Visualization Tools

DSC 106: Data Visualization

Jared Wilber

UC San Diego

How do people create visualizations?

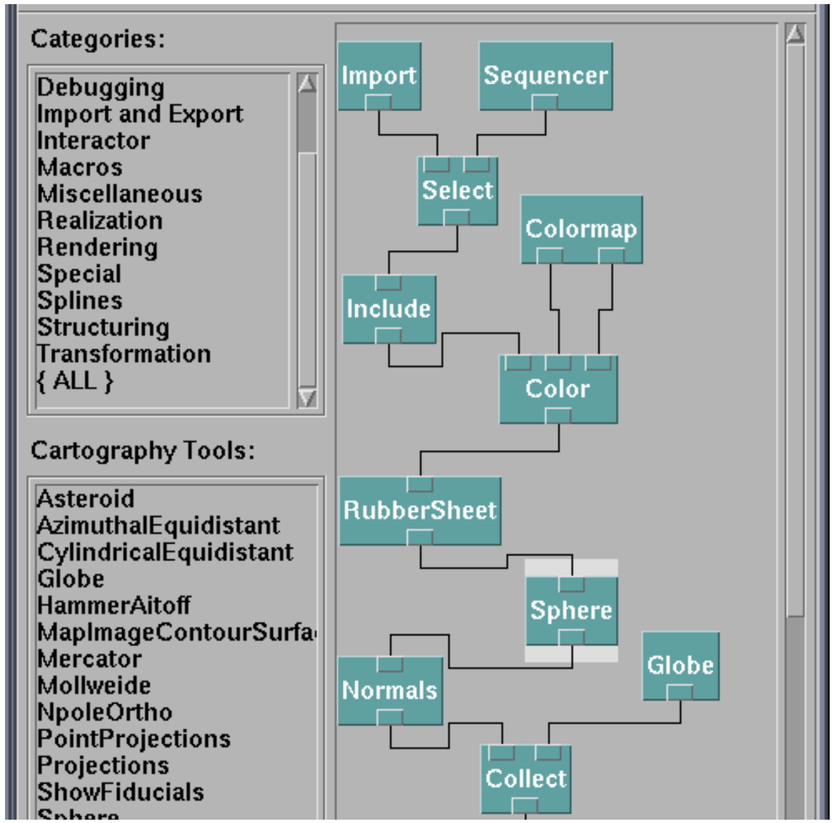


Chart Typology

Pick from a stock of templates

Easy-to-use but limited expressiveness

Prohibits novel designs, new data types

Component Architecture

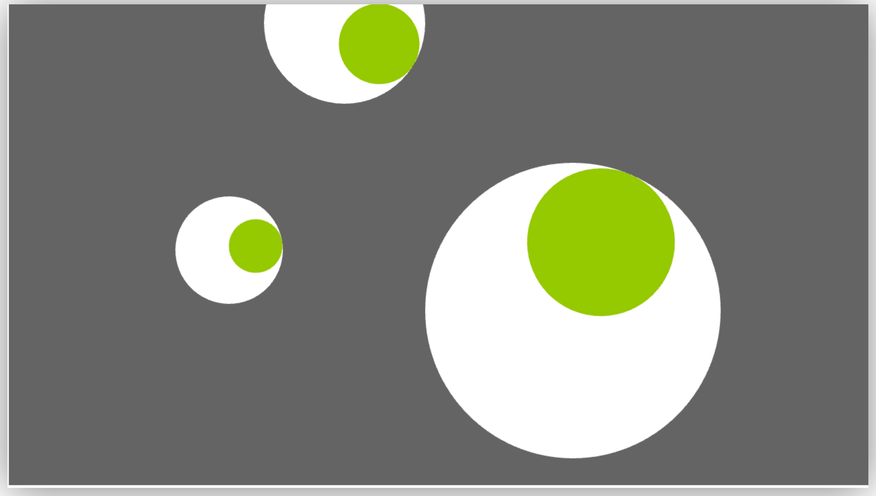
Permits more combinatorial possibilities

Novel views require new operators, which requires software engineering

Graphics APIs

Canvas, OpenGL, Processing, SVG

```
class Eye {
  int x, y;
  int size;
  float angle = 0.0;

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

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

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

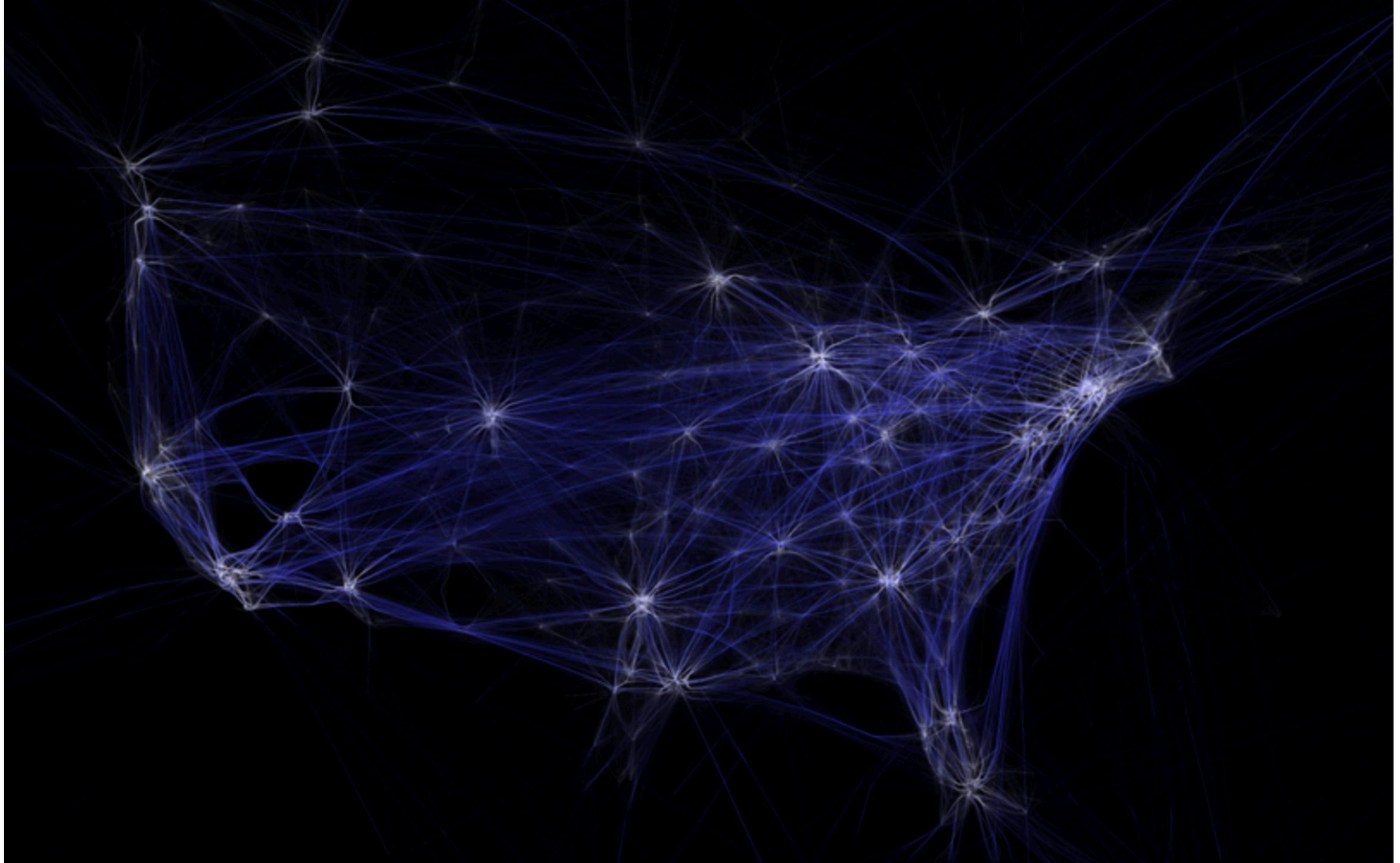


https://processing.org/

User needs to draw individual shapes

Shaders (WebGL)

```
#ifdef GL_ES
precision mediump float;
#endif

#define TWO_PI 6.28318530718

uniform vec2 u_resolution;
uniform float u_time;

//  Function from Iñigo Quiles
//  https://www.shadertoy.com/view/MsS3Wc
vec3 hsb2rgb( in vec3 c ){
    vec3 rgb = clamp(abs(mod(c.x*6.0+vec3(0.0,4.0,2.0),
                             6.0)-3.0)-1.0,
                     0.0,
                     1.0 );
    rgb = rgb*rgb*(3.0-2.0*rgb);
    return c.z * mix( vec3(1.0), rgb, c.y);
}

void main(){
    vec2 st = gl_FragCoord.xy/u_resolution;
    vec3 color = vec3(0.0);

    // Use polar coordinates instead of cartesian
    vec2 toCenter = vec2(0.5)-st;
    float angle = atan(toCenter.y,toCenter.x);
    float radius = length(toCenter)*2.0;

    // Map the angle (-PI to PI) to the Hue (from 0 to 1)
    // and the Saturation to the radius
    color = hsb2rgb(vec3((angle/TWO_PI)+0.5,radius,1.0));

