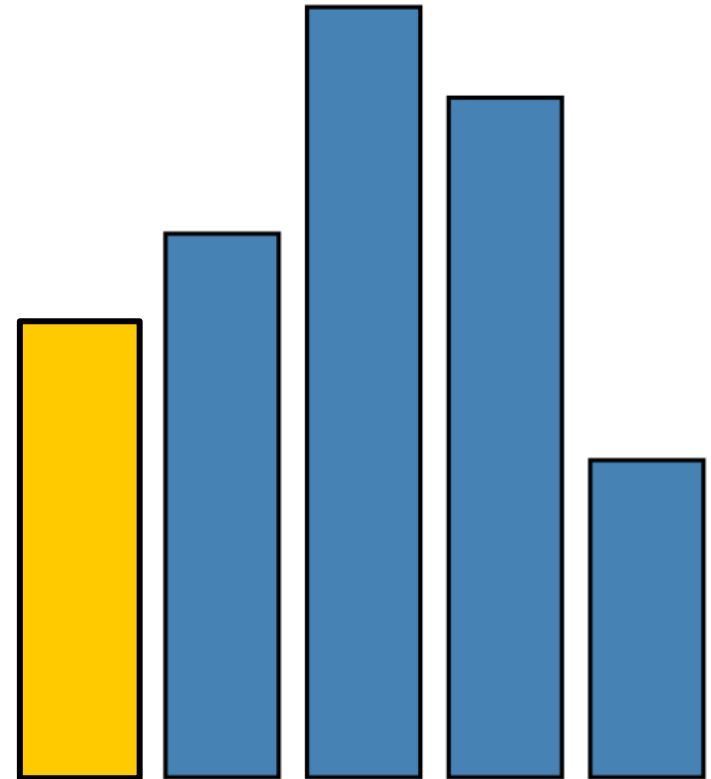
data	1	1.2	1.7	1.5	0.7
visible	true				
left	$\lambda: \text{index} * 25$				
bottom	0				
width	20				
height	$\lambda: \text{datum} * 80$				
fillStyle	blue				
strokeStyle	black				
lineWidth	1.5				
...	...				



RECT

$\lambda : D \rightarrow R$

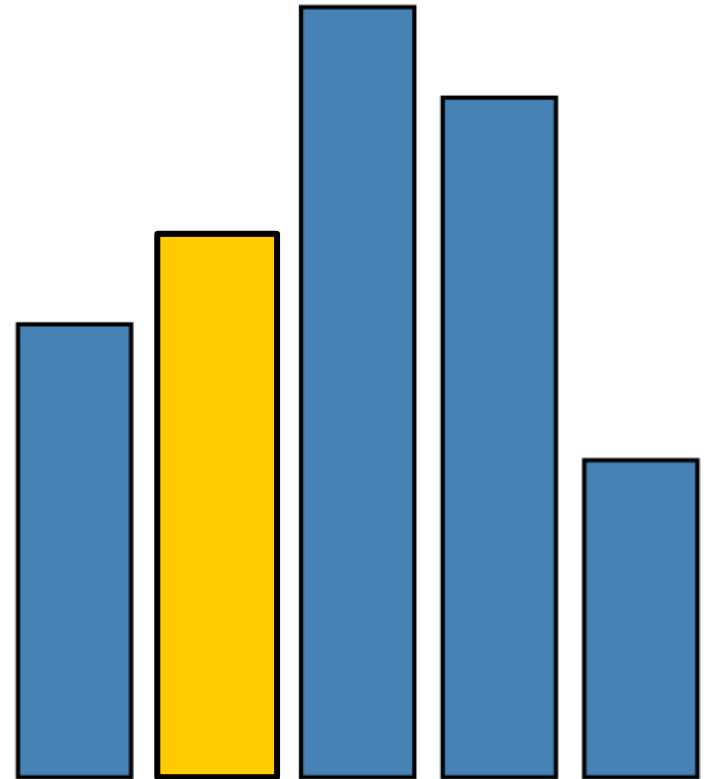
data	1	1.2	1.7	1.5	0.7
visible	true				
left	0 * 25				
bottom	0				
width	20				
height	1 * 80				
fillStyle	blue				
strokeStyle	black				
lineWidth	1.5				
...	...				



RECT

$\lambda : D \rightarrow R$

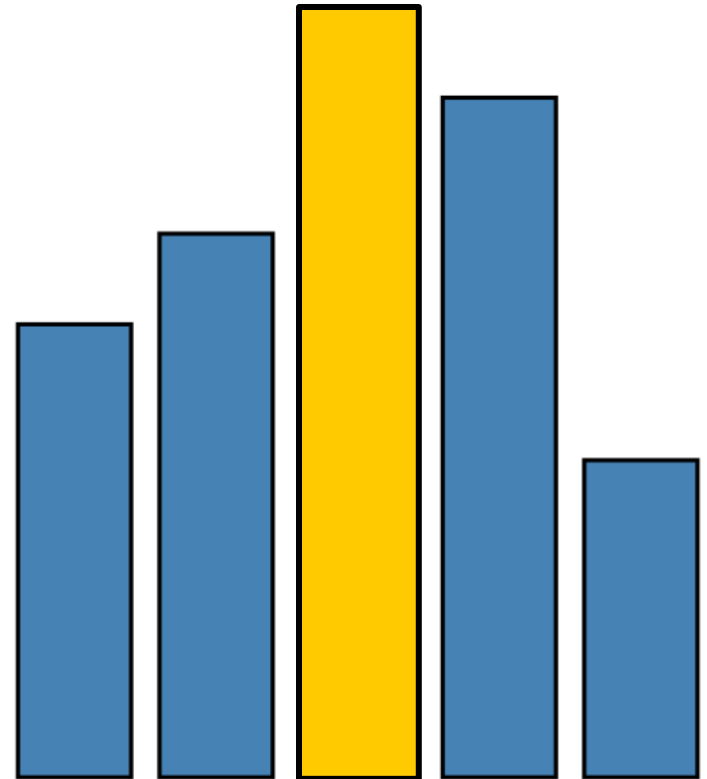
data	1	1.2	1.7	1.5	0.7
visible	true				
left	1 * 25				
bottom	0				
width	20				
height	1.2 * 80				
fillStyle	blue				
strokeStyle	black				
lineWidth	1.5				
...	...				



