VISUALIZATION TOOLS AND PARADIGMS

UNIVERSITY OF CALGARY

HOW CAN WE GENERATE GRAPHICAL REPRESENTATIONS?

WHAT TOOLS ARE CURRENTLY THE MOST INTERESTING?

WHEN & WHY TO CHOOSE DIFFERENT APPROACHES?

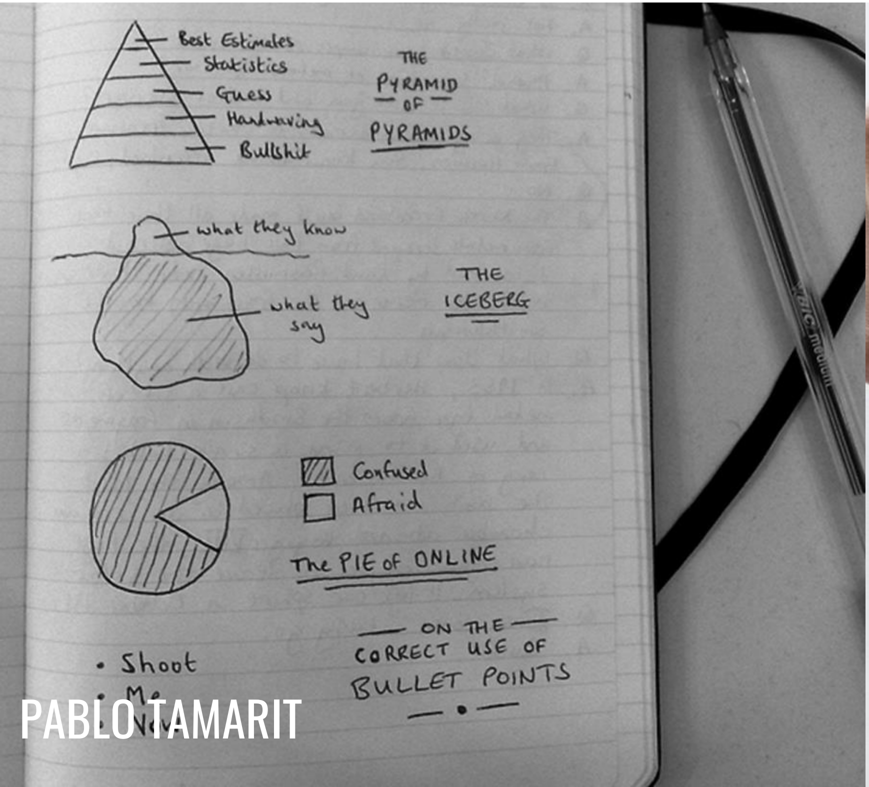
BRET VICTOR

DRAW USE CODE

DRAWING DYNAMIC VISUALIZATIONS
https://vimeo.com/66085662

DRAW

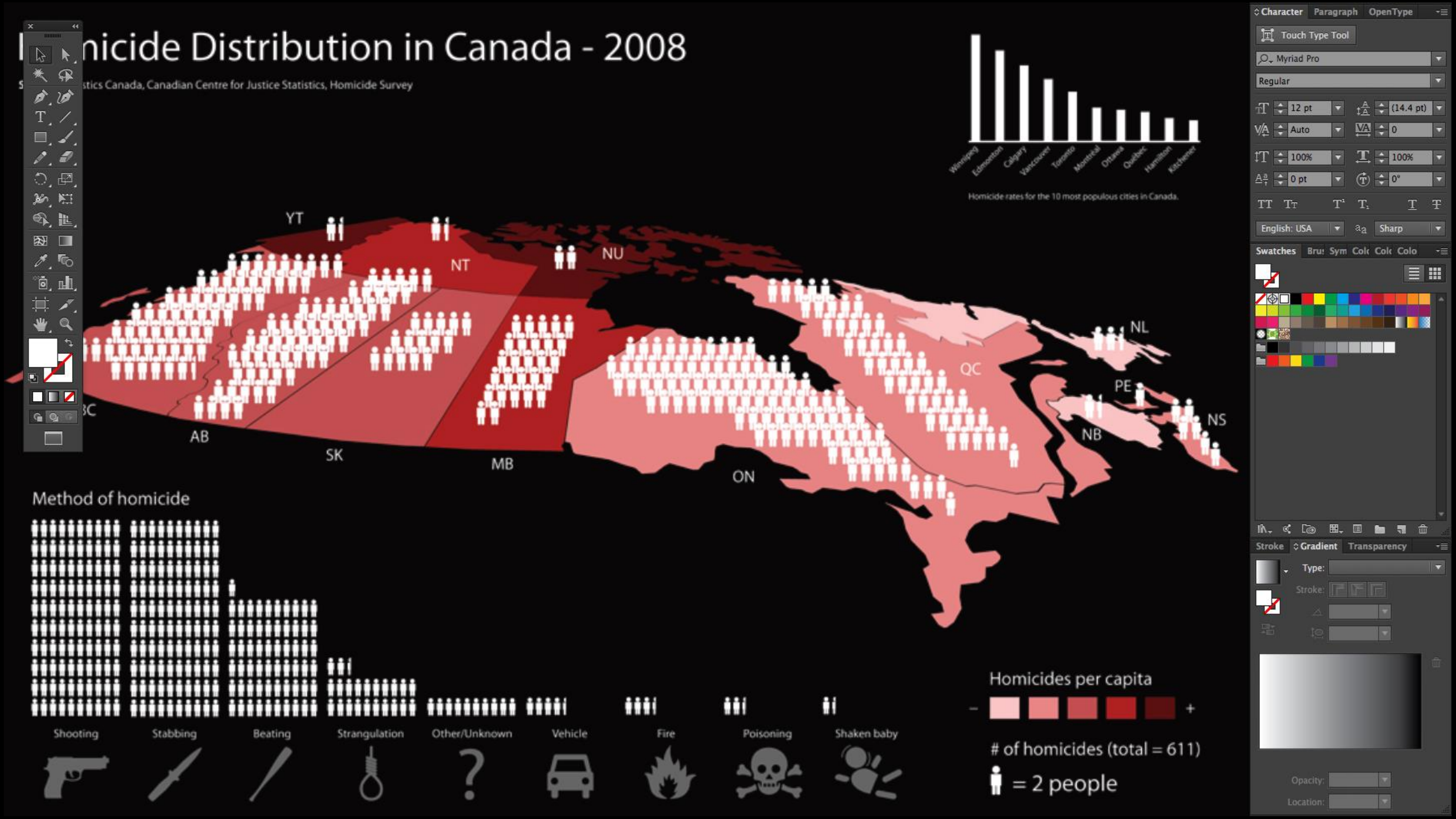
SKETCHING / CONSTRUCTING BY HAND



PABLO TAMARIT

SAMUEL HURON

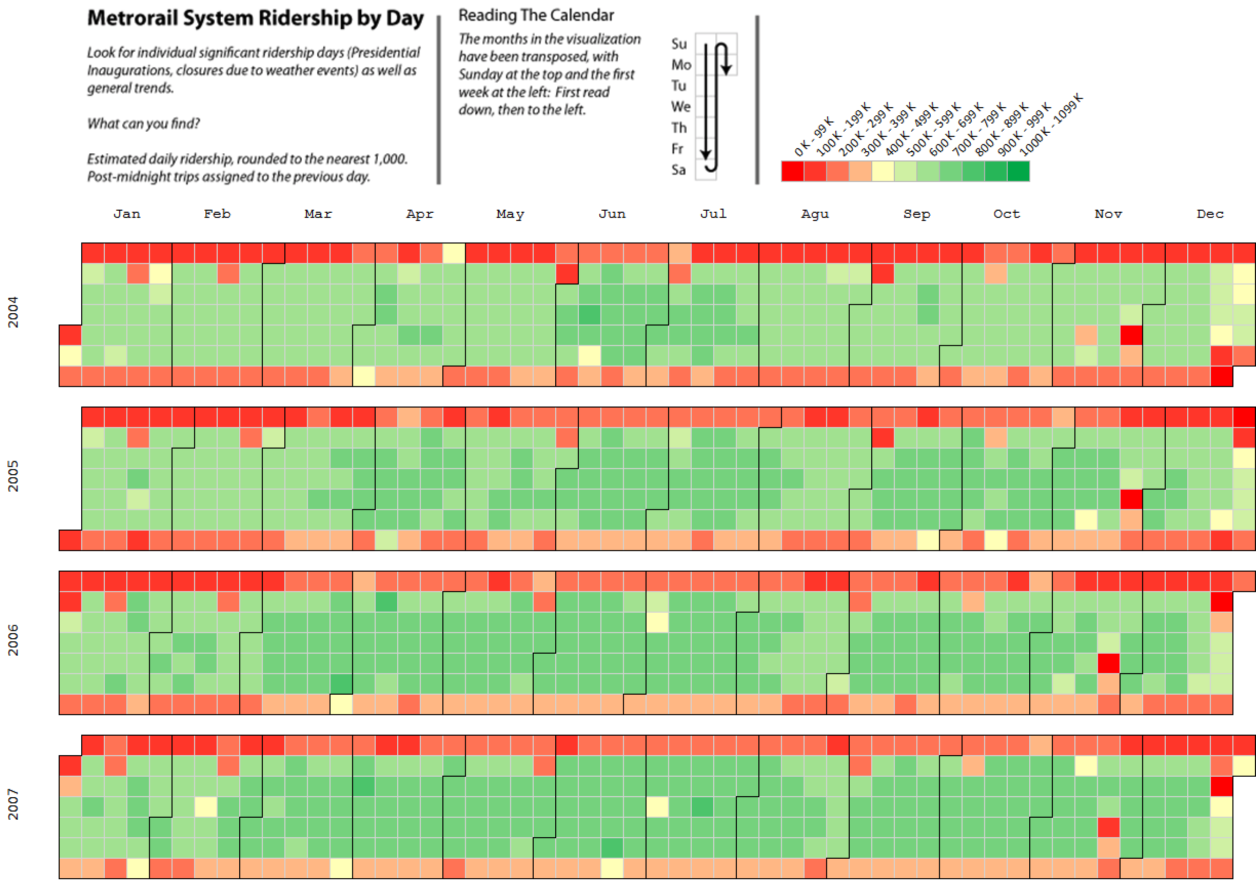
GRAPHIC DESIGN SOFTWARE (ILLUSTRATOR, PHOTOSHOP, ETC.)



http://www.kristinaneuman.com/

“DRAWING” IN EXCEL