    gl_FragColor = vec4(color,1.0);
}

```



Graphics APIs

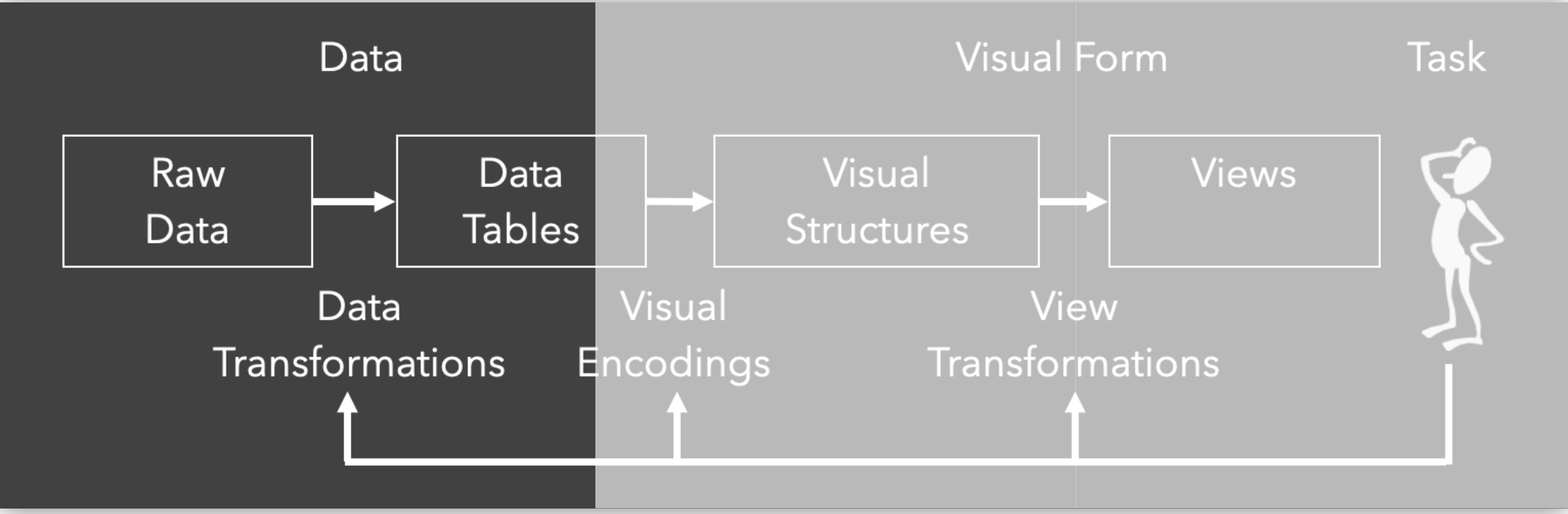
Canvas, OpenGL, Processing, SVG

Component Architectures

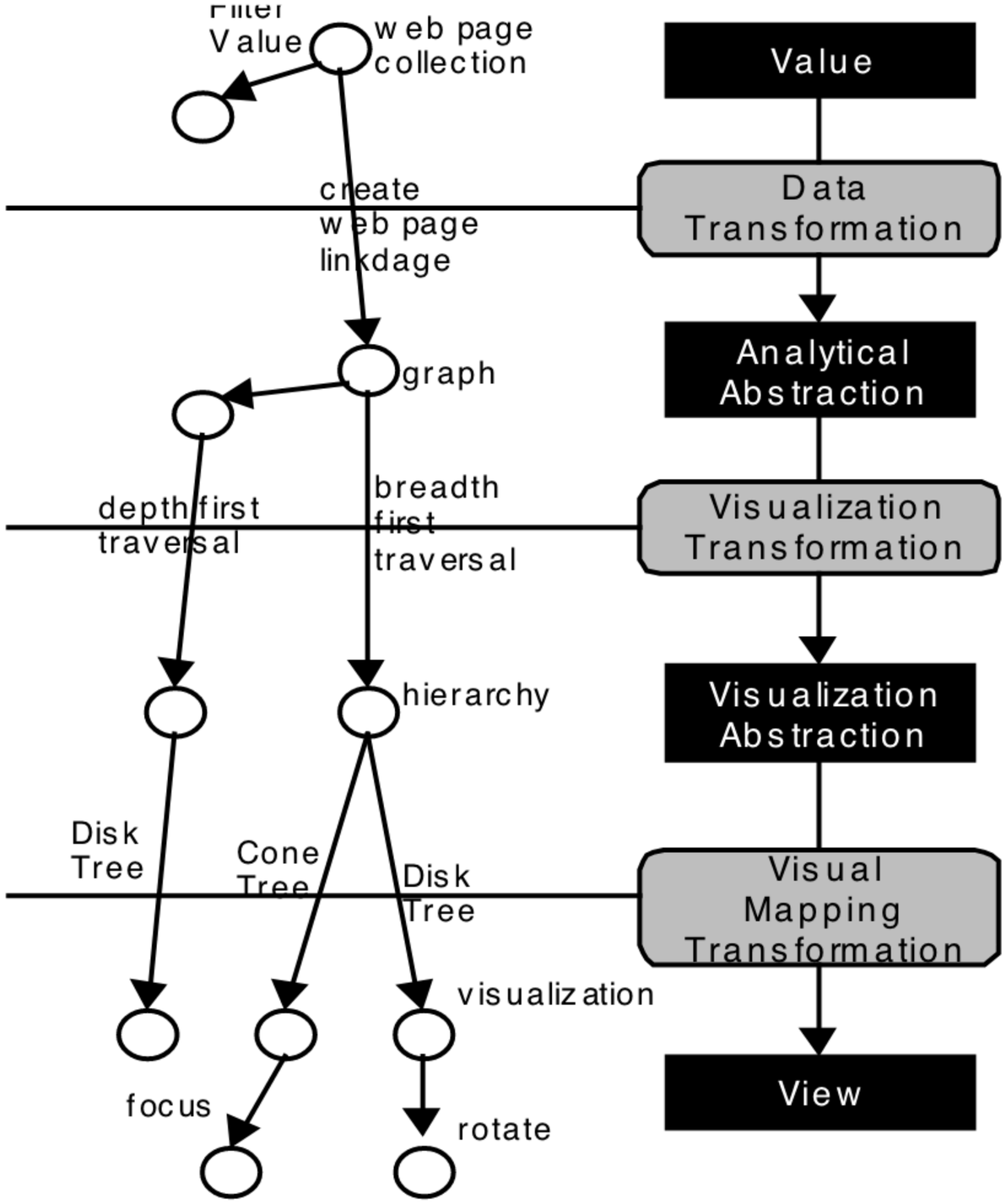
Prefuse, Flare, Improvise, VTK

Graphics APIs

Canvas, OpenGL, Processing, SVG


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

Data State Model
[Chi 98]



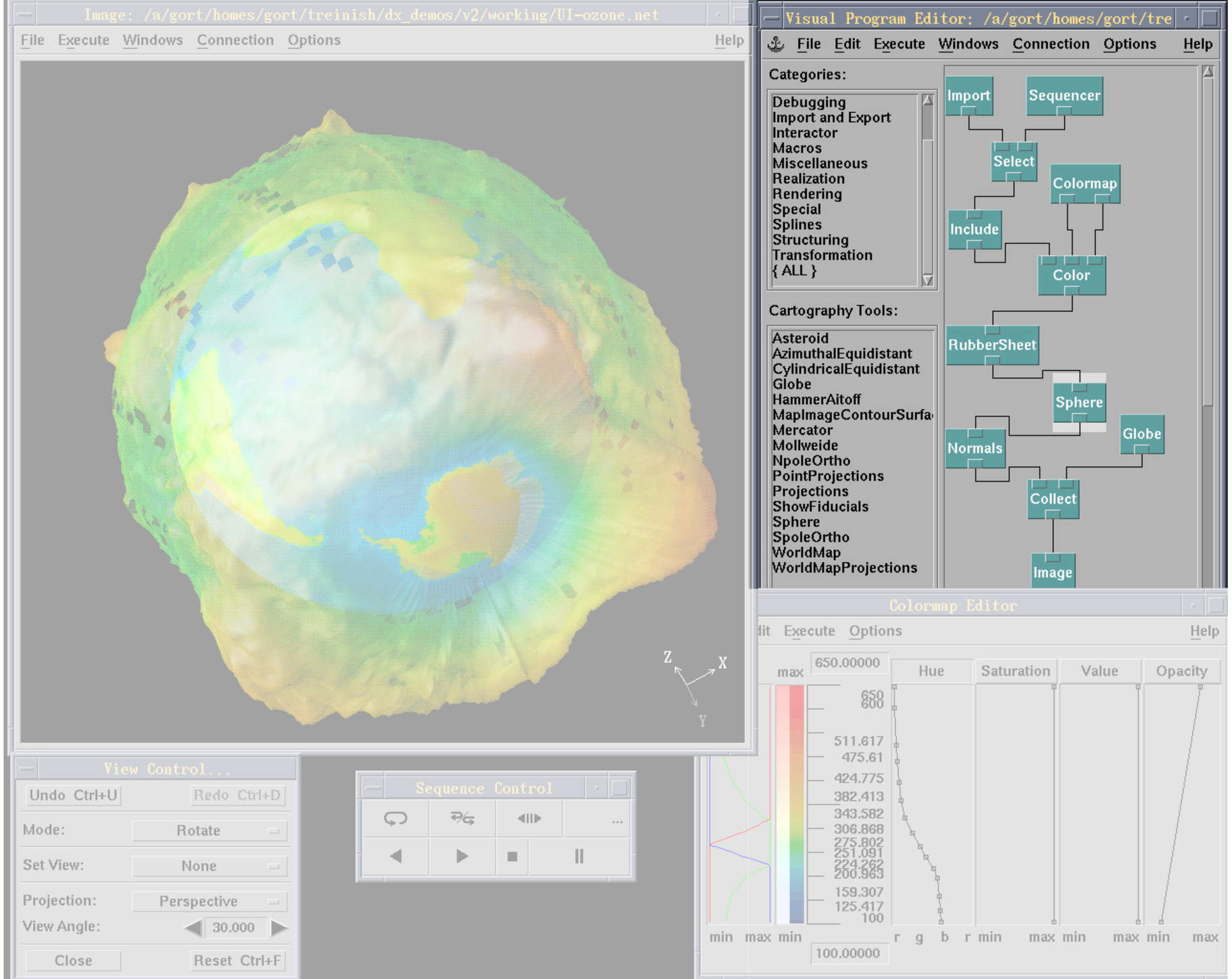
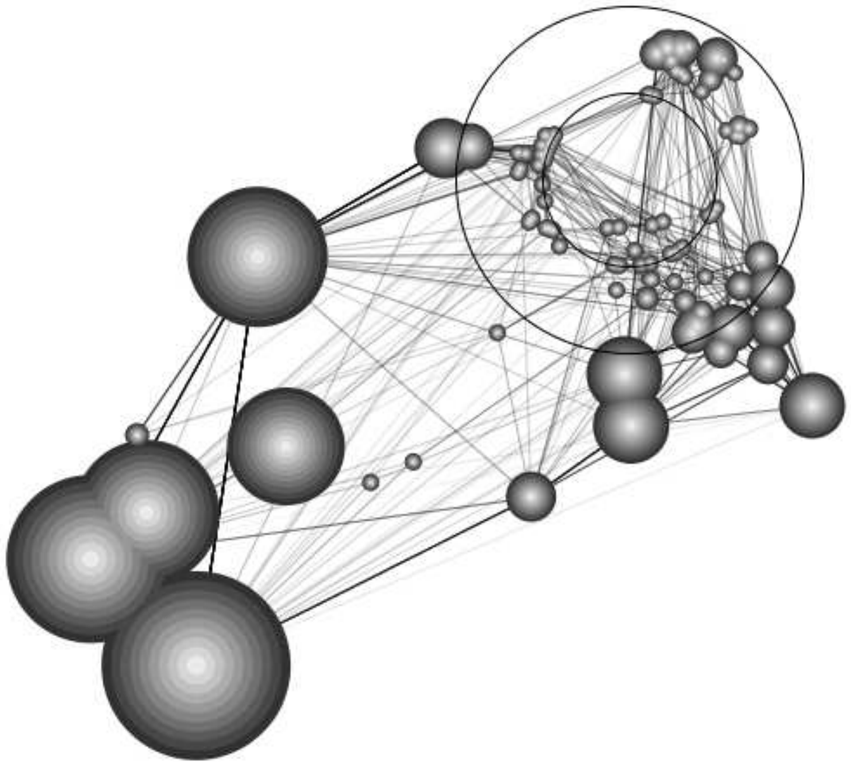


Image: /a/gort/homes/gort/treinish/dx_demos/v2/working/UI-ozone.net
File Execute Windows Connection Options Help
Visual Program Editor: /a/gort/homes/gort/tre
File Edit Execute Windows Connection Options Help
Categories:
Debugging
Import and Export
Interactor
Macros
Miscellaneous
Realization
Rendering
Special
Splines
Structuring
Transformation
{ ALL }
Cartography Tools:
Asteroid
AzimuthalEquidistant
CylindricalEquidistant
Globe
HammerAitoff
MapImageContourSurfa
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
dit Execute Options Help
max
650.00000
Hue
Saturation
Value
Opacity
650
600
511.617
475.61
424.775
382.413
343.582
306.868
275.802
251.091
224.262
200.963
159.307
125.417
100
min max min
r g b r min max min max min max
100.00000
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

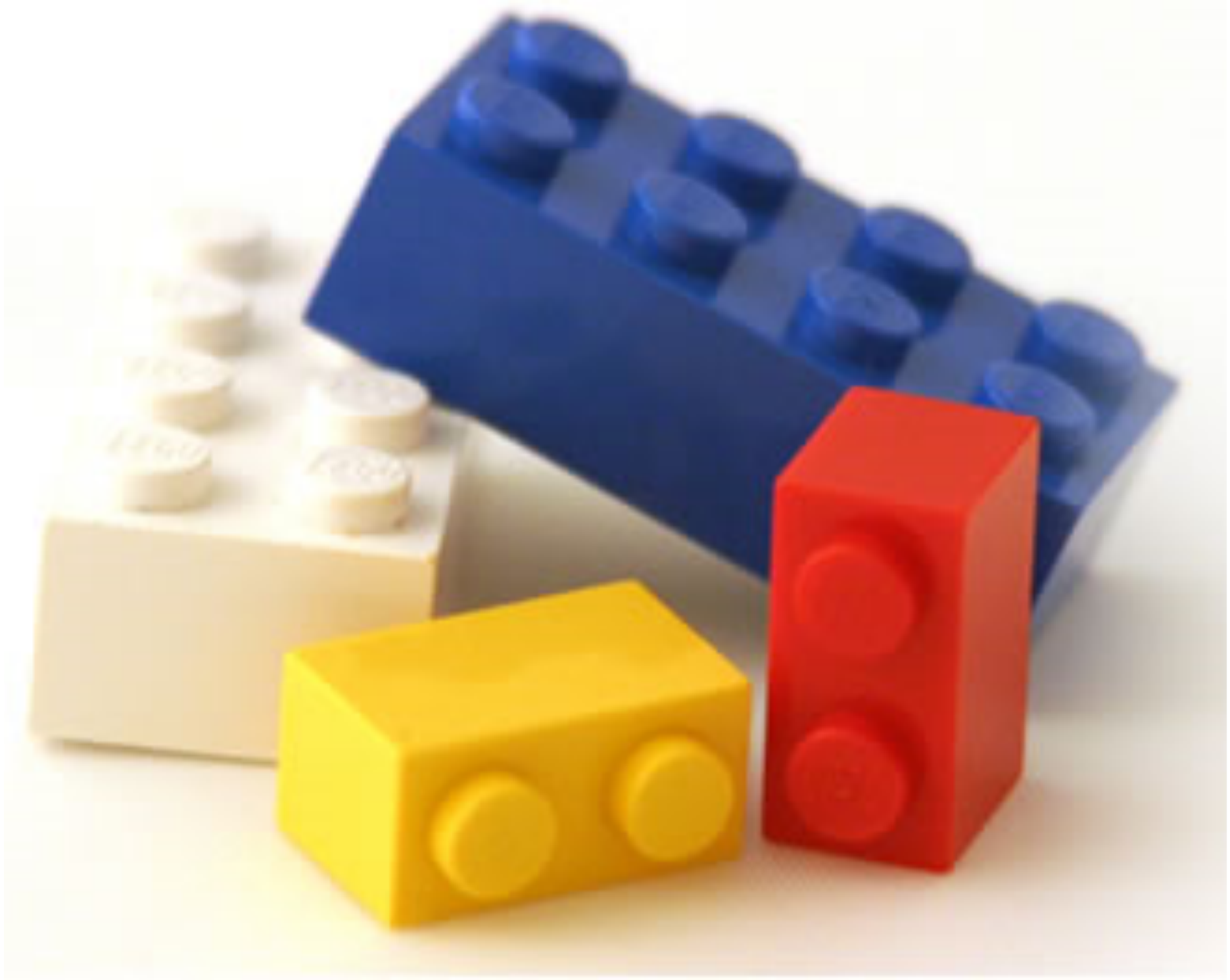


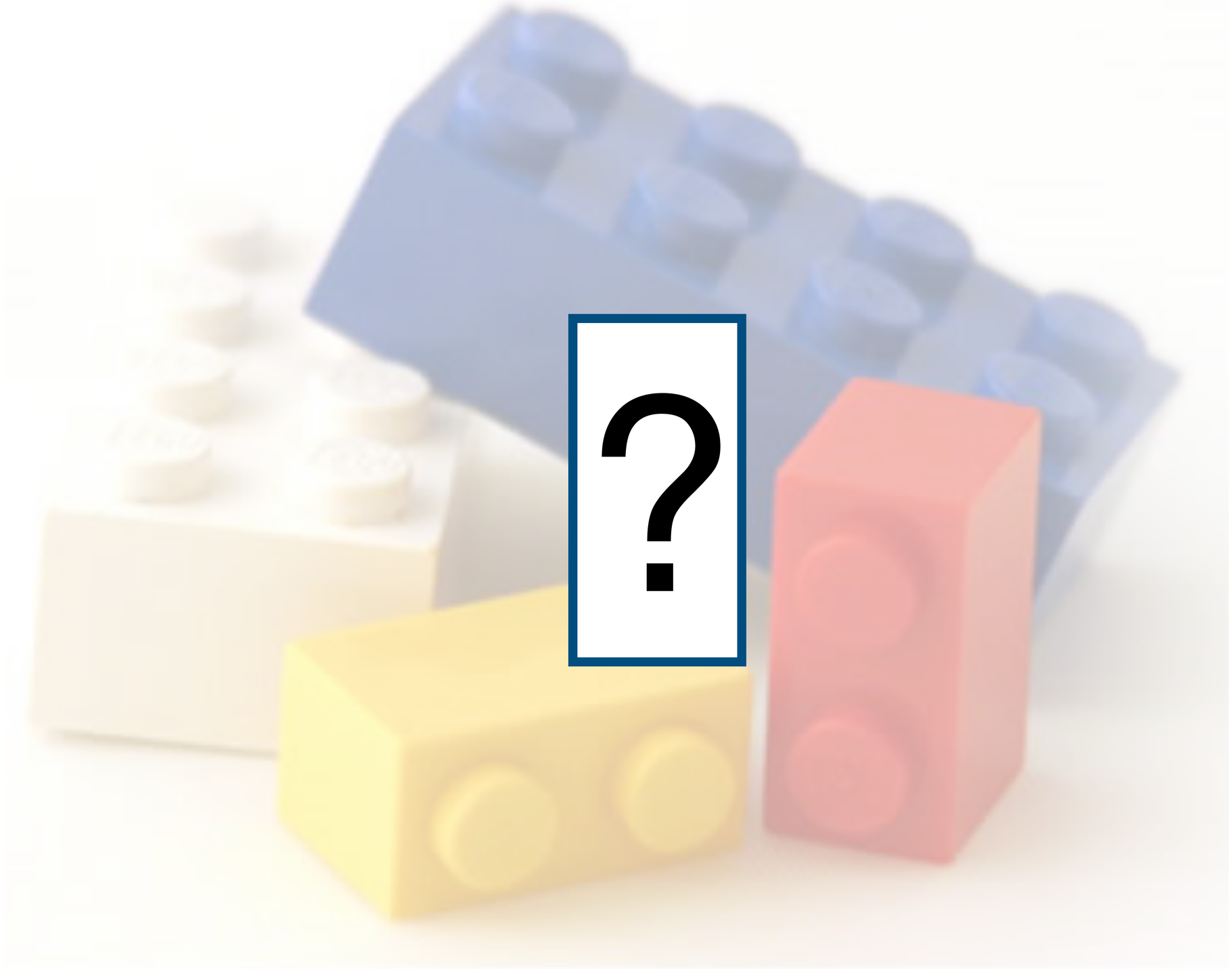
```
// initialize action lists
ActionList layout = new ActionList(registry);
layout.add(new TreeFilter(true));
layout.add(new RadialTreeLayout());
layout.add(new ColorFunction());
```

Users can define their own layouts, etc.

Heer, Jeffrey, Stuart K. Card, and James A. Landay. "Prefuse: a toolkit for interactive information visualization." *2005*


concept
idea
law
plan
kind
principle
value
Muslim Separatists
MNLF
MNLF Renegade
Moro Islamic
Philippines
Vicki
in the
just
banff
Mark Frevert
Simone La
Paula Rieker
James Derrick
Sue Ford
Kenneth Lay
Sherri Sera
Maureen Mcvicker
Rosalee Fleming
Karen Denne
Joannie Williamson
Leonardo Pacheco
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
CI13R
synaptic transmission
CI6R
membrane
CI24R
CI5R
Title Nagarjuna
URL http://en.wikipedia.org/
$1,000,000
$0
NY OK PA RI SD TX VA VT
Joylette
Alan
Chris
Amanda
Shane
Rikard
David
Justin
Shane
Micaela
Steven
Matt
Fernando
WA
OR
MT
ID
WY
ND
SD
MN
WI
MI
NE
IA
UT
NV
CO
KS
MO
IL
IN
OH
KY
CA
VT
ME
NH
NY
PA
NJ
MD
VA
RI
Pam
Scott
Irina
Danyel
Scott
Ideology
Science
Humanities
Technology
Knowledge
Historical
The Millenium
Events
Generic
Civilizations
Individuals
People
S. People
Organizations
Categories
Mohamed Atta
Hani Hanjour
West Hollywood
Beverly Hills
Los Angeles
Culver City
Inglewood
South Pasadena
Alhambra
San Gabriel
Montebello