RECT

$\lambda : D \rightarrow R$

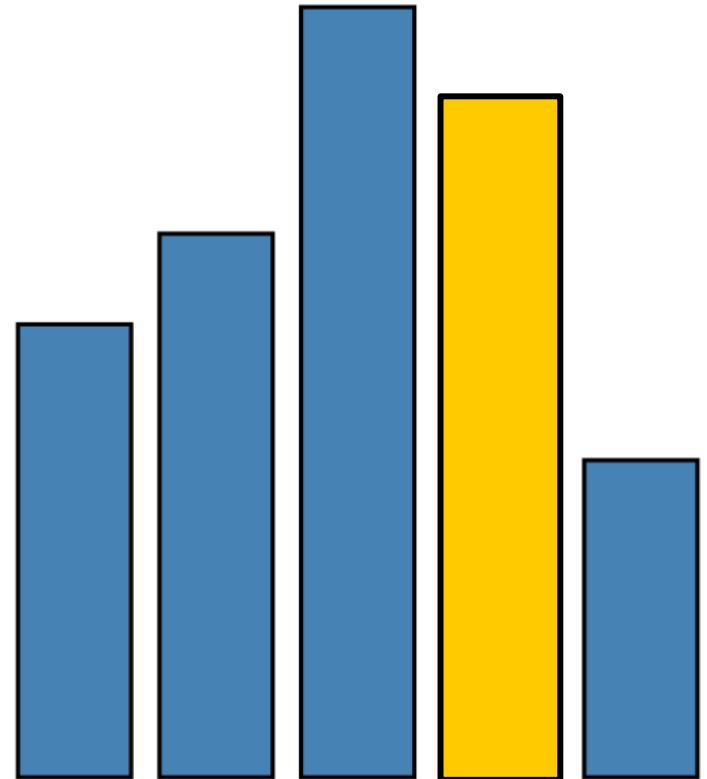
data	1	1.2	1.7	1.5	0.7
visible	true				
left	2 * 25				
bottom	0				
width	20				
height	1.7 * 80				
fillStyle	blue				
strokeStyle	black				
lineWidth	1.5				
...	...				



RECT

$\lambda : D \rightarrow R$

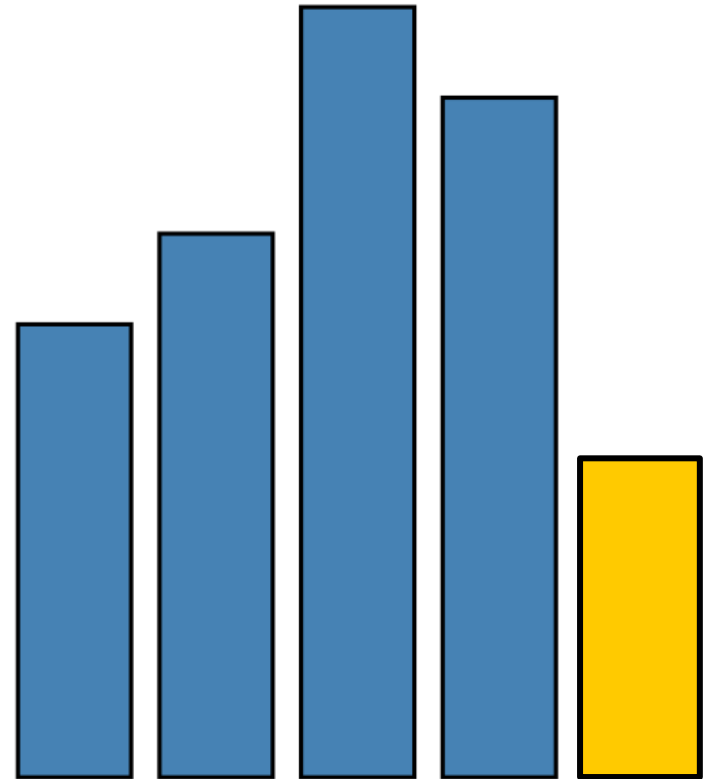
data	1	1.2	1.7	1.5	0.7
visible	true				
left	3 * 25				
bottom	0				
width	20				
height	1.5 * 80				
fillStyle	blue				
strokeStyle	black				
lineWidth	1.5				
...	...				



RECT

$\lambda : D \rightarrow R$

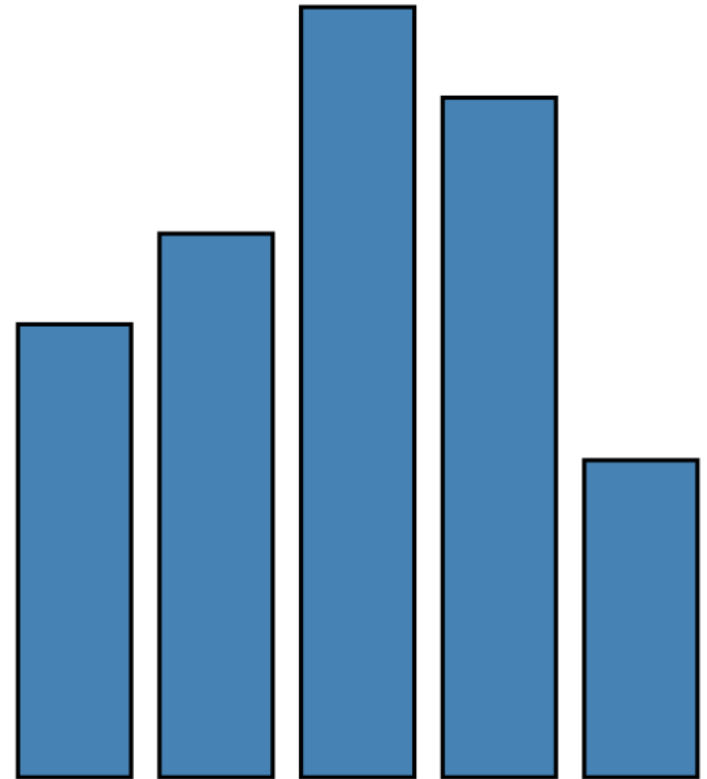
data	1	1.2	1.7	1.5	0.7
visible	true				
left	4 * 25				
bottom	0				
width	20				
height	0.7 * 80				
fillStyle	blue				
strokeStyle	black				
lineWidth	1.5				
...	...				