https://sites.google.com/site/e90e50fx/home/calendar-based-heatmap-in-excel



Metrorail System Ridership by Day

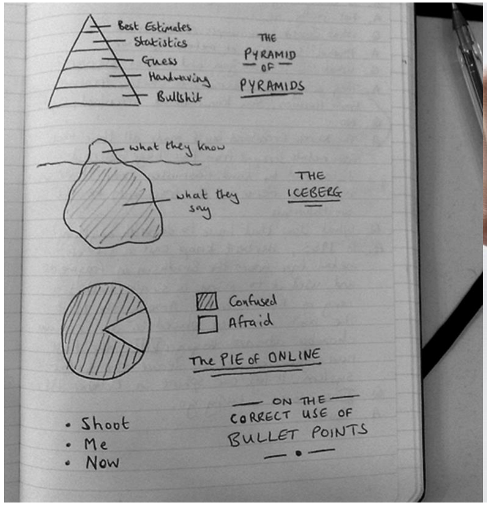
Look for individual significant ridership days (Presidential Inaugurations, closures due to weather events) as well as general trends.

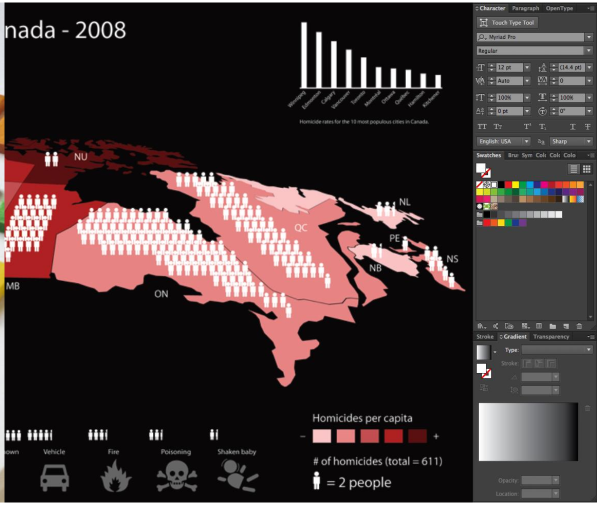
What can you find?

Estimated daily ridership, rounded to the nearest 1,000. Post-midnight trips assigned to the previous day.

Reading The Calendar

The months in the visualization have been transposed, with Sunday at the top and the first week at the left: First read down, then to the left.