Huntington Park
South Gate





Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Canvas, OpenGL, Processing, SVG

Chart Typologies

Excel, Google Charts

Component Architectures

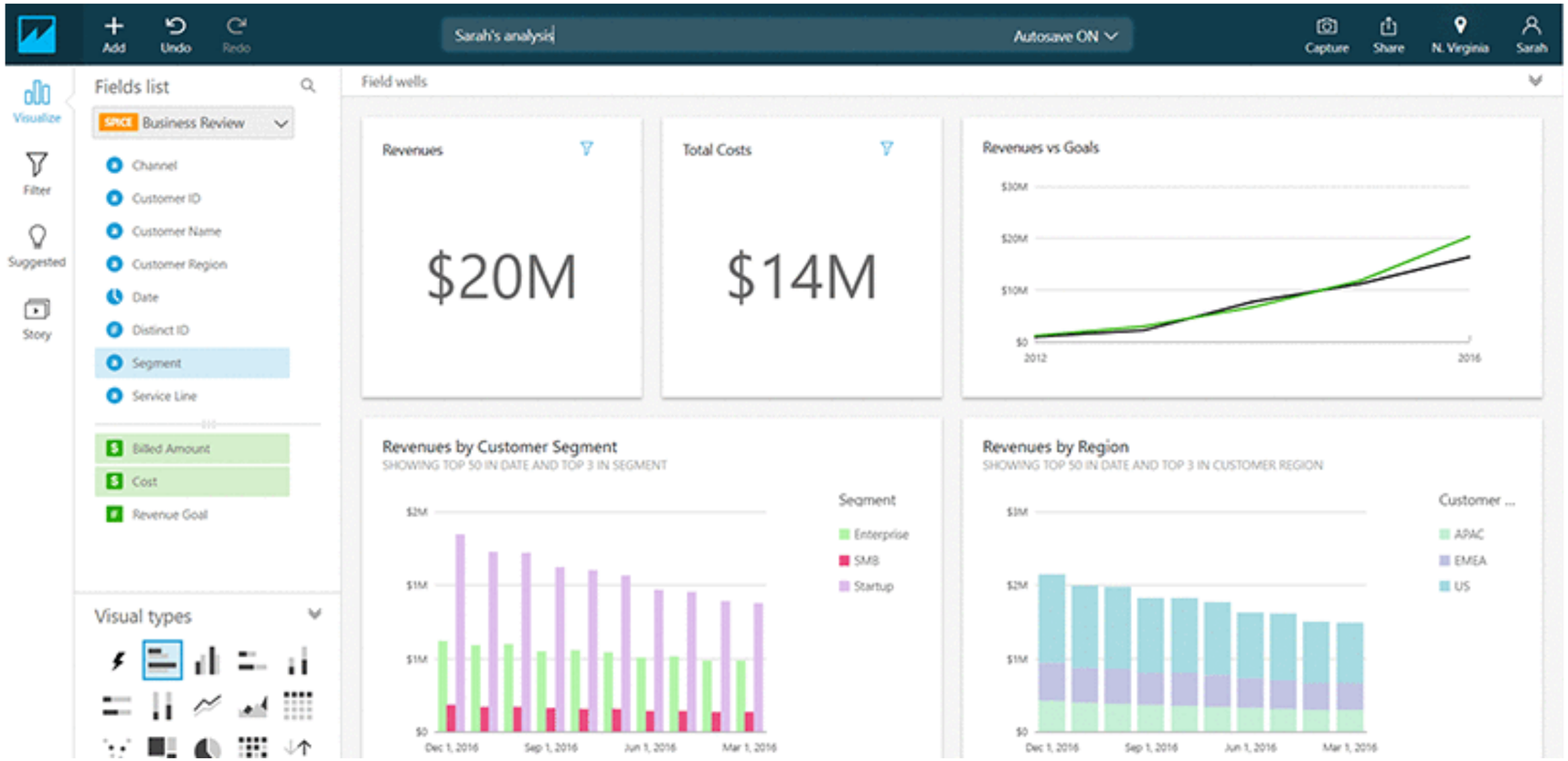
Prefuse, Flare, Improvise, VTK

Graphics APIs

Canvas, OpenGL, Processing, SVG


2D
3D
Interactive

AWS QuickSight



Data Sets : State Quick Facts

Uploaded By: zinggoat
Data Source: US Census Bureau
Description:
Tags: people census

Created at: Friday May 18, 3:08 PM

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
14	Indiana	6271973	0.03	6080485	0.1	0.07	0.26	0.12
...	...	...	...	...	...	...	...	...

Choosing a visualization type for State Quick Facts

Analyze a text

Tag Cloud

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

Learn more

Wordle

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

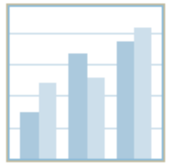
Learn more

Word Tree

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

Learn more

Compare a set of values



Bar Chart

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

Learn more

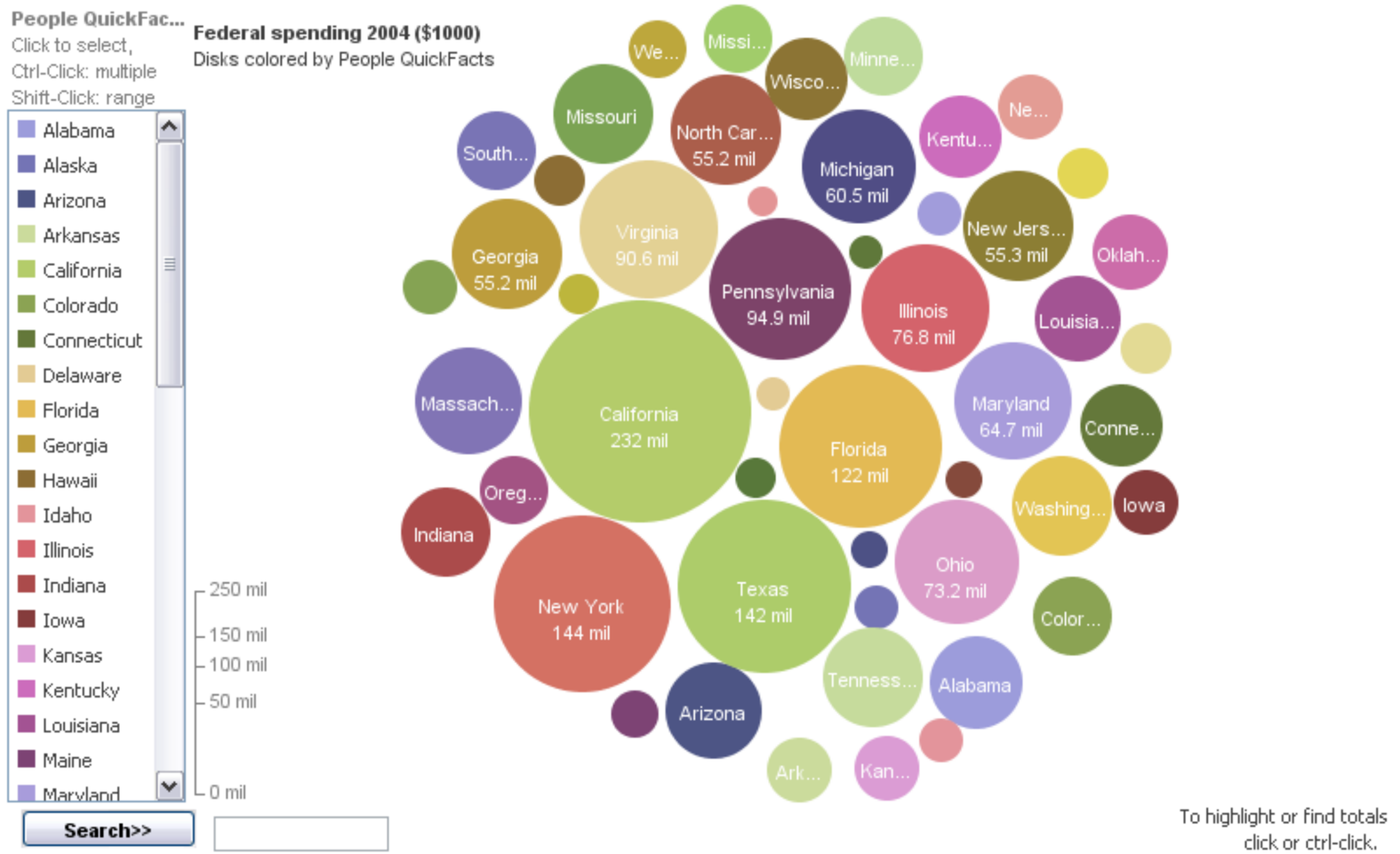
Block Histogram

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

Learn more

Visualizations : Federal Spending by State, 2004

Creator: Anonymous
Tags: census people



MAD LIBS®

MY MUSIC LESSON

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

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

Leland Wilkinson
The Grammar of Graphics, 1999

Chart Typologies

Excel, Google Charts

Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Canvas, OpenGL, Processing, SVG

Chart Typologies

Excel, Google Charts

Visual Analysis Grammars

VizQL, ggplot2, Vega-Lite, Altair

Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Canvas, OpenGL, Processing, SVG

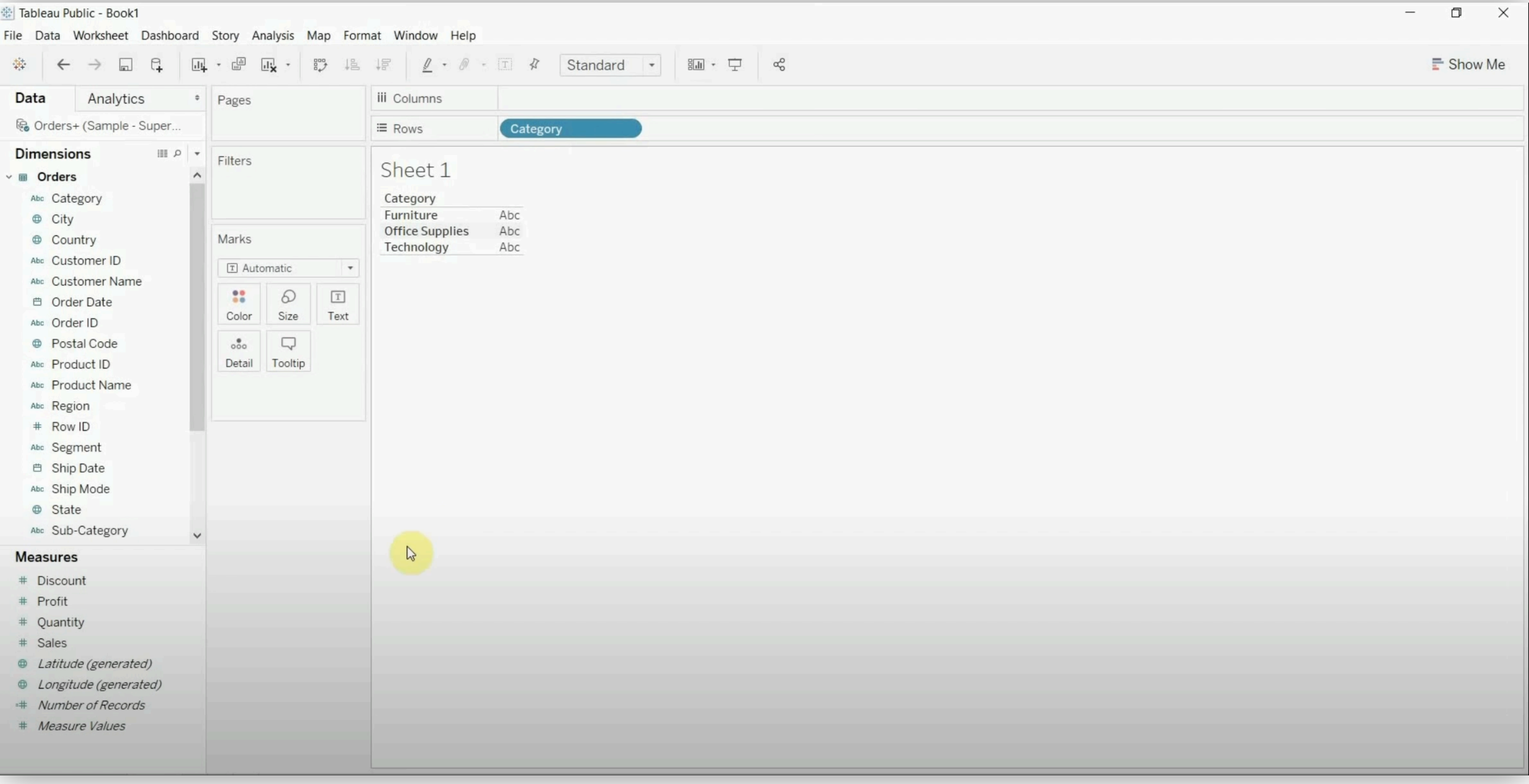
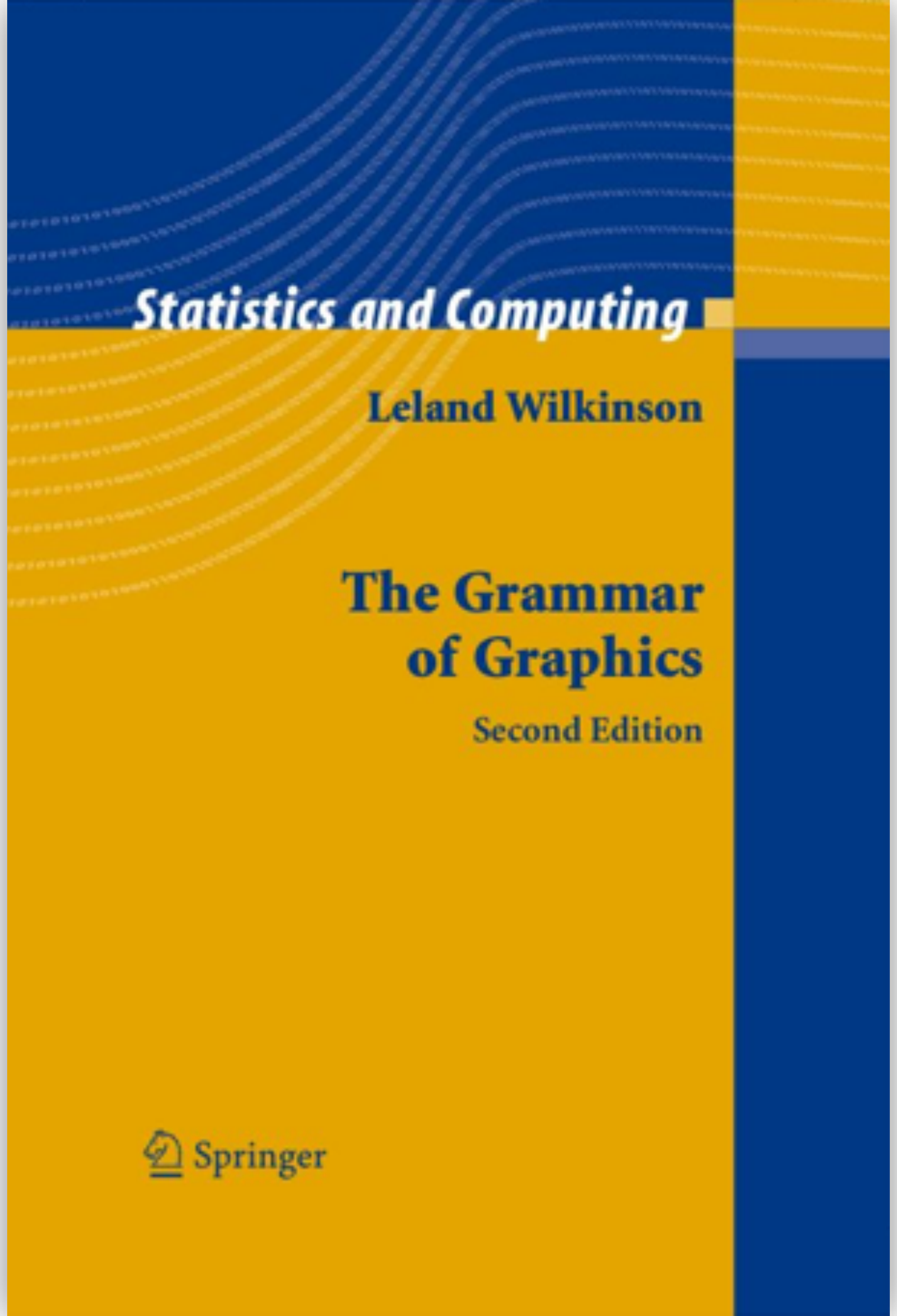
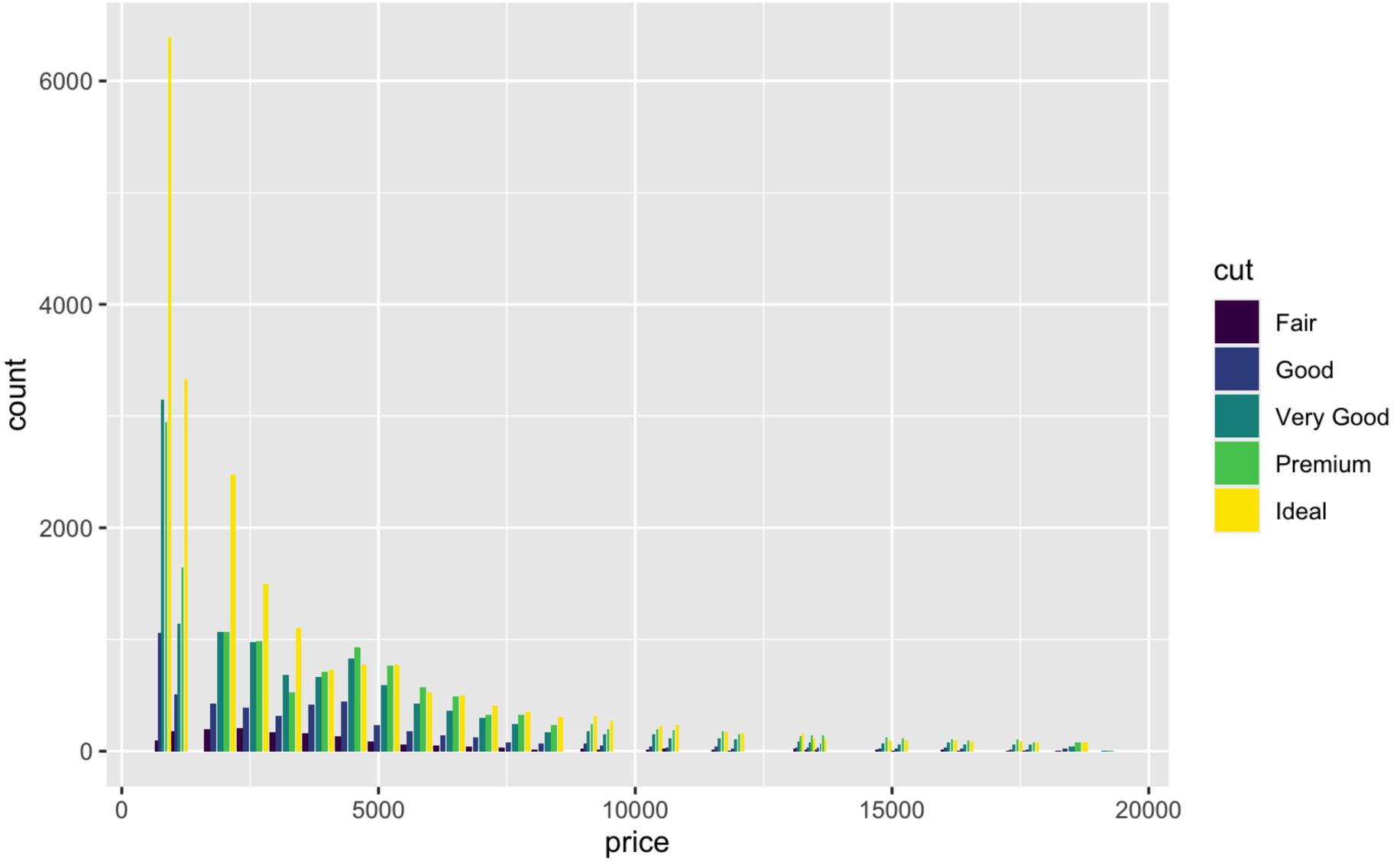
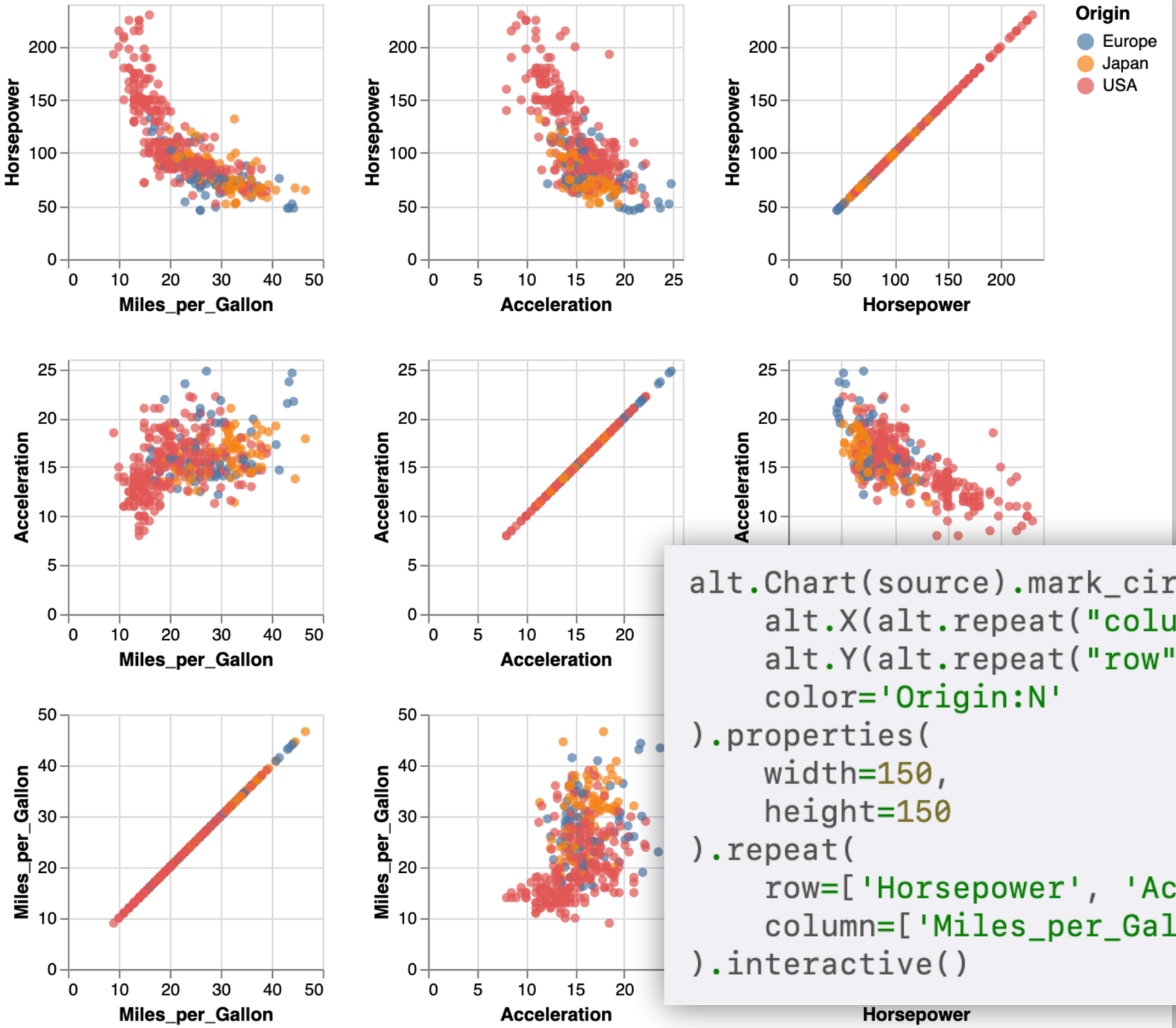


Tableau Public - Book1
File Data Worksheet Dashboard Story Analysis Map Format Window Help
Standard
Show Me
Data
Analytics
Orders+ (Sample - Super...
Dimensions
Orders
Category
City
Country
Customer ID
Customer Name
Order Date
Order ID
Postal Code
Product ID
Product Name
Region
Row ID
Segment
Ship Date
Ship Mode
State
Sub-Category
Measures
Discount
Profit
Quantity
Sales
Latitude (generated)
Longitude (generated)
Number of Records
Measure Values
Pages
Filters
Marks
Automatic
Color
Size
Text
Detail
Tooltip
Columns
Rows
Category
Sheet 1
Category
Furniture Abc
Office Supplies Abc
Technology Abc


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




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