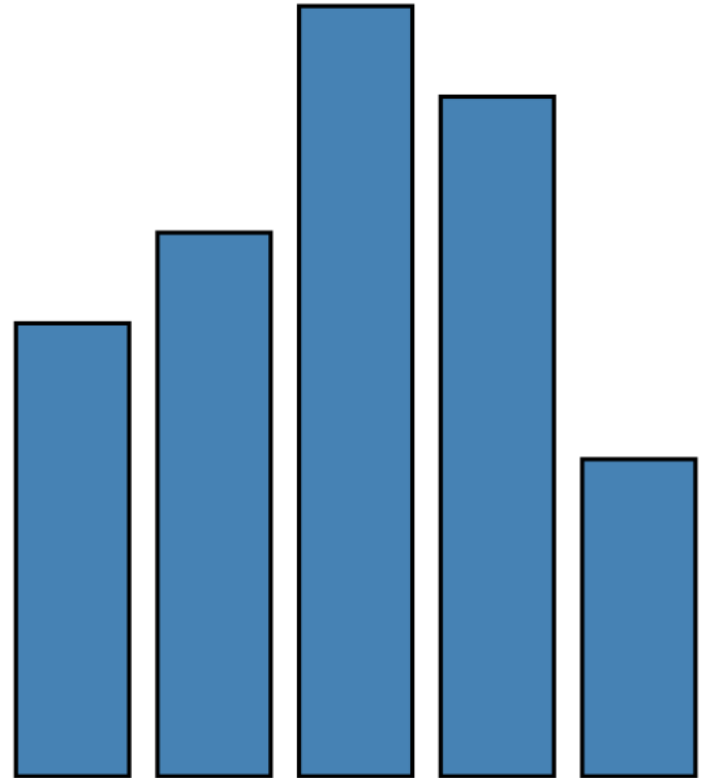
RECT

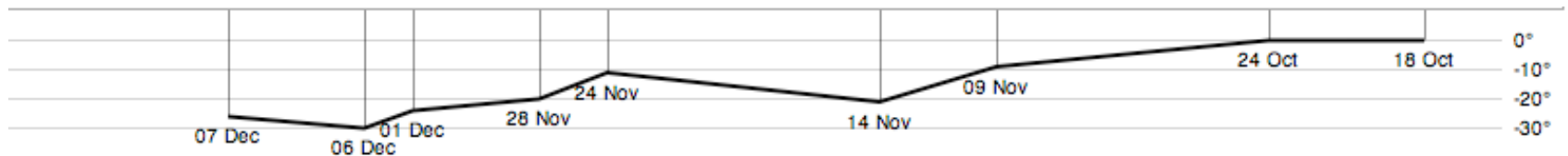
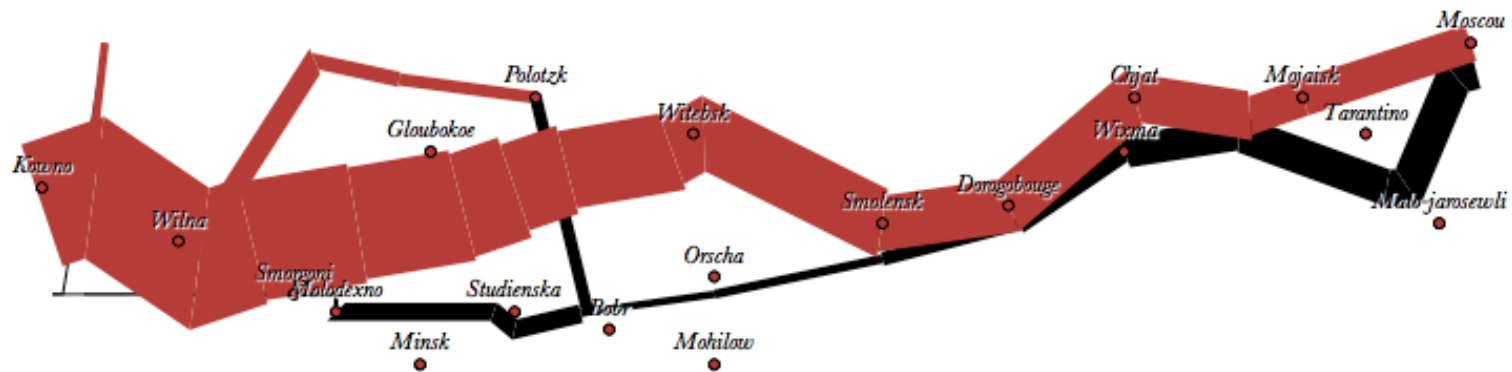
$\lambda : D \rightarrow R$