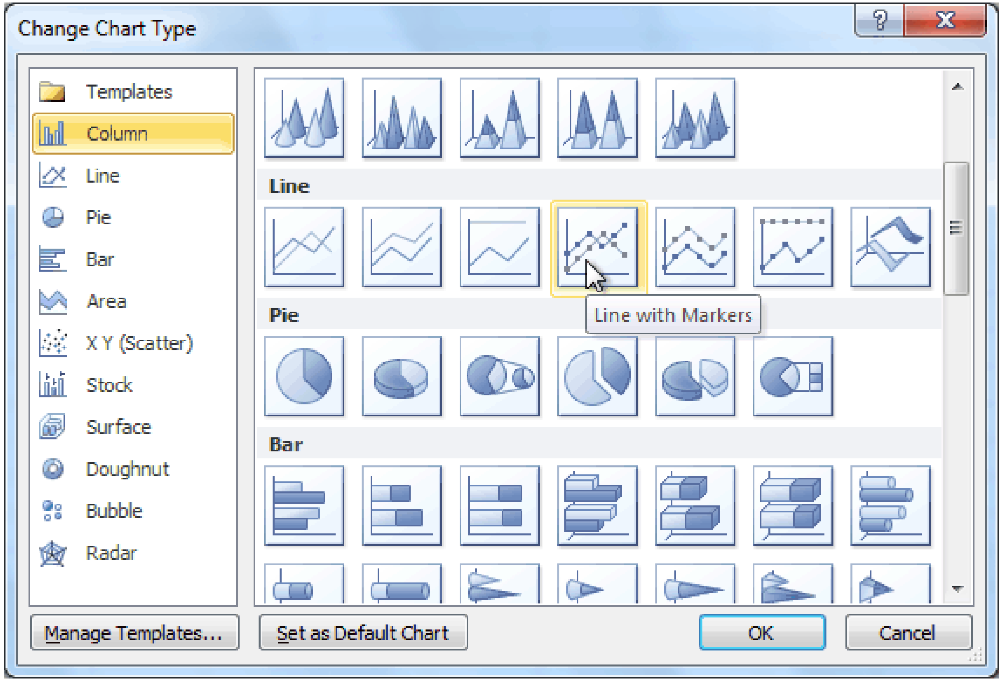
DRAW

+ FLEXIBLE & EXPRESSIVE

- SCALES BADLY
- DESIGNS ARE ONE-OFFS

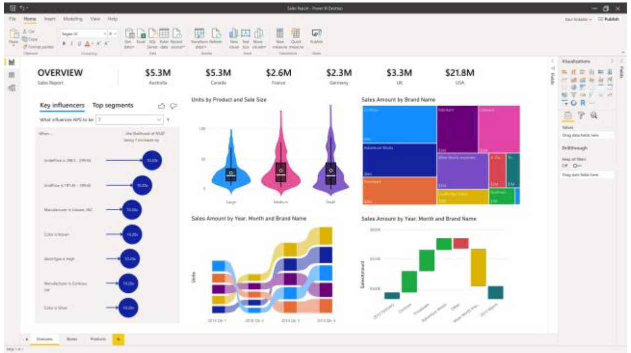
USE

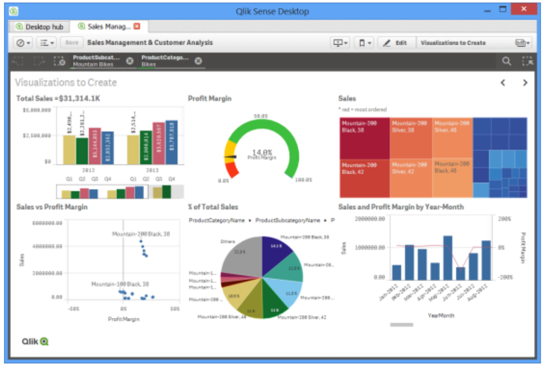
EXCEL


Change Chart Type
Templates
Column
Line
Pie
Bar
Area
X Y (Scatter)
Stock
Surface
Doughnut
Bubble
Radar
Line
Line with Markers
Pie
Bar
Manage Templates...
Set as Default Chart
OK
Cancel