```
alt.Chart(source).mark_circle().encode(
    alt.X(alt.repeat("column"), type='quantitative'),
    alt.Y(alt.repeat("row"), type='quantitative'),
    color='Origin:N'
).properties(
    width=150,
    height=150
).repeat(
    row=['Horsepower', 'Acceleration', 'Miles_per_Gallon'],
    column=['Miles_per_Gallon', 'Acceleration', 'Horsepower']
).interactive()
```

Chart Typologies

Excel, Google Charts

Visual Analysis Grammars

VizQL, ggplot2, Vega-Lite, Altair

Component Architectures

Prefuse, Flare, Improvise, VTK

Graphics APIs

Canvas, OpenGL, Processing, SVG

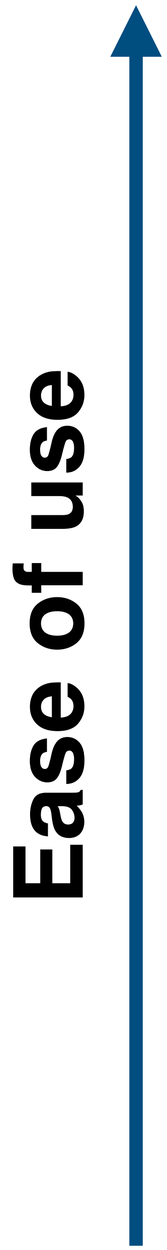


Chart Typologies
Excel, Google Charts

Visual Analysis Grammars
VizQL, ggplot2, Vega-Lite, Altair

Component Architectures
Prefuse, Flare, Improvise, VTK

Graphics APIs
Canvas, OpenGL, Processing, SVG

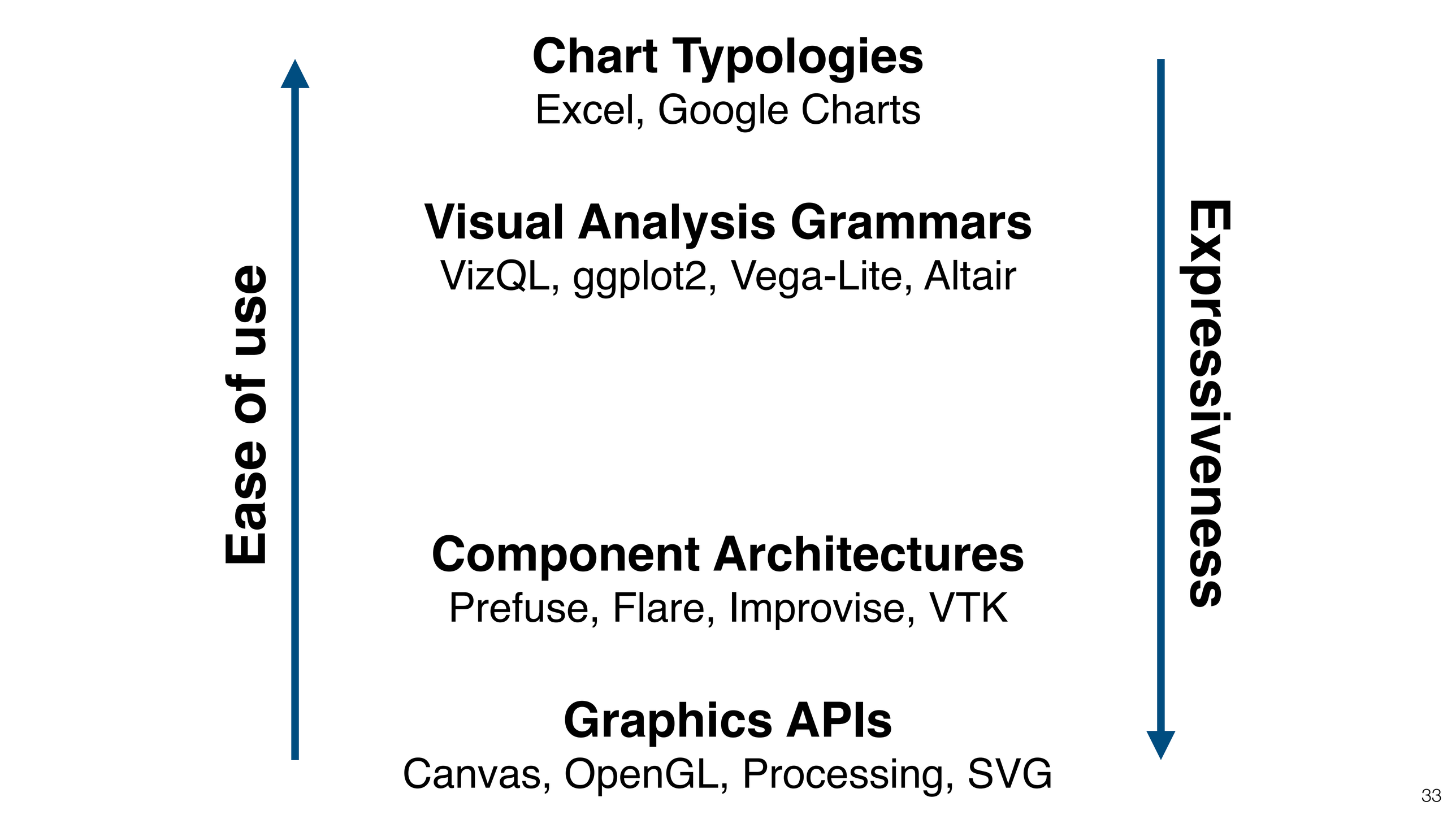


Chart Typologies
Excel, Google Charts
Visual Analysis Grammars
VizQL, ggplot2, Vega-Lite, Altair
Component Architectures
Prefuse, Flare, Improvise, VTK
Graphics APIs
Canvas, OpenGL, Processing, SVG
Ease of use
Expressiveness

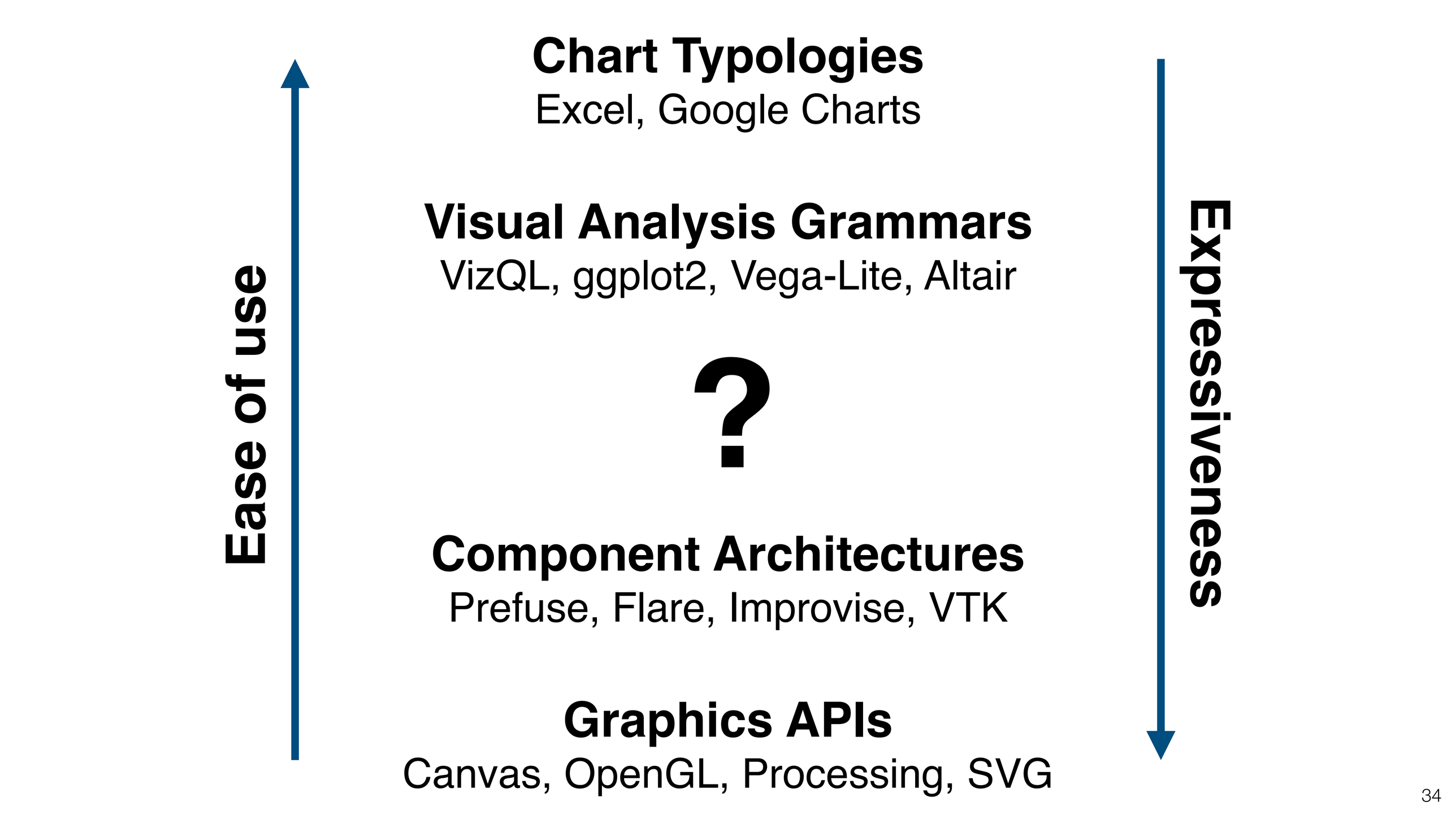


Chart Typologies
Excel, Google Charts
Visual Analysis Grammars
VizQL, ggplot2, Vega-Lite, Altair
?
Component Architectures
Prefuse, Flare, Improvise, VTK
Graphics APIs
Canvas, OpenGL, Processing, SVG
Ease of use
Expressiveness

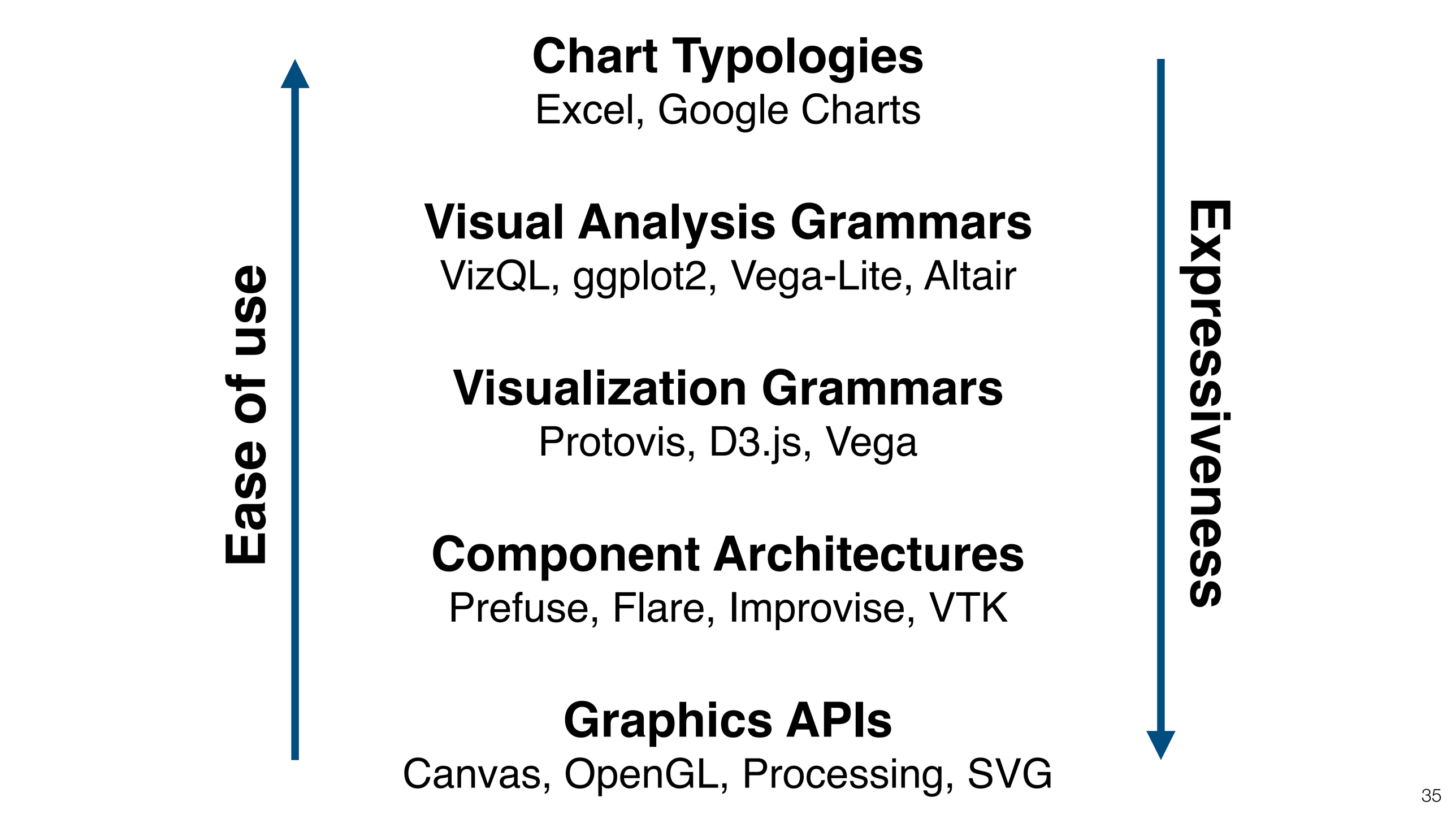


Chart Typologies
Excel, Google Charts
Visual Analysis Grammars
VizQL, ggplot2, Vega-Lite, Altair
Visualization Grammars
Protovis, D3.js, Vega
Component Architectures
Prefuse, Flare, Improvise, VTK
Graphics APIs
Canvas, OpenGL, Processing, SVG
Ease of use
Expressiveness

Visualization Grammar

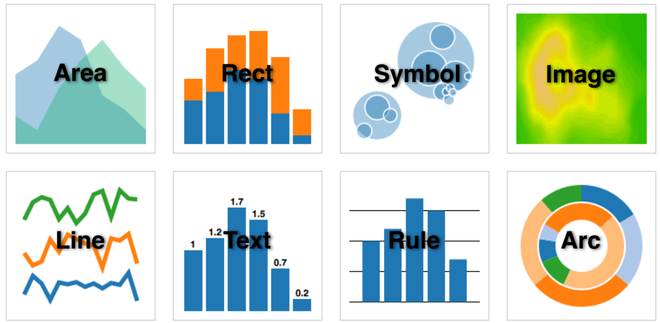
Data Input data to visualize