data	1	1.2	1.7	1.5	0.7
visible	true				
left	$\lambda: \text{index} * 25$				
bottom	0				
width	20				
height	$\lambda: \text{datum} * 80$				
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[panelIndex][0].dir]);
```

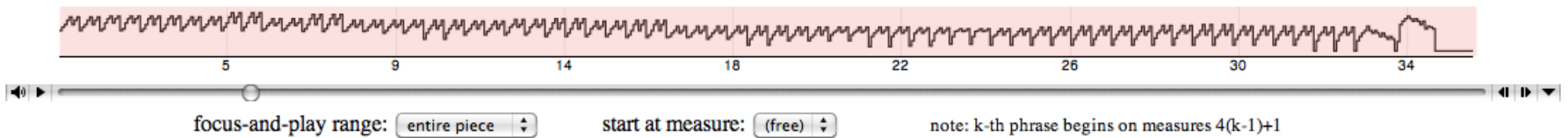
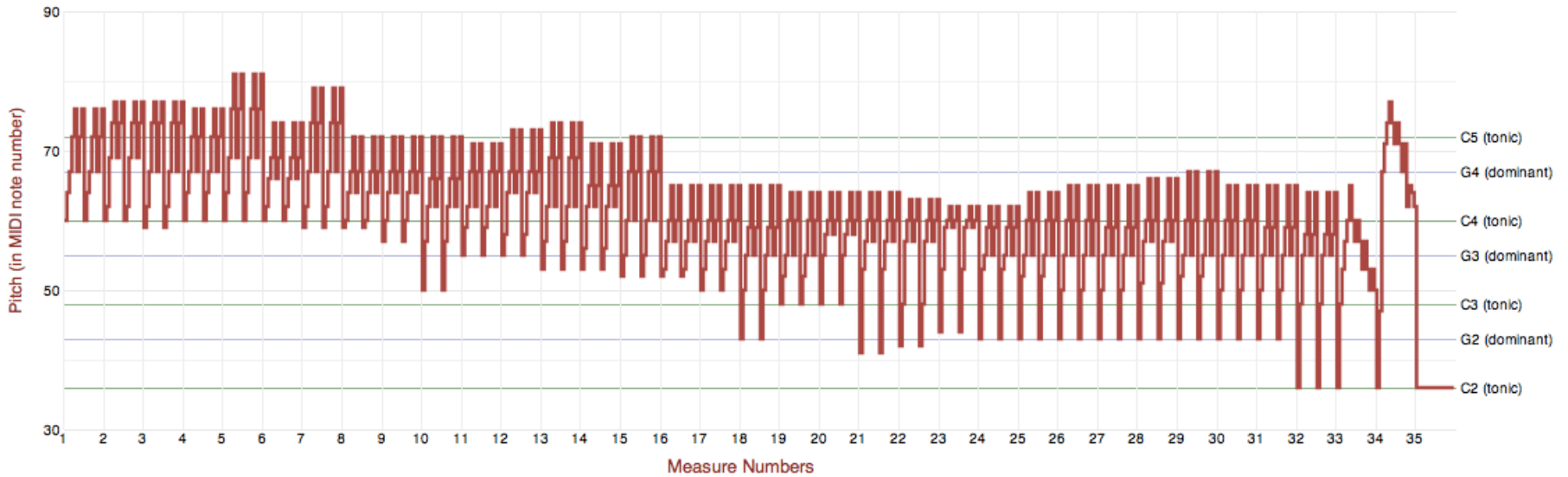
```
vis.add(pv.Label).data(napoleon.cities)
  .left(lon).top(lat)
  .text((d) => d.city).font("italic 10px Georgia")
  .textAlign("center").textBaseline("middle");
```

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

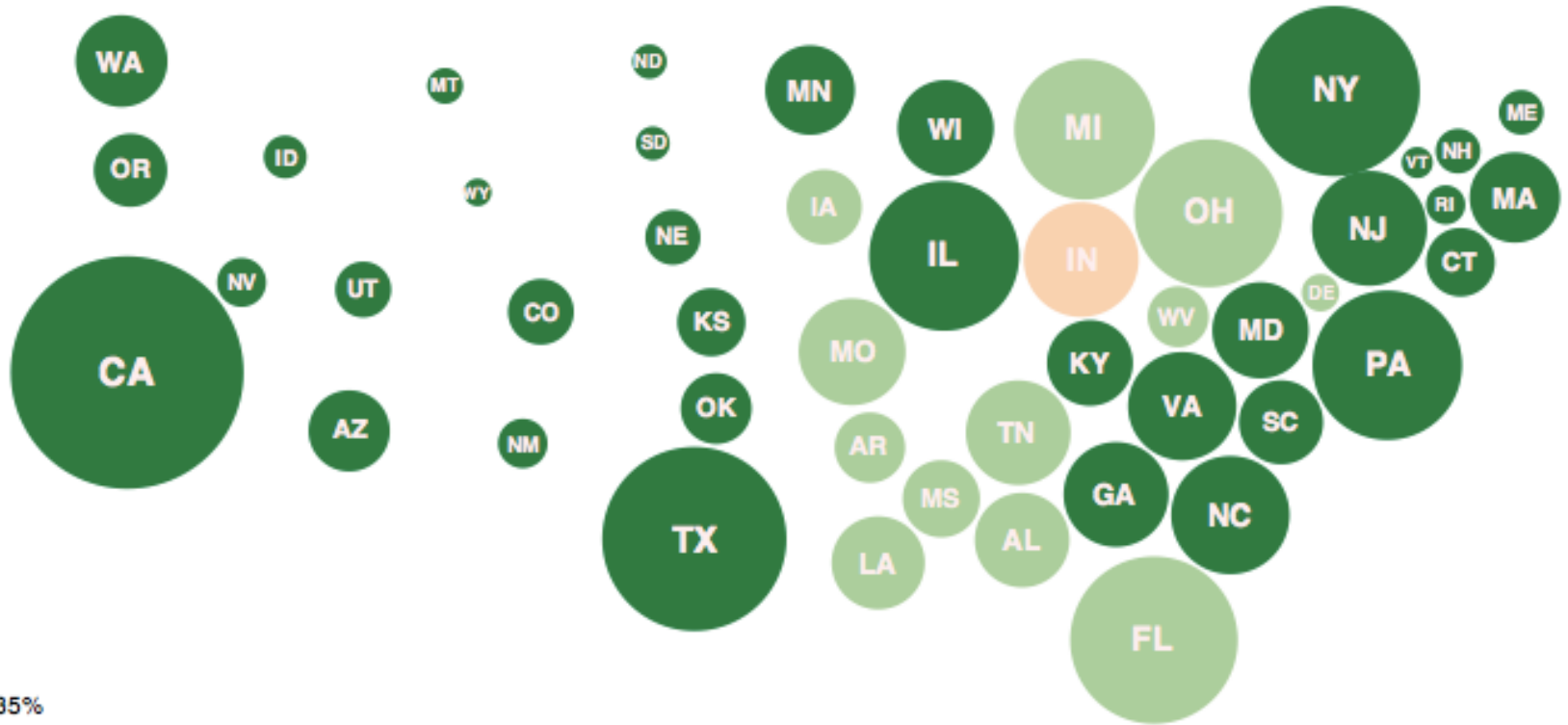
vis.add(pv.Line).data(napoleon.temp)
  .left(lon).top(tmp) .strokeStyle("#0")
  .add(pv.Label)
  .top((d) => 5 + tmp(d))
  .text((d) => d.temp+"° "+d.date.substr(0,6))
  .textBaseline("top").font("italic 10px Georgia");
```