INTERACTIVE TOOLS

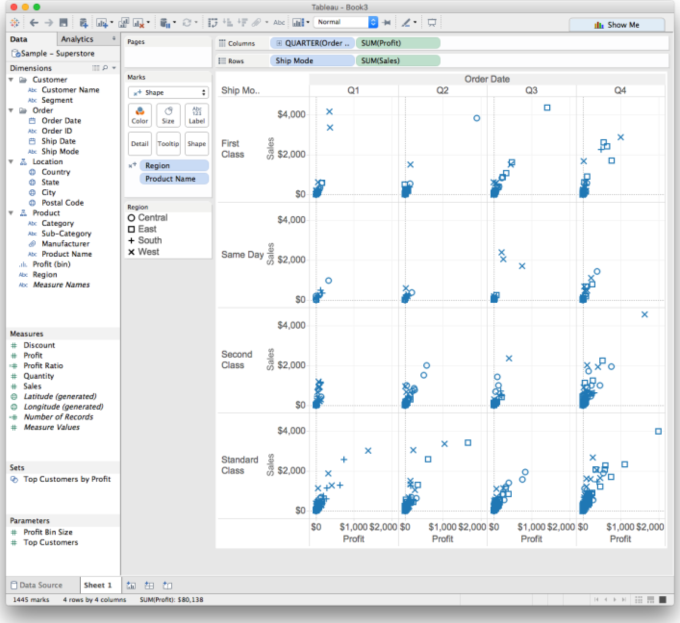
Power BI



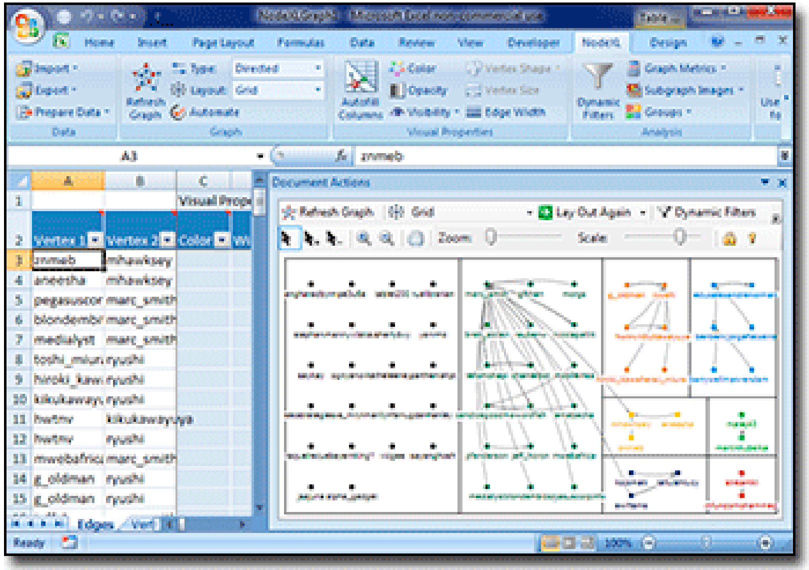


Qlik

+ a b | e a u



NETWORK AND GRAPH DATA



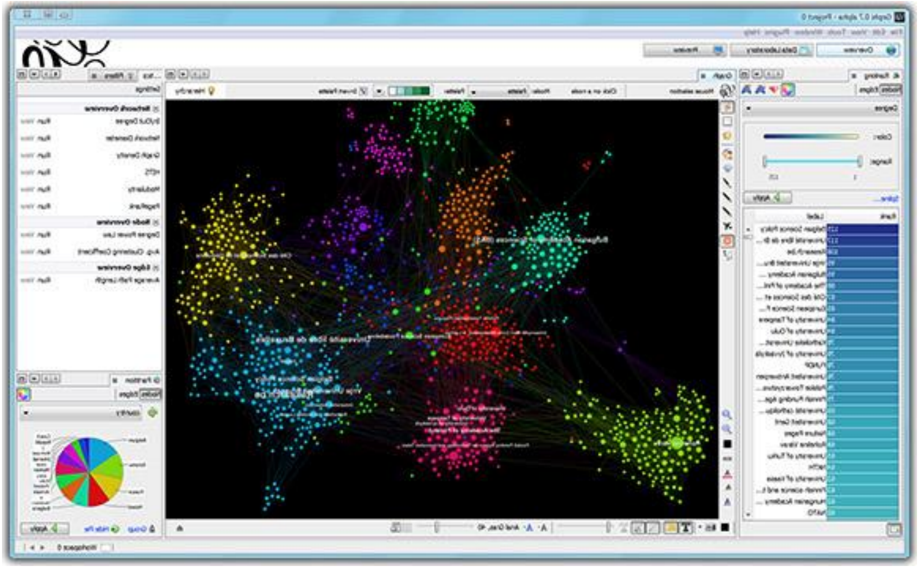
Type: Directed

Layout: Grid

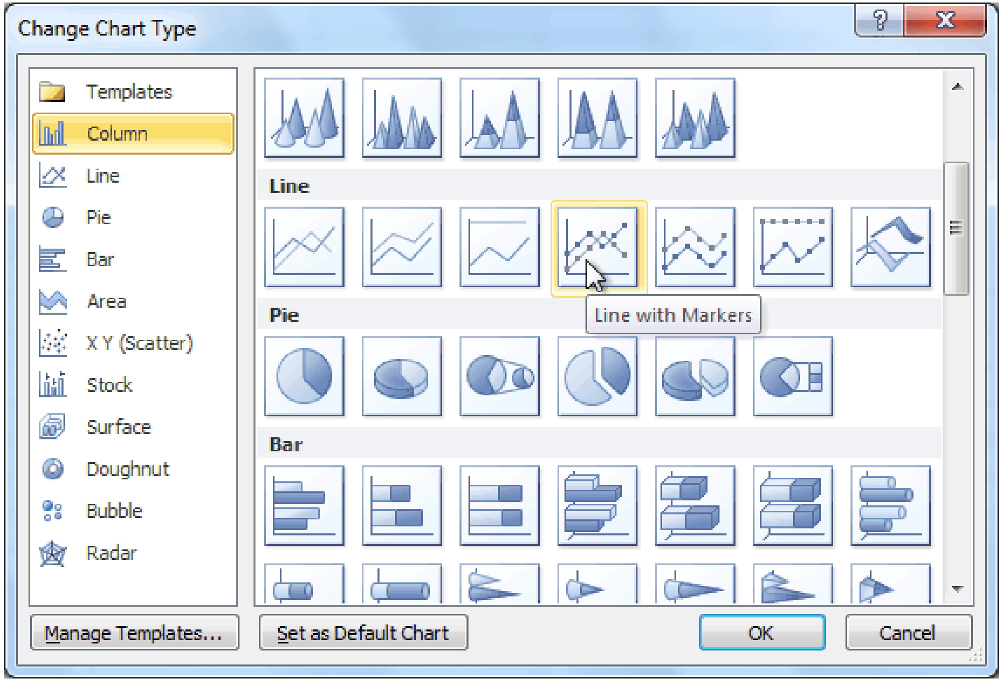
A3 = znmeb

	A	B	C
1			Visual Prope
2	Vertex 1	Vertex 2	Color
3	znmeb	mhawksey	
4	aneesha	mhawksey	
5	pegasuscor	marc_smith	
6	blondembr	marc_smith	
7	medialyst	marc_smith	
8	toshi_miun	ryushi	
9	hiroki_kawa	ryushi	
10	kikukaways	ryushi	
11	hwtmr	kikukawayuya	
12	hwtmr	ryushi	
13	mwebafrica	marc_smith	
14	g_oldman	ryushi	
15	g_oldman	ryushi	