Transforms Group, aggregate, layout

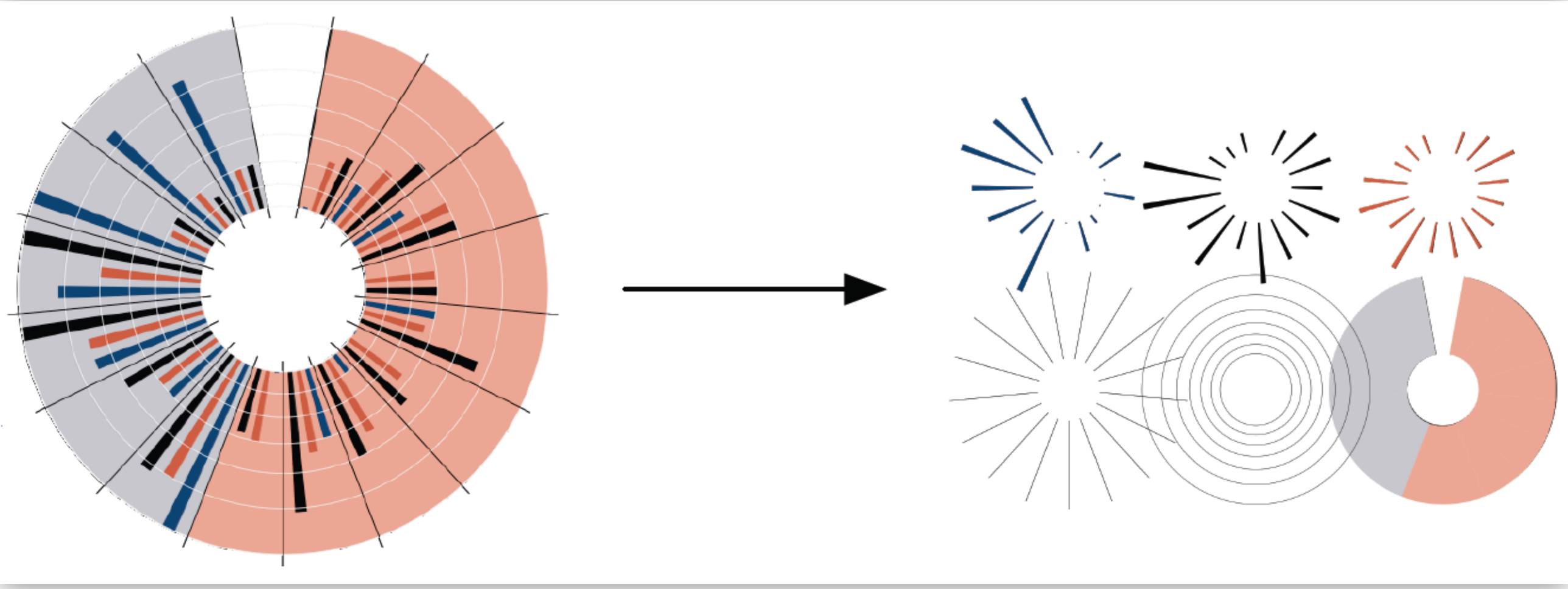
Scales Map data values to visual values

Guides Axes & legends

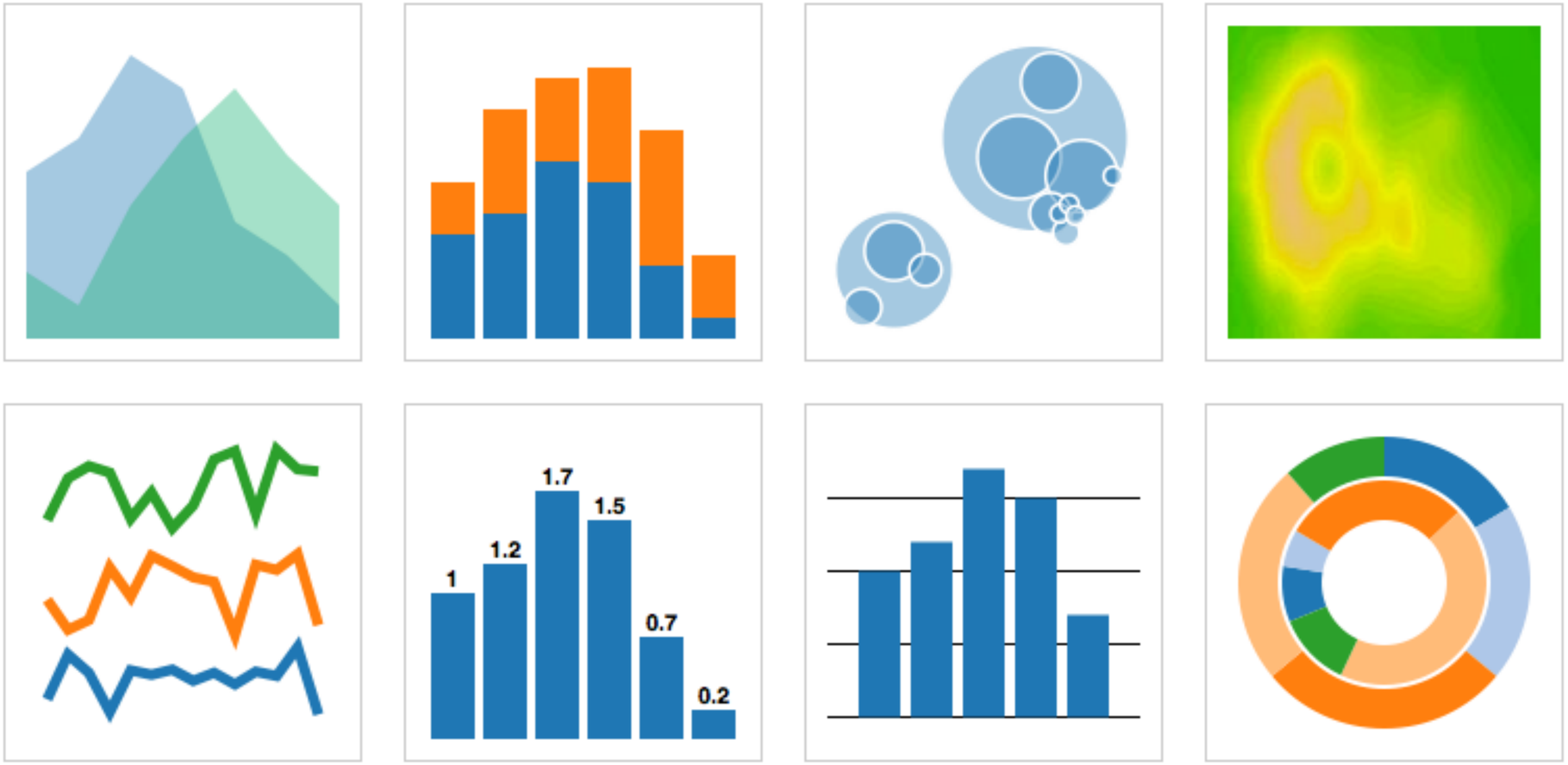
Marks Data-representative graphics



Protovis (D3 predecessor)



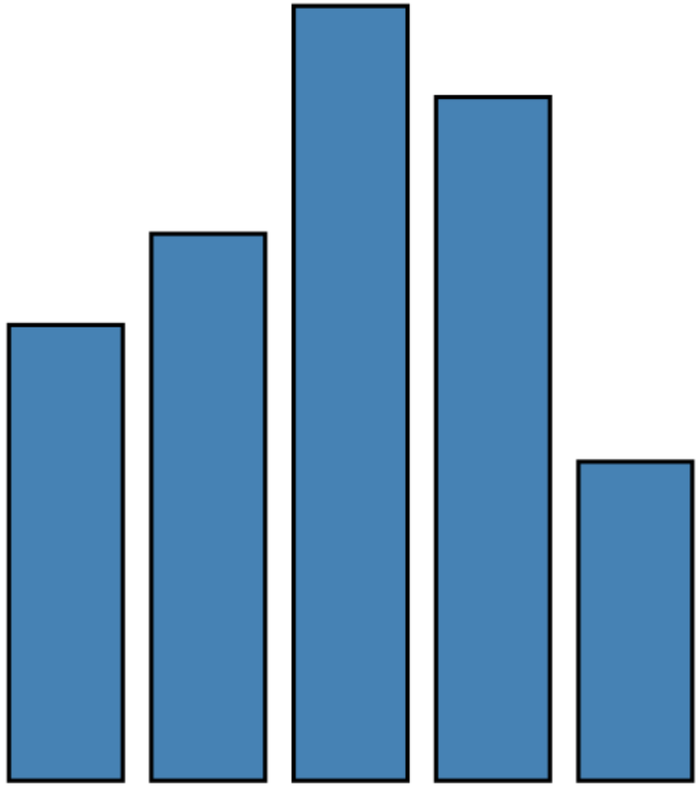
Bostock, Michael, and Jeffrey Heer. "Protovis: A graphical toolkit for visualization." 2009



Marks

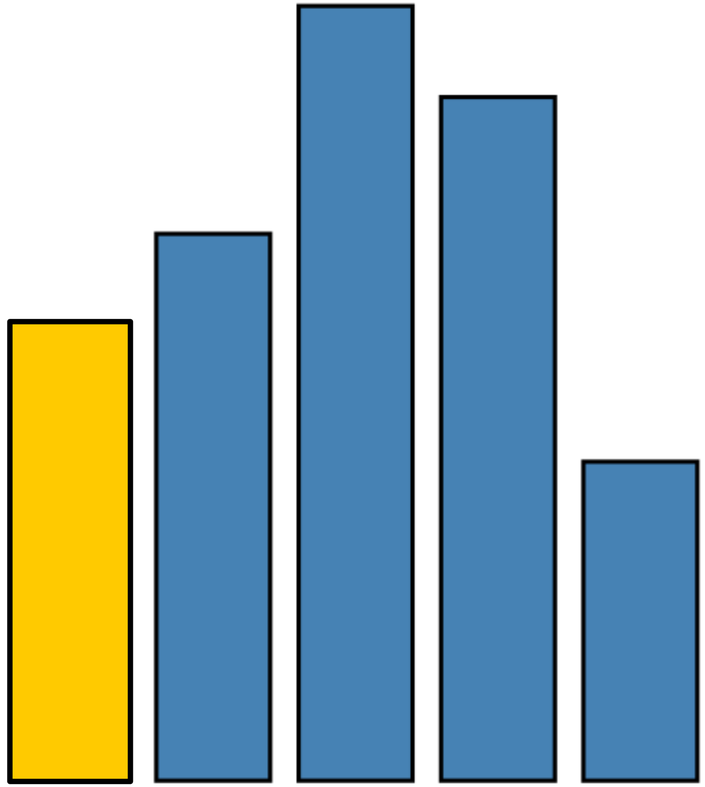
MARK	$\lambda : D \rightarrow R$
data	λ
visible	λ
left	λ
bottom	λ
width	λ
height	λ
fillStyle	λ
strokeStyle	λ
lineWidth	λ
...	λ

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				
...	...				



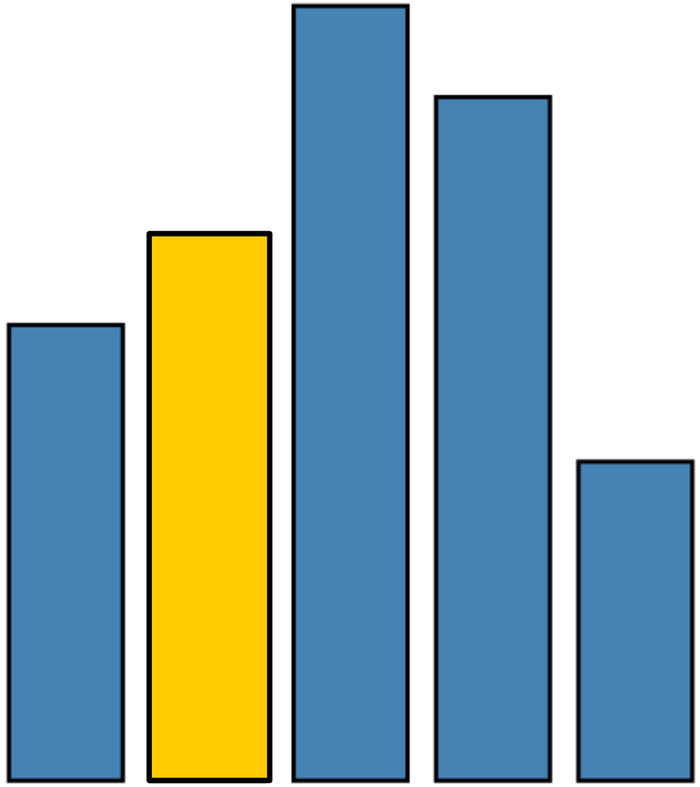
RECT $\lambda : D \rightarrow R$

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				
...	...				



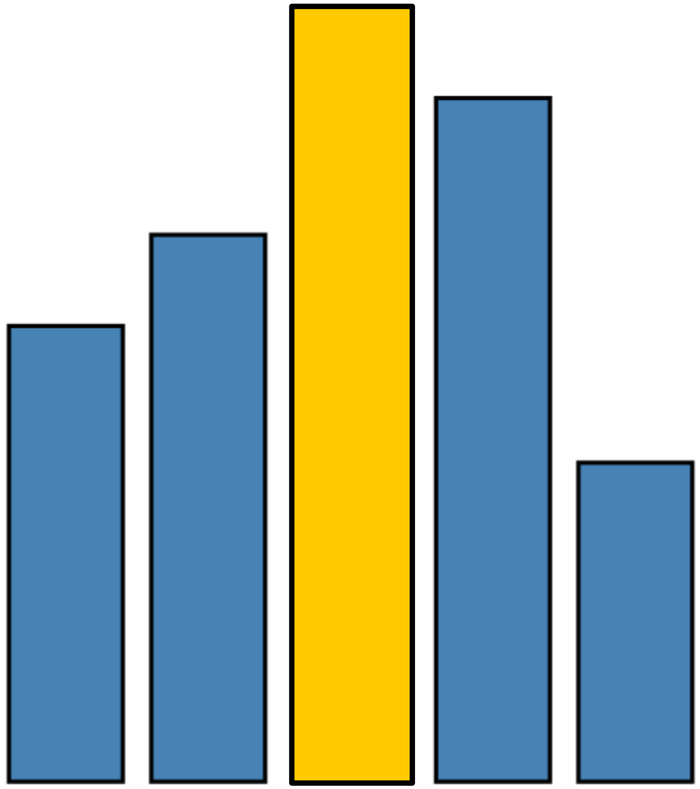
RECT $\lambda : D \rightarrow R$

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				
...	...				



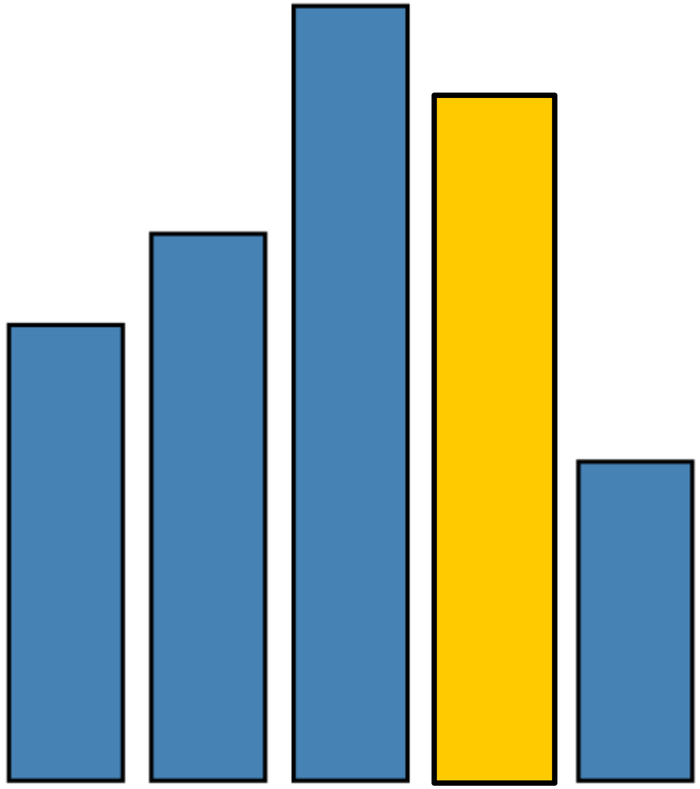
RECT $\lambda : D \rightarrow R$

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				
...	...				



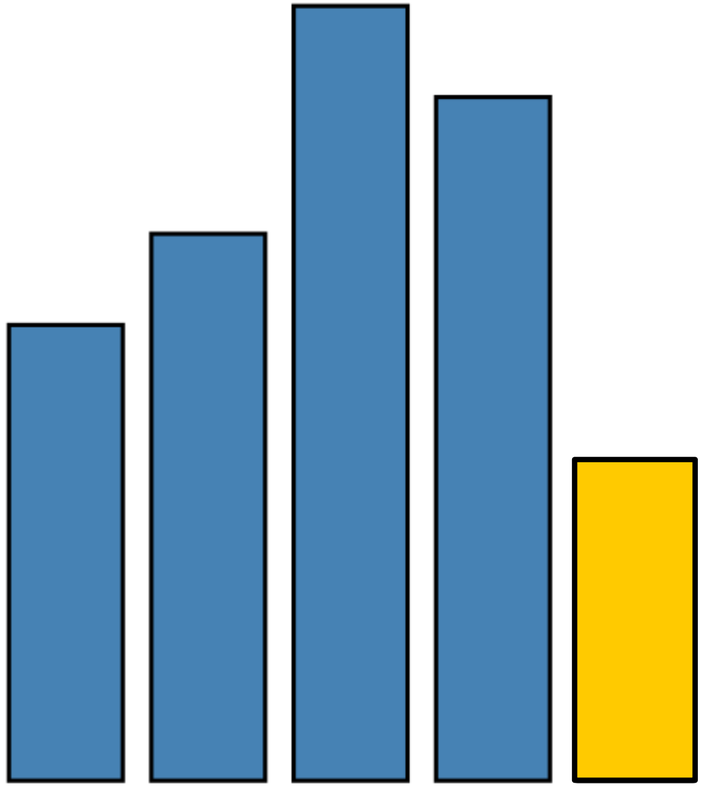
RECT $\lambda : D \rightarrow R$

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				
...	...				

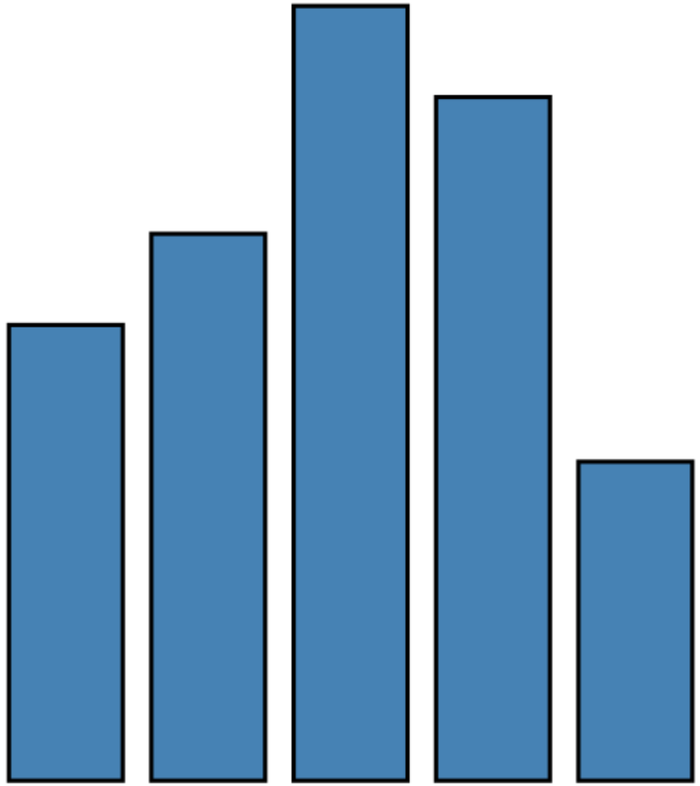


RECT $\lambda : D \rightarrow R$

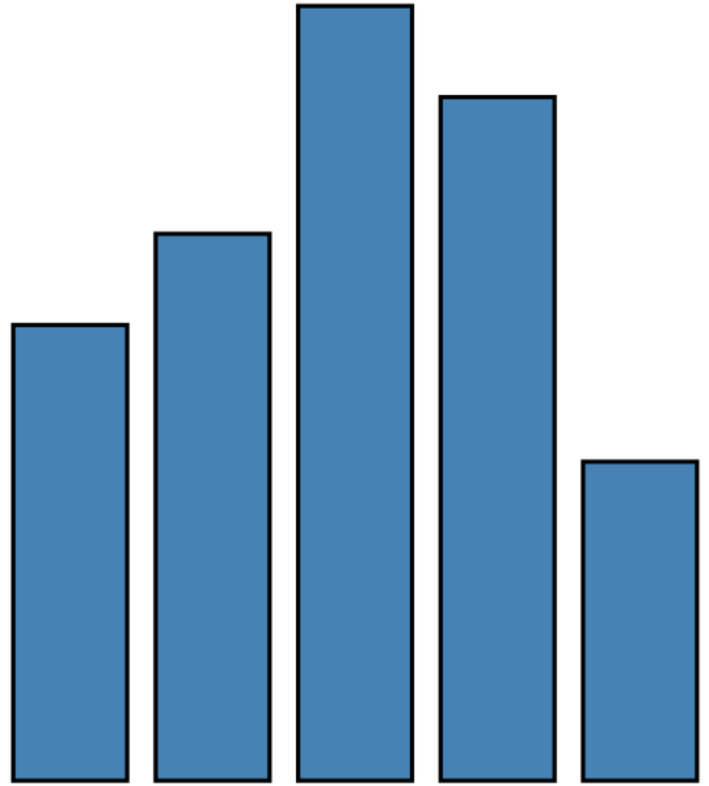
data	1	1.2	1.7	1.5	0.7
visible	true				
left	4 * 25				
bottom	0				
width	20				
height	0.7 * 80				
fillStyle	blue				
strokeStyle	black				
lineWidth	1.5				
...	...				

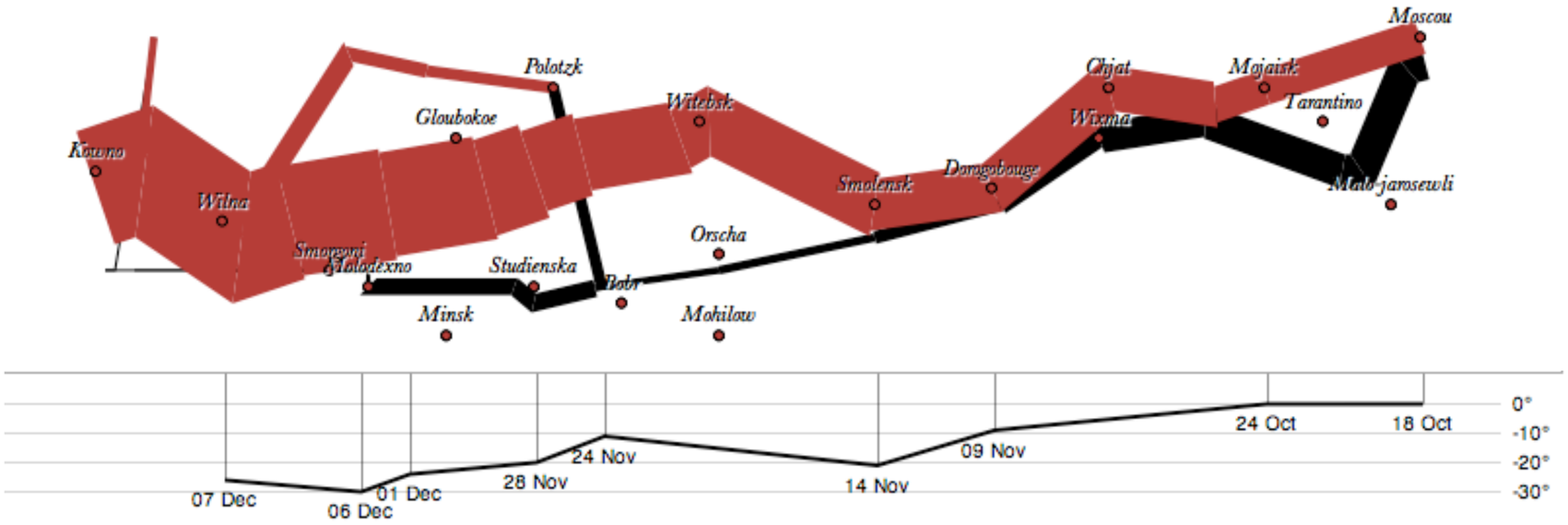


RECT	**λ : *D* → *R***				
data	1	1.2	1.7	1.5	0.7
visible	true				
left	λ: index * 25				
bottom	0				
width	20				
height	λ: datum * 80				
fillStyle	blue				
strokeStyle	black				
lineWidth	1.5				
...	...				