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

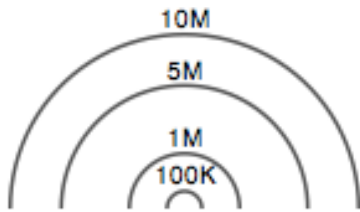
BY J.S. BACH

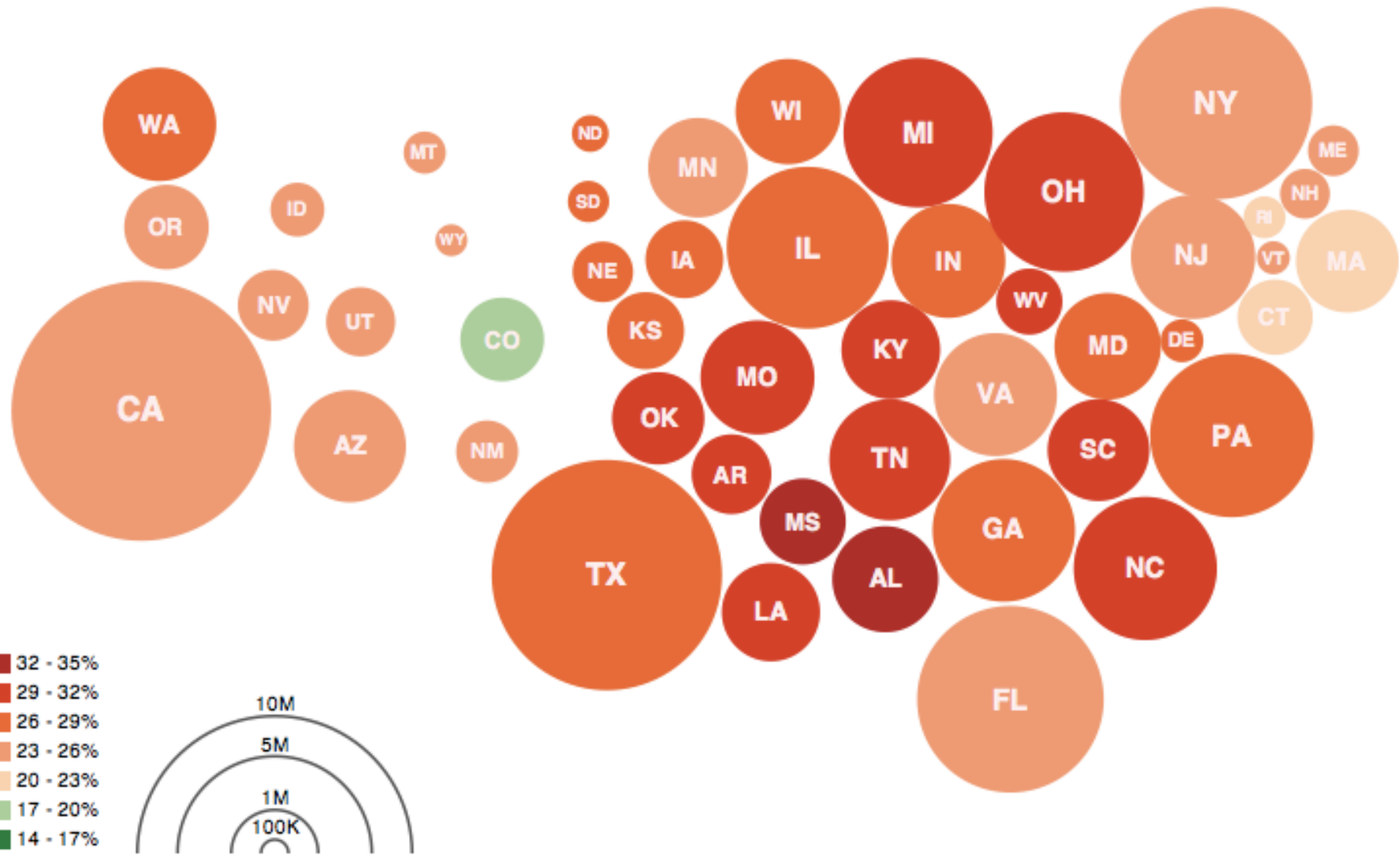


Bach's Prelude #1 in C Major | Jieun Oh