NodeXL



Gephi



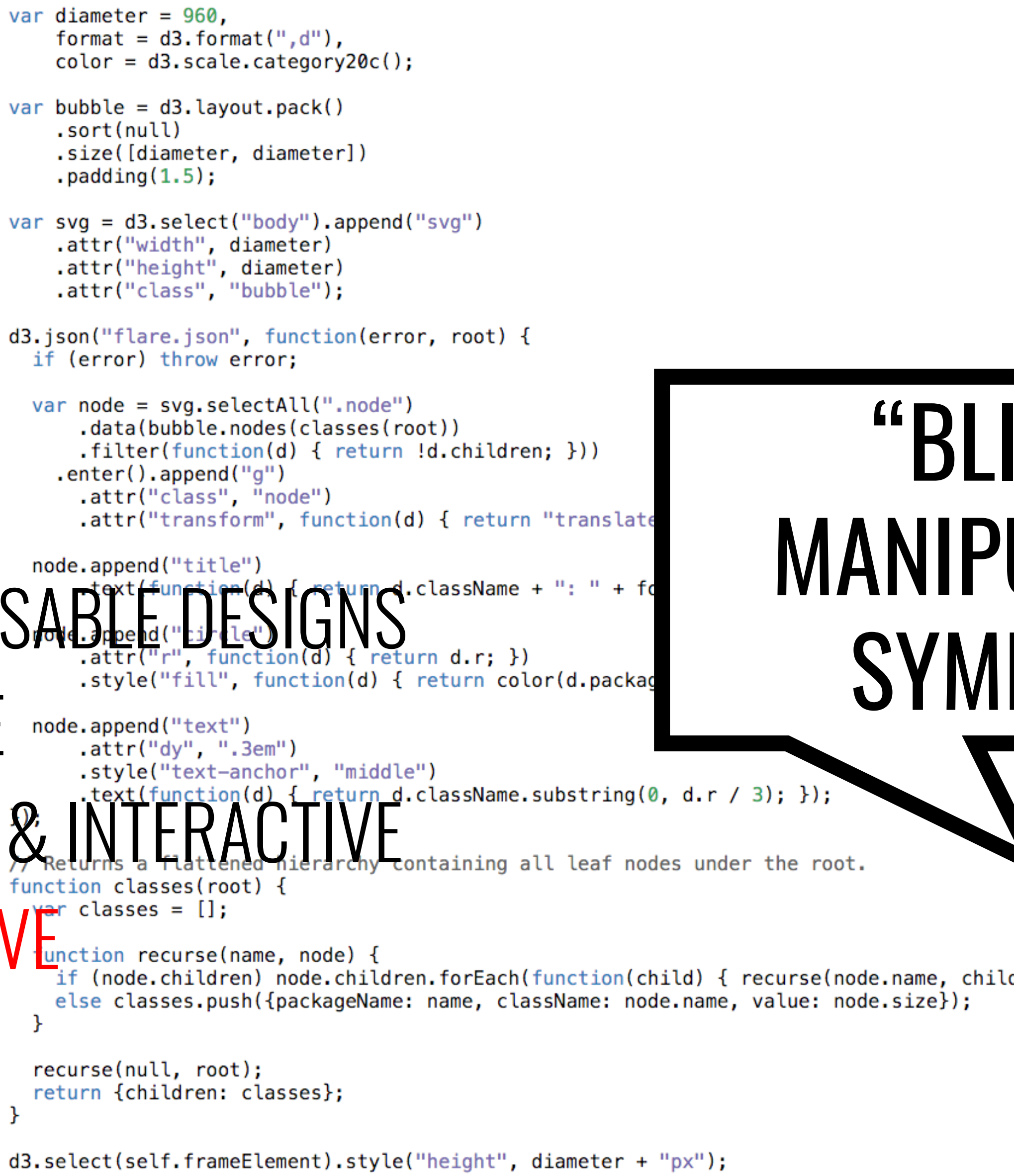
USE

+ EASY
+ SCALABLE
- LIMITED FLEXIBILITY
- LIMITED EXPRESSIVENESS
- WHAT IF I NEED A NEW CHART TYPE?

CODE

+ NEW, REUSABLE DESIGNS
+ SCALABLE
+ DYNAMIC & INTERACTIVE
~ EXPRESSIVE
- HARD

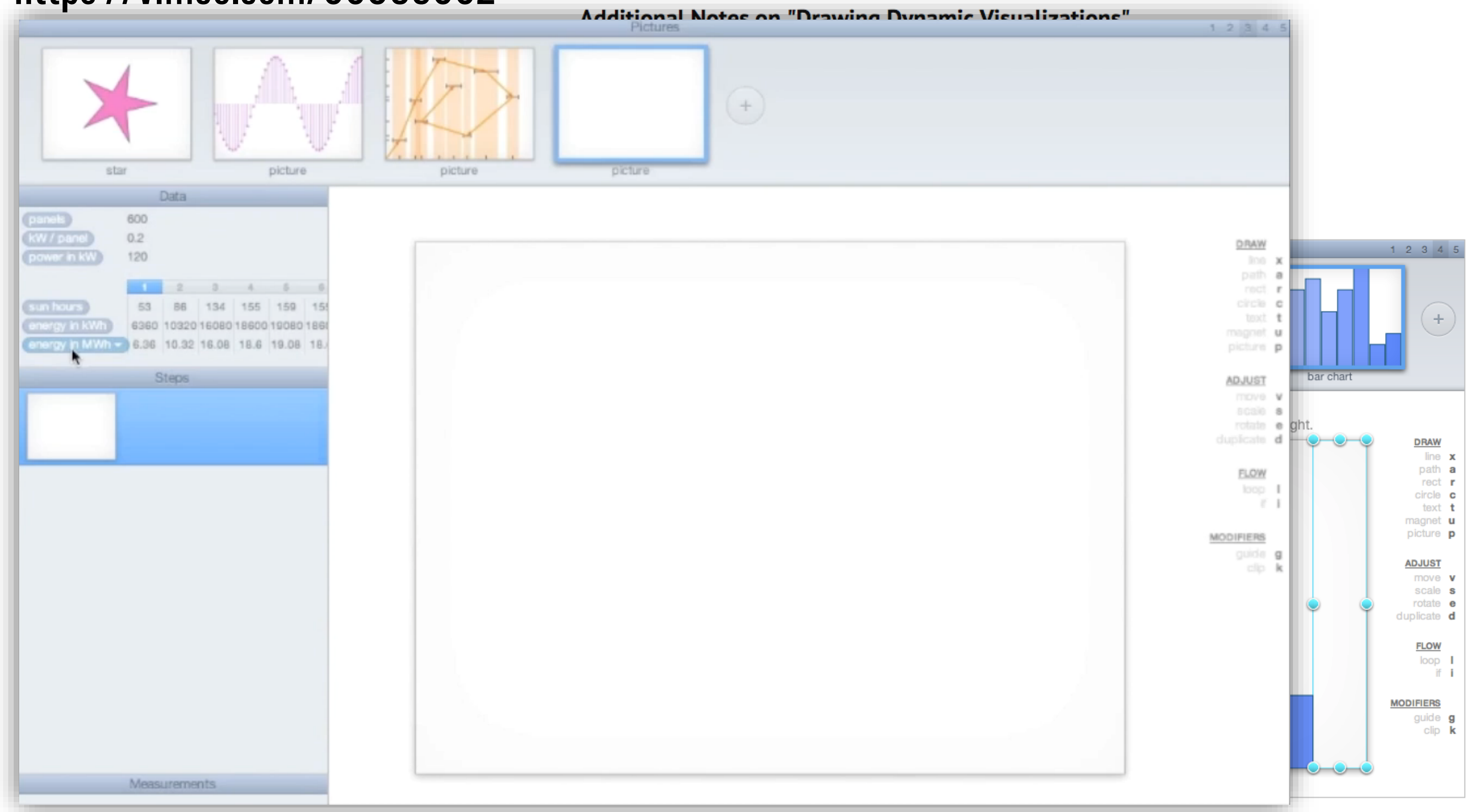
“BLINDLY MANIPULATING SYMBOLS”



```
var diameter = 960,
    format = d3.format(",d"),
    color = d3.scale.category20c();

var bubble = d3.layout.pack()
    .sort(null)
    .size([diameter, diameter])
    .padding(1.5);

var svg = d3.select("body").append("svg")
    .attr("width", diameter)
    .attr("height", diameter)
    .attr("class", "bubble");

d3.json("flare.json", function(error, root) {
  if (error) throw error;

  var node = svg.selectAll(".node")
      .data(bubble.nodes(classes(root))
      .filter(function(d) { return !d.children; }))
    .enter().append("g")
      .attr("class", "node")
      .attr("transform", function(d) { return "translate

  node.append("title")
      .text(function(d) { return d.className + ": " + fo

  node.append("circle")
      .attr("r", function(d) { return d.r; })
      .style("fill", function(d) { return color(d.packag

  node.append("text")
      .attr("dy", ".3em")
      .style("text-anchor", "middle")
      .text(function(d) { return d.className.substring(0, d.r / 3); });
});

// Returns a flattened hierarchy containing all leaf nodes under the root.
function classes(root) {
  var classes = [];

  function recurse(name, node) {
    if (node.children) node.children.forEach(function(child) { recurse(node.name, child
    else classes.push({packageName: name, className: node.name, value: node.size});
  }

  recurse(null, root);
  return {children: classes};
}

d3.select(self.frameElement).style("height", diameter + "px");
```