```
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”)
```





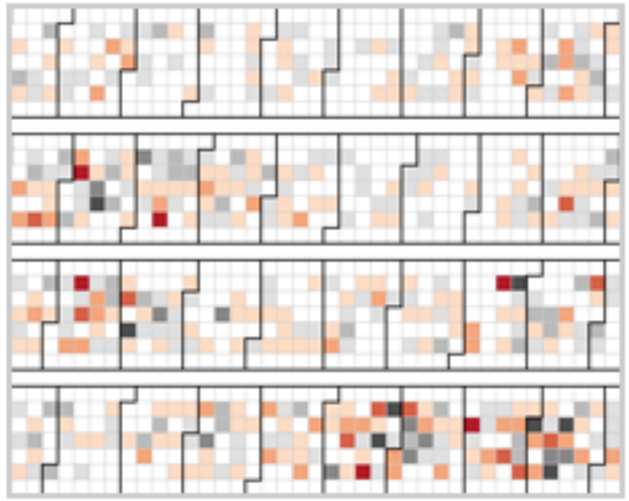
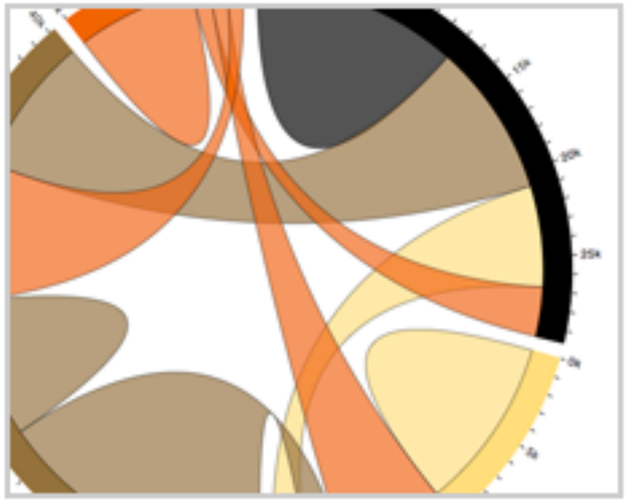
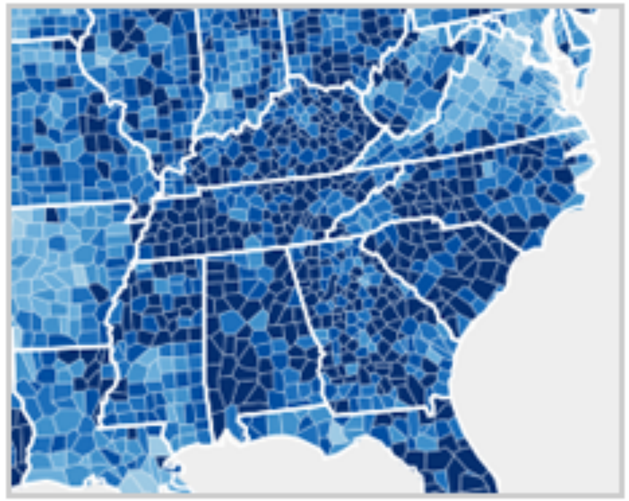
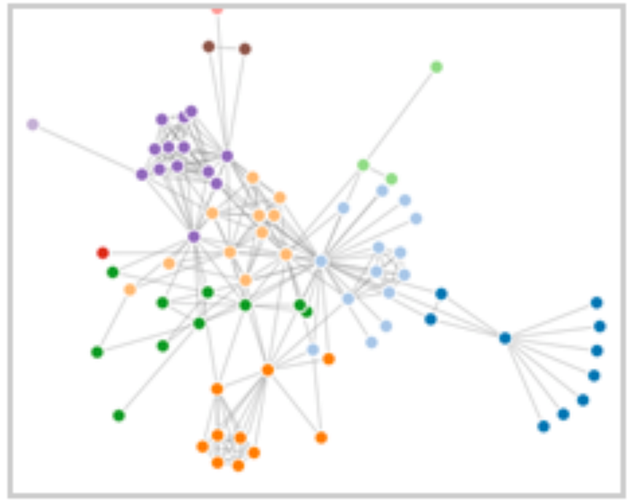
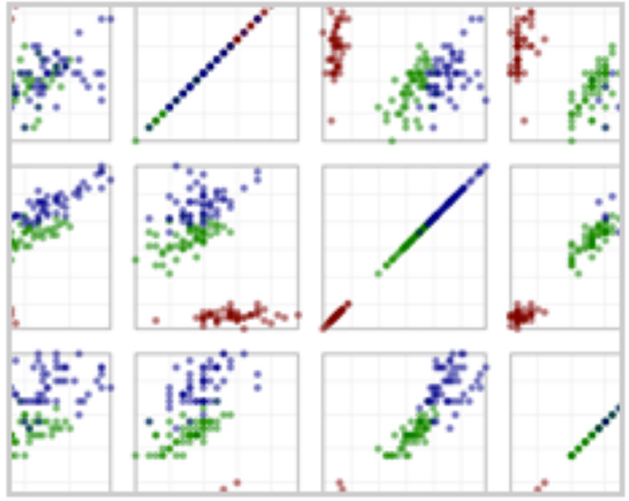
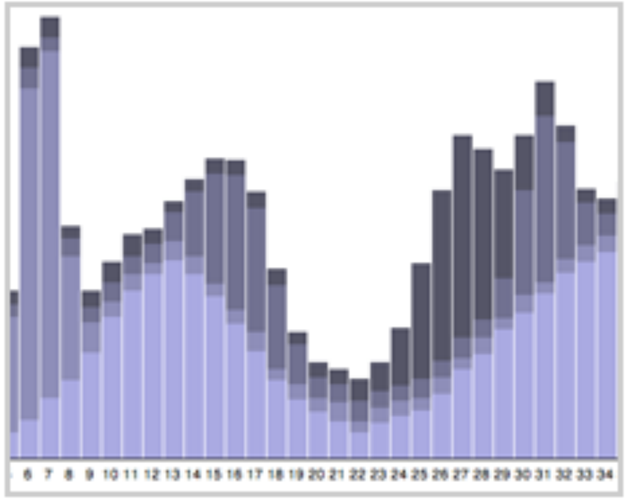