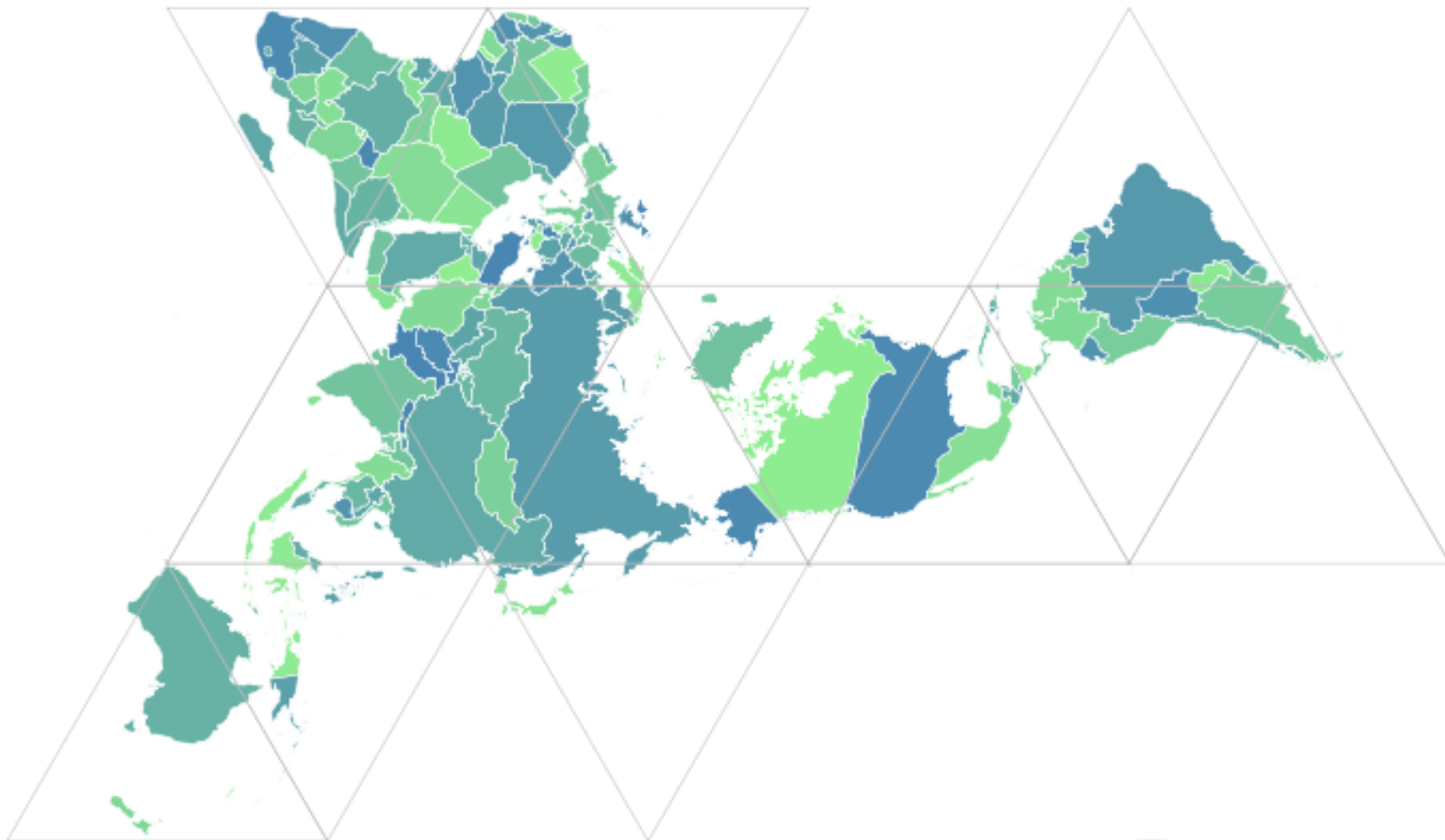


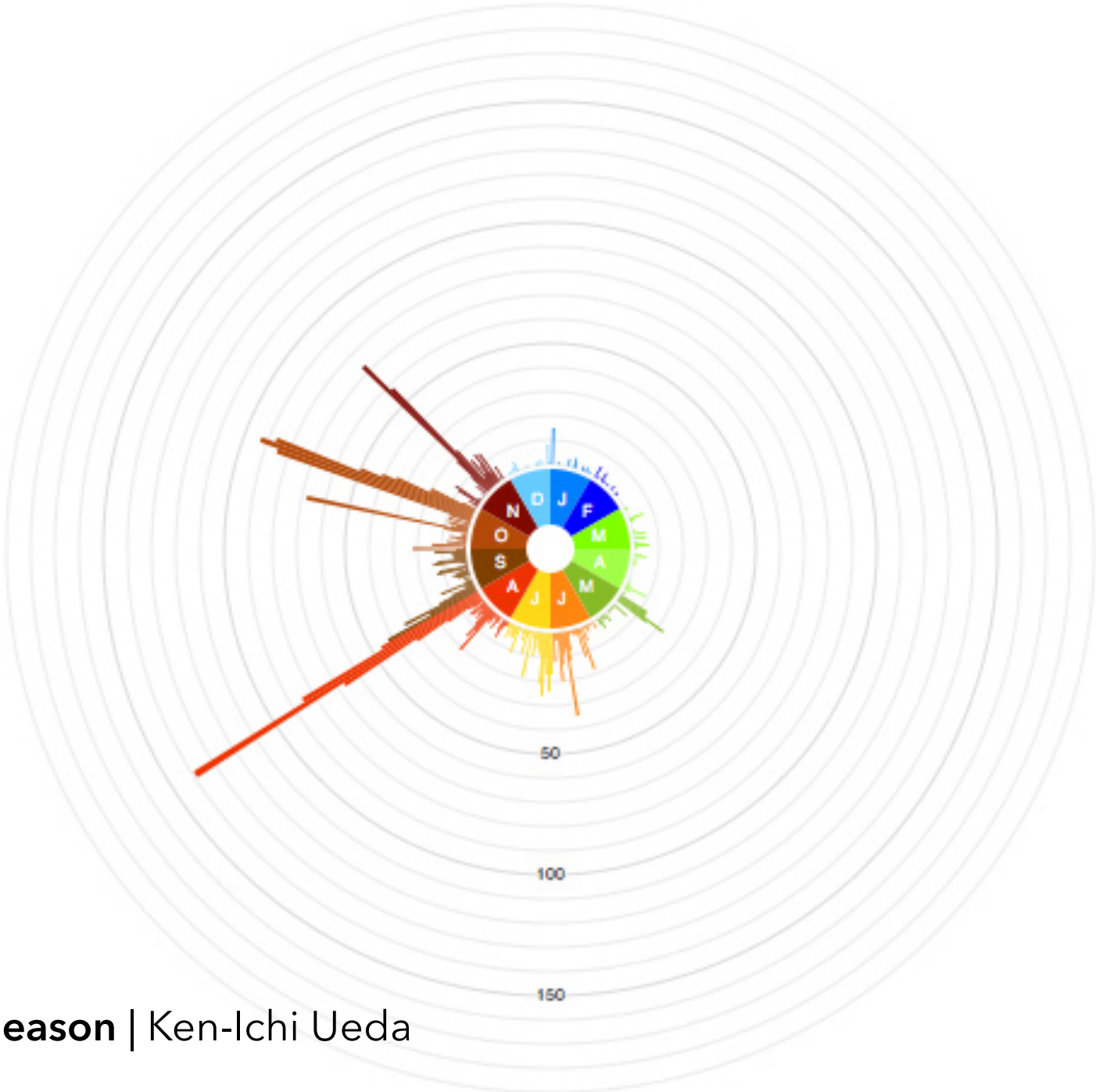
- 32 - 35%
- 29 - 32%
- 26 - 29%
- 23 - 26%
- 20 - 23%
- 17 - 20%
- 14 - 17%