DRAWING DYNAMIC VISUALIZATIONS

https://vimeo.com/66085662

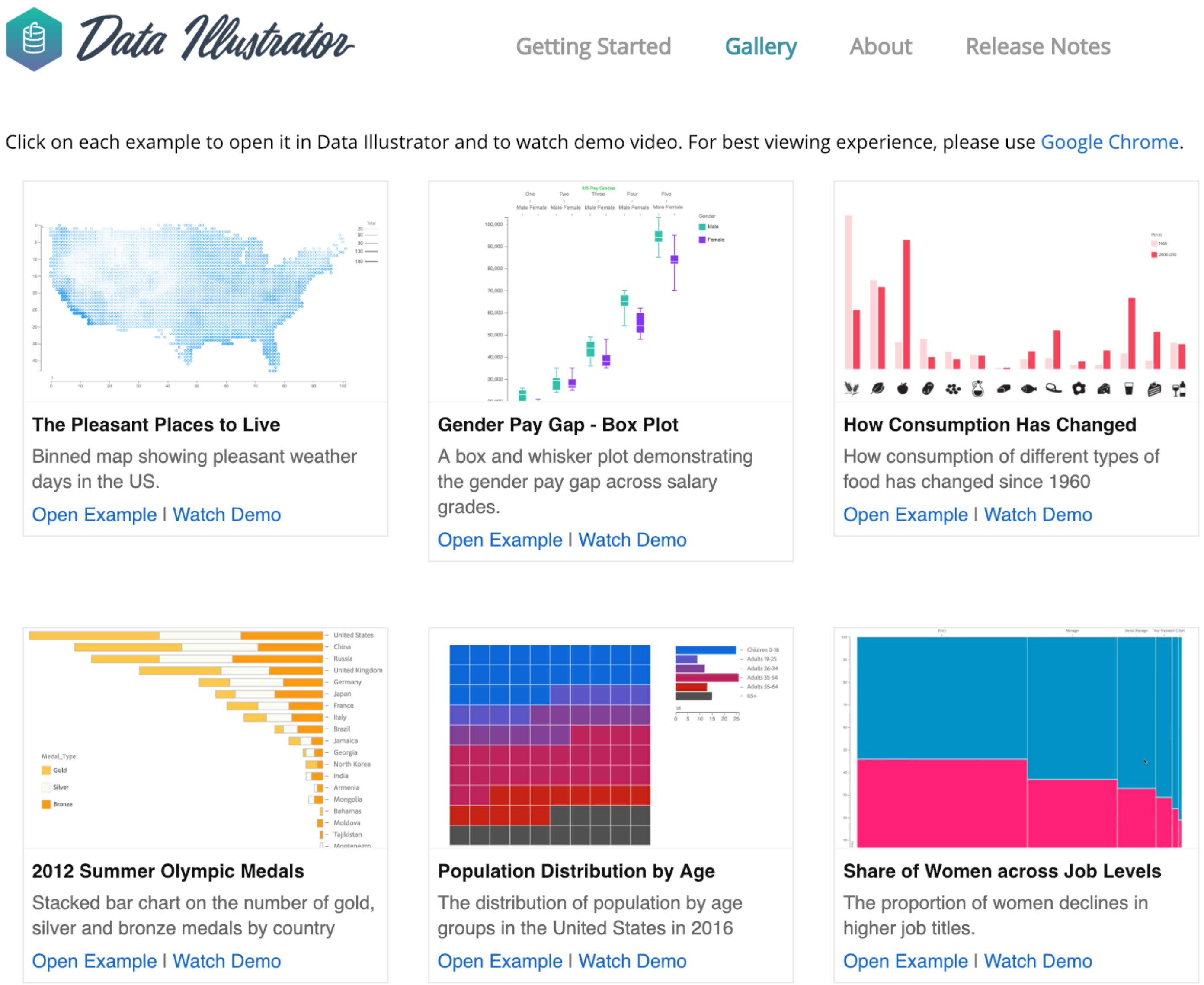


Additional Notes on "Drawing Dynamic Visualizations"

Data

panels	600
kW / panel	0.2
power in kW	120

	1	2	3	4	5	6
sun hours	53	86	134	155	159	15
energy in kWh	6360	10320	16080	18600	19080	186
energy in MWh	6.36	10.32	16.08	18.6	19.08	18.



Click on each example to open it in Data Illustrator and to watch demo video. For best viewing experience, please use Google Chrome.

The Pleasant Places to Live

Binned map showing pleasant weather days in the US.

Open Example | Watch Demo

Gender Pay Gap - Box Plot

A box and whisker plot demonstrating the gender pay gap across salary grades.

Open Example | Watch Demo

How Consumption Has Changed

How consumption of different types of food has changed since 1960

Open Example | Watch Demo

2012 Summer Olympic Medals

Stacked bar chart on the number of gold, silver and bronze medals by country

Open Example | Watch Demo

Population Distribution by Age

The distribution of population by age groups in the United States in 2016

Open Example | Watch Demo

Share of Women across Job Levels

The proportion of women declines in higher job titles.