```
var army = pv.nest(napoleon.army, "dir",
"group“);
var vis = new pv.Panel();

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
   ");
```

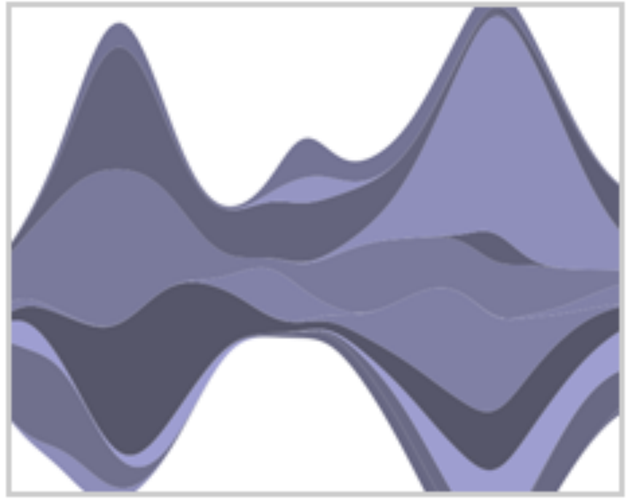
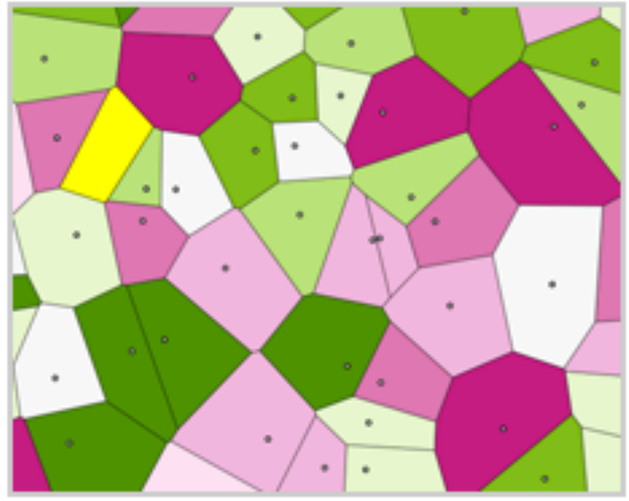
```
vis.add(pv.Rule).data([0,-10,-20,-30])
   .top((d) => 300 - 2*d -
   0.5).left(200).right(150)
   .lineWidth(1).strokeStyle("#ccc")
   .anchor("right").add(pv.Label)
```

Create axes manually