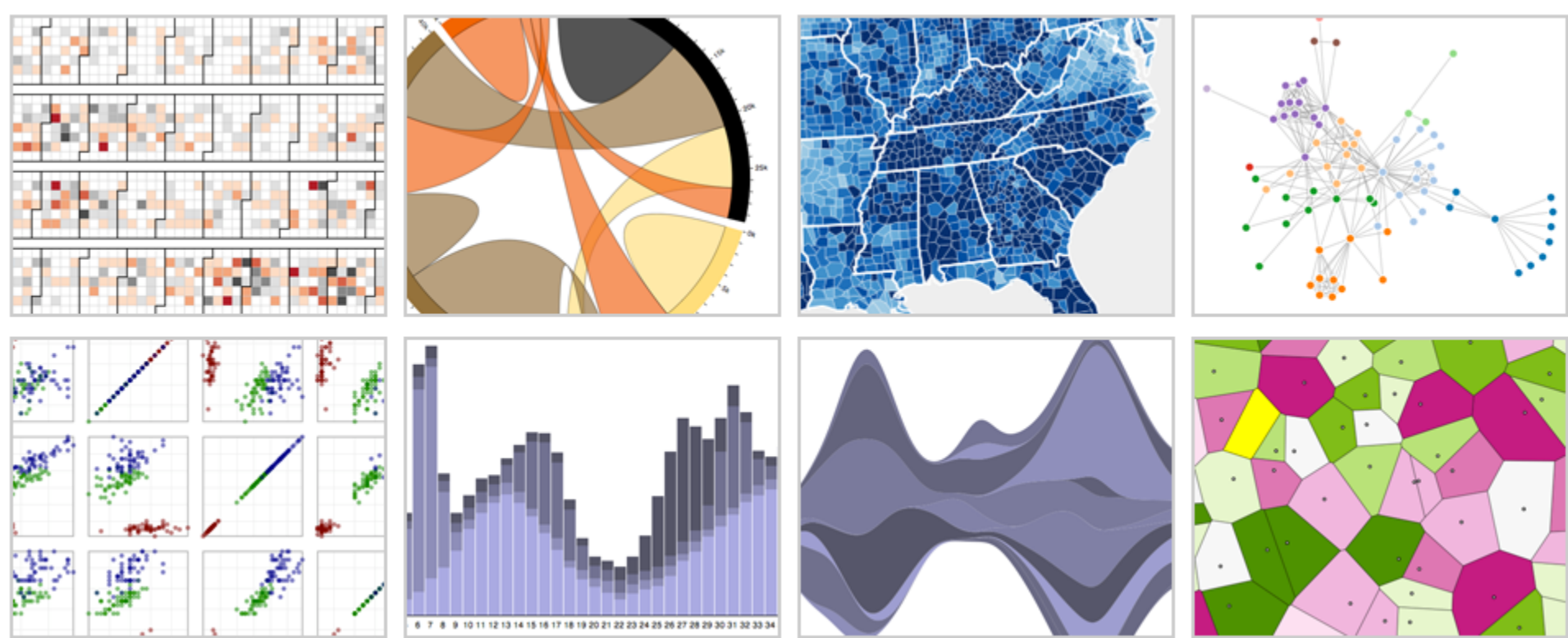


Obesity Map | Vadim Ogievetsky





d3.js Data-Driven Documents



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

Protovis

Specialized mark types

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

Specify a scene (nouns)

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

Protovis

Specialized mark types

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

Specify a scene (nouns)

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

D3

Bind data to DOM

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

Transform a scene (verbs)

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

Ease-of-Use



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

Expressiveness



Administrivia

Reminders!

Assignment 3 Due **tonight, Mon 2/8, 11:59pm**

<https://courses.cs.washington.edu/courses/cse412/21wi/a3.html>

Final Project Proposal Due **Fri 2/12, 11:59pm**

<https://courses.cs.washington.edu/courses/cse412/21wi/fp.html>

Four Peer Evaluations Due **Mon 2/15, 11:59pm**

<https://courses.cs.washington.edu/courses/cse412/21wi/a3b.html>

A3 Assignment Peer Critiques

Review **four** A3 submissions (assigned on Canvas)

Submit **four** critique forms by **Mon 2/15, 11:59**

Assignments will be posted **tomorrow afternoon** after the A3 deadline (announced on Ed).

Please submit A3 on time! Assignments submitted late will not receive any peer evaluations.

A3 Assignment Peer Critiques

Review **four** A3 submissions (assigned on Canvas)

Submit **four** critique forms by **Mon 2/15, 11:59**

Follow **I like / I wish / What if?** format for critiques
Be positive! Be constructive! Share crazy ideas!