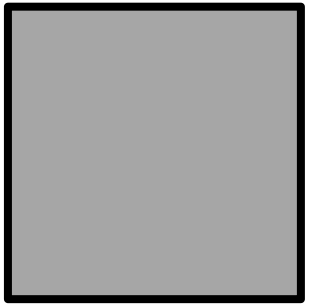
Open Example | Watch Demo

SPECIFYING VISUAL REPRESENTATIONS AS CODE

SOME DIFFERENT APPROACHES

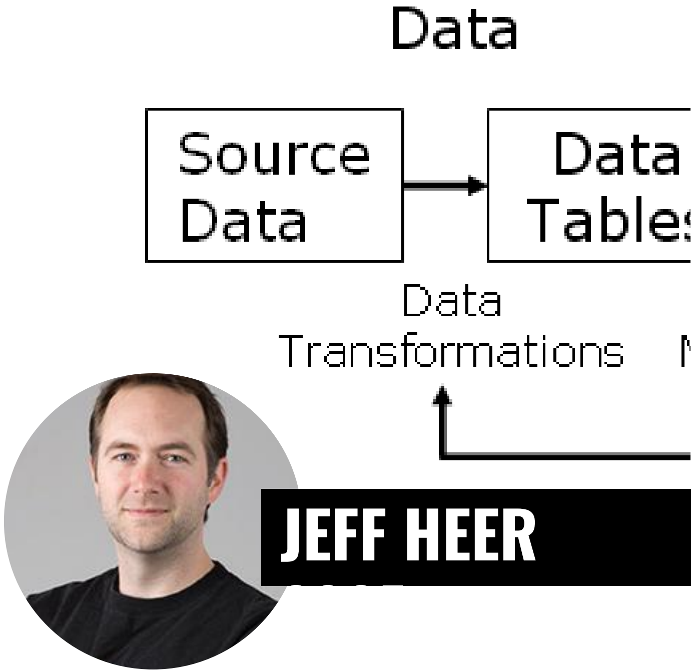
DRAWING PIXELS

```
background(255); // Setting the background to white
stroke(0); // Setting the outline (stroke) to black
fill(150); // Setting the interior of a shape (fill)
to grey rect(50,50,75,100); // Drawing the rectangle
```

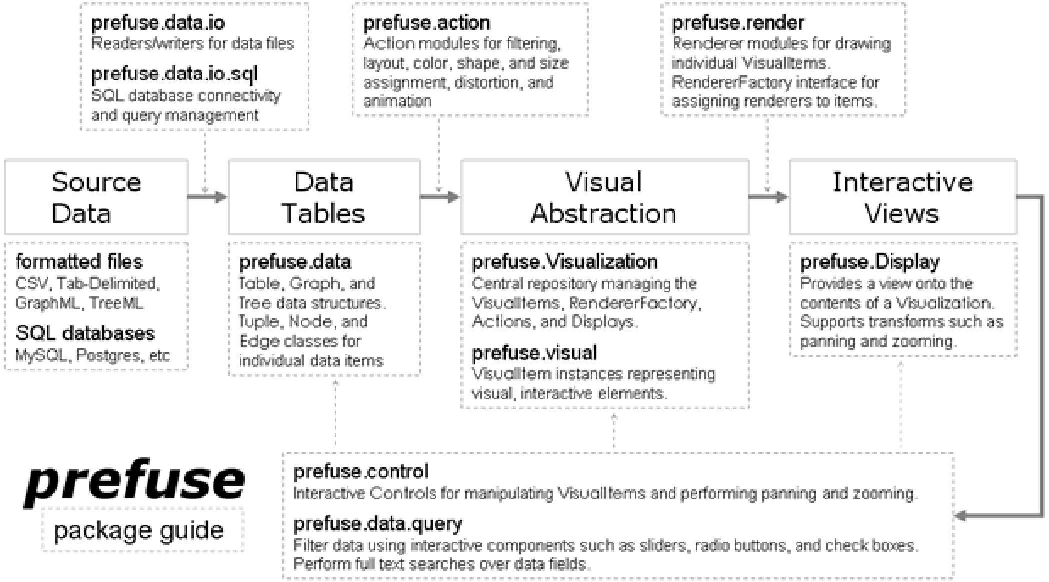


SOME DIFFERENT APPROACHES

COMPUTATIONALLY IMPLEMENTING VISUALIZATION REFERENCE MODEL



JEFF HEER



prefuse.data.io
Readers/writers for data files

prefuse.data.io.sql
SQL database connectivity and query management

prefuse.action
Action modules for filtering, layout, color, shape, and size assignment, distortion, and animation

prefuse.render
Renderer modules for drawing individual VisualItems. RendererFactory interface for assigning renderers to items.

formatted files
CSV, Tab-Delimited, GraphML, TreeML

SQL databases
MySQL, Postgres, etc

prefuse.data
Table, Graph, and Tree data structures. Tuple, Node, and Edge classes for individual data items

prefuse.Visualization
Central repository managing the VisualItems, RendererFactory, Actions, and Displays.

prefuse.visual
VisualItem instances representing visual, interactive elements.

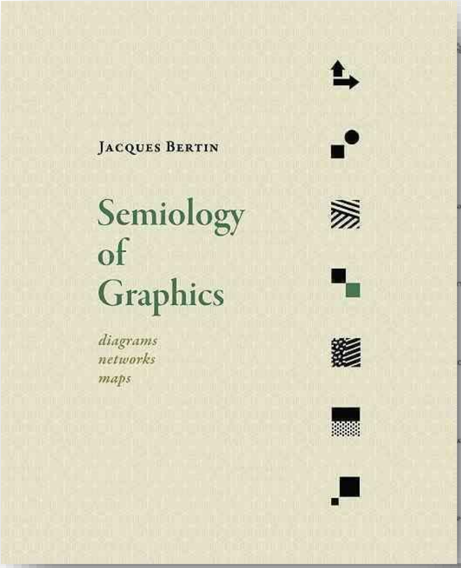
prefuse.Display
Provides a view onto the contents of a Visualization. Supports transforms such as panning and zooming.

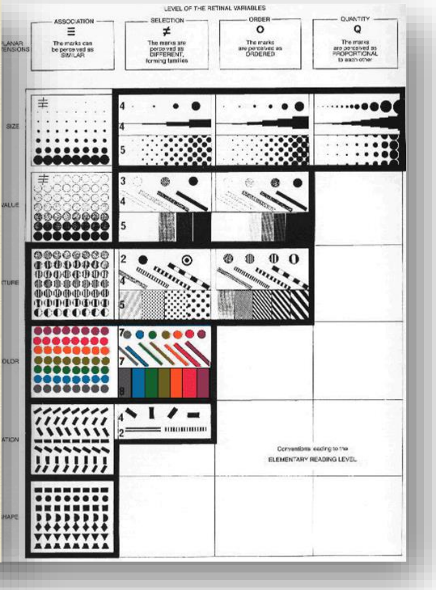
prefuse.control
Interactive Controls for manipulating VisualItems and performing panning and zooming.

prefuse.data.query
Filter data using interactive components such as sliders, radio buttons, and check boxes. Perform full text searches over data fields.