```
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");
```

d3.js: Data-Driven Documents

M Bostock, V Ogievetsky, J Heer. D³ data-driven documents. 2011

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/…

✅ 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

```
// Add a layer of dots.
svg.append("g")
    .attr("stroke", "steelblue")
    .attr("stroke-width", 1.5)
    .attr("fill", "none")
  .selectAll("circle")
  .data(cars)
  .join("circle")
    .attr("cx", d => x(d.m
    .attr("cy", d => y(d.h
    .attr("r", 3);
```

Bind one <circle> for every element in cars.

If <circle> doesn't exist, create.
If <circle> exists, update.
If data doesn't exist, remove <circle>.

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, …)

D3 Modules, Visualized

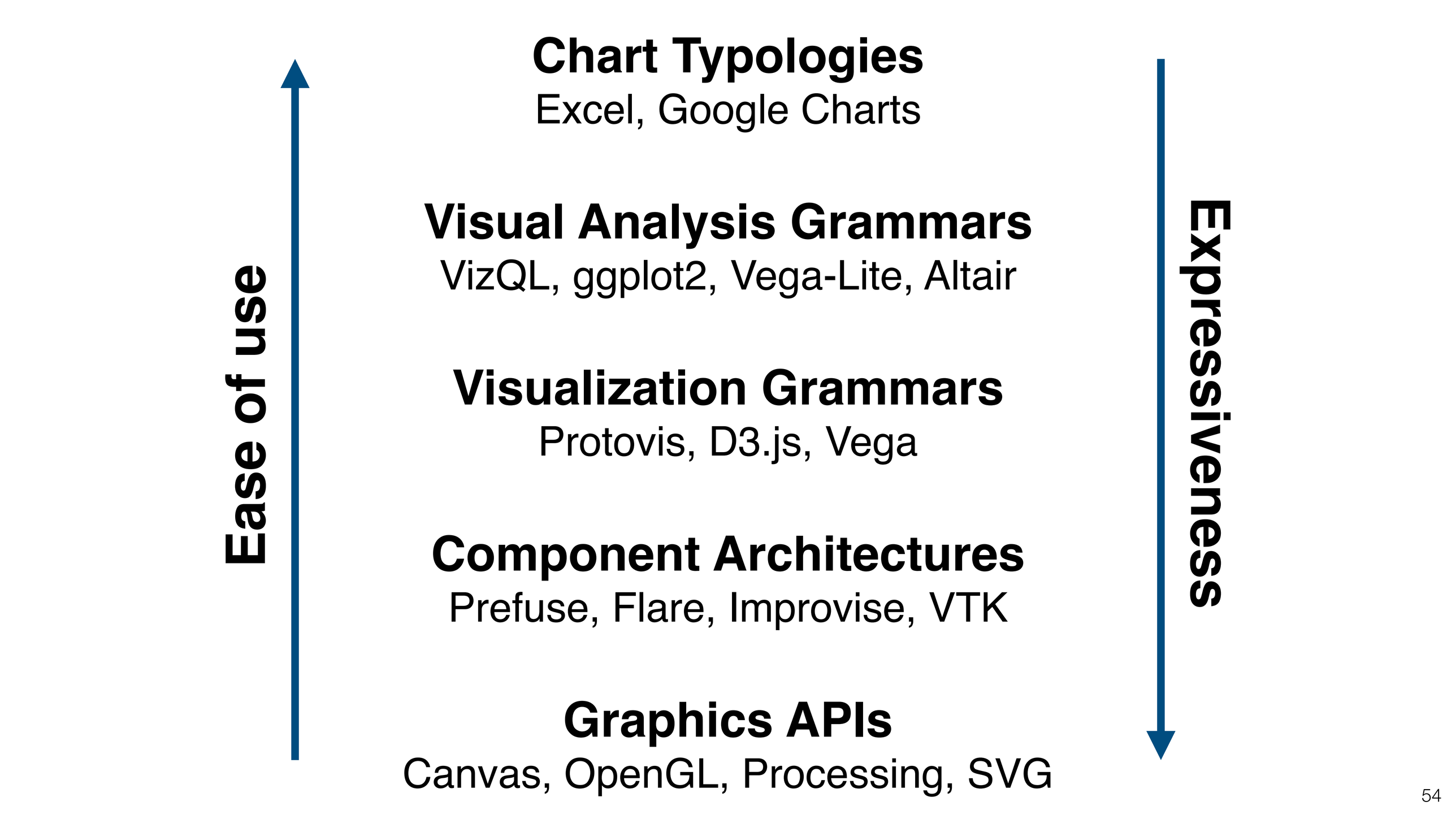


Chart Typologies
Excel, Google Charts
Visual Analysis Grammars
VizQL, ggplot2, Vega-Lite, Altair
Visualization Grammars
Protovis, D3.js, Vega
Component Architectures
Prefuse, Flare, Improvise, VTK
Graphics APIs
Canvas, OpenGL, Processing, SVG
Ease of use
Expressiveness