Results discussed in class on Wed 2/17

<https://courses.cs.washington.edu/courses/cse412/21wi/a3b.html>

Critique Questions

What is the purpose of the visualization?

Does it serve its purpose well?

Does it convey the data honestly?

Does it show the appropriate level of detail?

Are expressive & effective visual encodings used?

Is the design well-organized? Is it innovative?

What would you like to change or refine?

How might things be done differently?

I Like... / I Wish... / What If?

I LIKE...

Praise for design ideas and/or well-executed implementation details. *Example: "I like the navigation through time via the slider; the patterns observed as one moves forward are compelling!"*

I WISH...

Constructive statements on how the design might be improved or further refined. *Example: "I wish moving the slider caused the visualization to update immediately, rather than the current lag."*

WHAT IF?

Suggest alternative design directions, or even wacky half-baked ideas. *Example: "What if we got rid of the slider and enabled direct manipulation navigation by dragging data points directly?"*

Critique Categories

Visualization Design (Visual Encodings)

Choice of visual encodings (expressive, effective?)

Is the appropriate information visible by default?

Overall Design Quality

Organization, legibility, fitness for chosen goals

Task Effectiveness

Is the viewer readily able to answer the question?

Is the *ethical* design clear and transparent?

Is the *deceptive* design subtly misleading?

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

Declarative
Languages

Visualization Grammars

Protovis, D3.js

Component Architectures

Prefuse, Flare, Improvise, VTK

Programming
Toolkits

Graphics APIs

Processing, OpenGL, Java2D

Chart Typologies

Excel, Many Eyes, Google Charts

Charting
Tools

Visual Analysis Grammars

VizQL, ggplot2

Declarative
Languages

Visualization Grammars

Protovis, D3.js

Component Architectures

Prefuse, Flare, Improvise, VTK

Programming
Toolkits

Graphics APIs

Processing, OpenGL, Java2D

What is a Declarative Language?

Programming by describing *what*, not *how*

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

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

What is a Declarative Language?

Programming by describing *what*, not *how*

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

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

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

The New York Times

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

Search [ING DIRECT](#)

— 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)

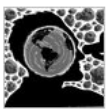


Joshua Kristal for The New York Times

Painting at 99, With No Compromises

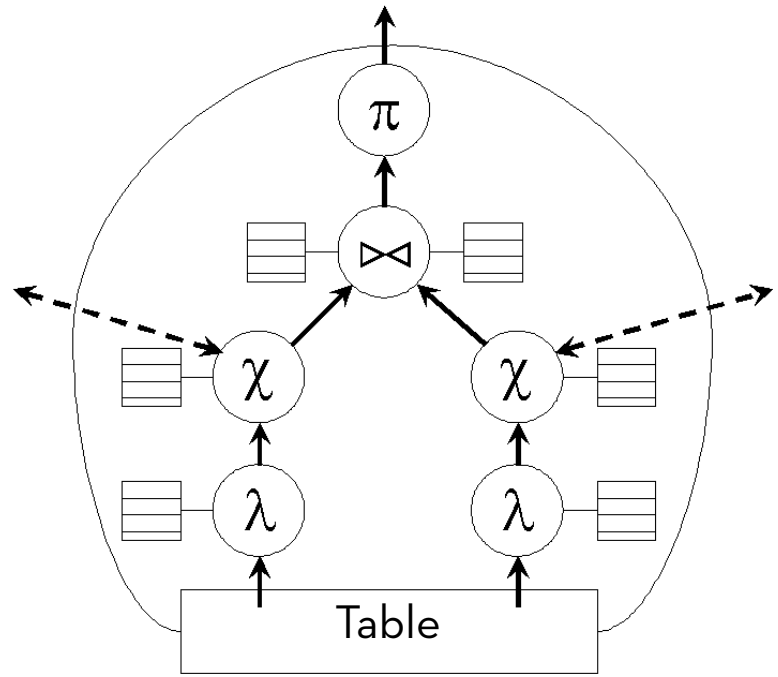
By ROBIN FINN
An exhibition celebrating Will Barnet's centennial year traces his evolution as a modern American artist.

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>
</script>
<span id="to_scrip"></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

Why Declarative Languages?

Faster iteration. Less code. Larger user base.

Better visualization. *Smart defaults.*

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

Performance. *Optimization, scalability.*

Portability. *Multiple devices, renderers, inputs.*

Programmatic generation.

Write programs which output visualizations.

Automated search & recommendation.

Chart Typologies

Excel, Many Eyes, Google Charts

Charting
Tools

Visual Analysis Grammars

VizQL, ggplot2

Declarative
Languages

Visualization Grammars

Protovis, D3.js

Component Architectures

Prefuse, Flare, Improvise, VTK

Programming
Toolkits

Graphics APIs

Processing, OpenGL, Java2D

Chart Typologies

Excel, Many Eyes, Google Charts

Charting
Tools

Visual Analysis Grammars

VizQL, ggplot2, *Vega-Lite*

Declarative
Languages

Visualization Grammars

Protovis, D3.js, *Vega*

Component Architectures

Prefuse, Flare, Improvise, VTK

Programming
Toolkits

Graphics APIs

Processing, OpenGL, Java2D

Chart Typologies

Excel, Many Eyes, Google Charts



Charting
Tools

Visual Analysis Grammars

VizQL, ggplot2, *Vega-Lite*

Declarative
Languages

Visualization Grammars

Protovis, D3.js, *Vega*

Component Architectures

Prefuse, Flare, Improvise, VTK

Programming
Toolkits

Graphics APIs

Processing, OpenGL, Java2D

Visual Analysis Grammars

VizQL, ggplot2, *Vega-Lite*

Declarative
Languages

Visualization Grammars

Protovis, 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, *Vega-Lite*

Declarative
Languages

Visualization Grammars

Protovis, D3.js, *Vega*

Component Architectures

Prefuse, Flare, Improvise, VTK

Programming
Toolkits

Graphics APIs

Processing, OpenGL, Java2D