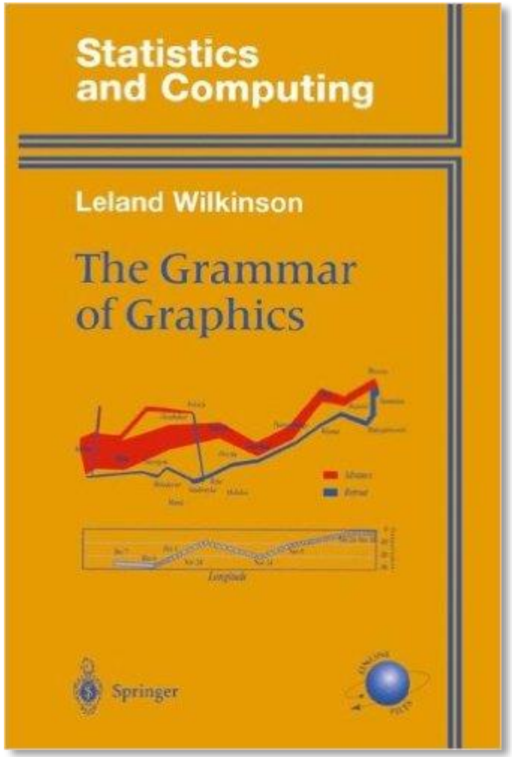
SOME DIFFERENT APPROACHES

DESCRIBING CONCEPTUAL PROPERTIES OF VISUALIZATIONS




LEVEL OF THE RETINAL VARIABLES
ASSOCIATION
≡
The marks can be perceived as SIMILAR
SELECTION
≠
The marks are perceived as DIFFERENT, forming families
ORDER
O
The marks are perceived as ORDERED
QUANTITY
Q
The marks are perceived as PROPORTIONAL to each other
LANAR
ENSIONS
SIZE
ALUE
TURE
OLOR
ATION
HAPE
Conventions leading to the
ELEMENTARY READING LEVEL

JACQUES BERTIN 1963



Statistics
and Computing

Leland Wilkinson

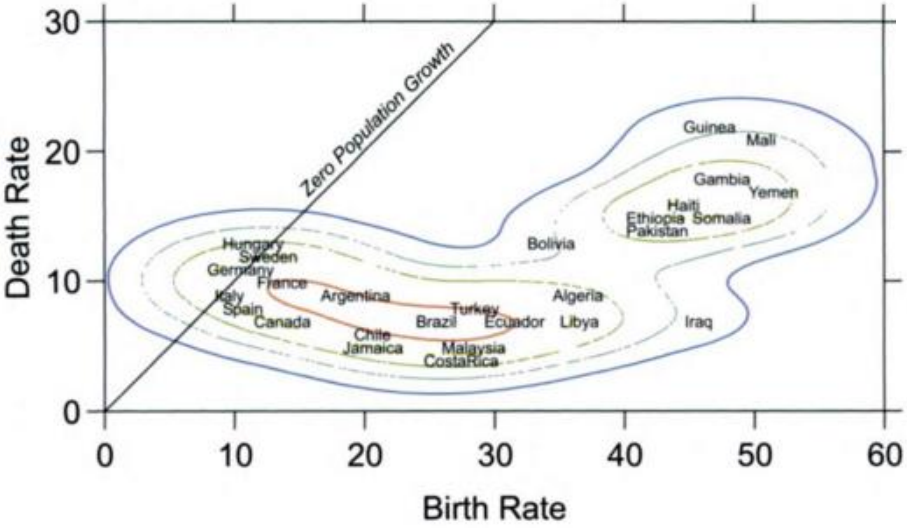
The Grammar
of Graphics

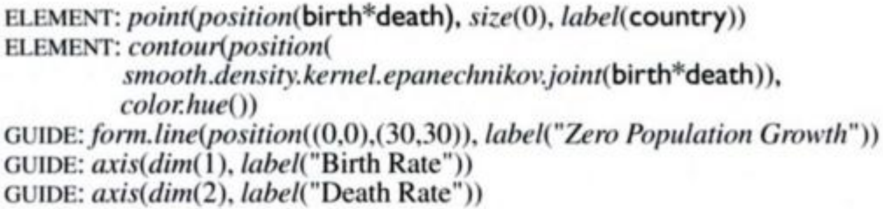
Springer

THE GRAMMAR OF GRAPHICS

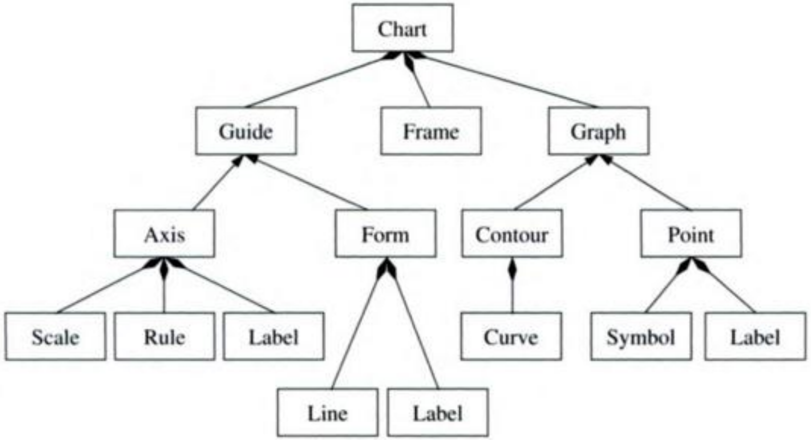
LELAND WILKINSON 1999

A FORMAL LANGUAGE FOR **DESCRIBING** DATA GRAPHICS





```
ELEMENT: point(position(birth*death), size(0), label(country))
ELEMENT: contour(position(
            smooth.density.kernel.epanechnikov.joint(birth*death)),
            color.hue())
GUIDE: form.line(position((0,0),(30,30)), label("Zero Population Growth"))
GUIDE: axis(dim(1), label("Birth Rate"))
GUIDE: axis(dim(2), label("Death Rate"))
```



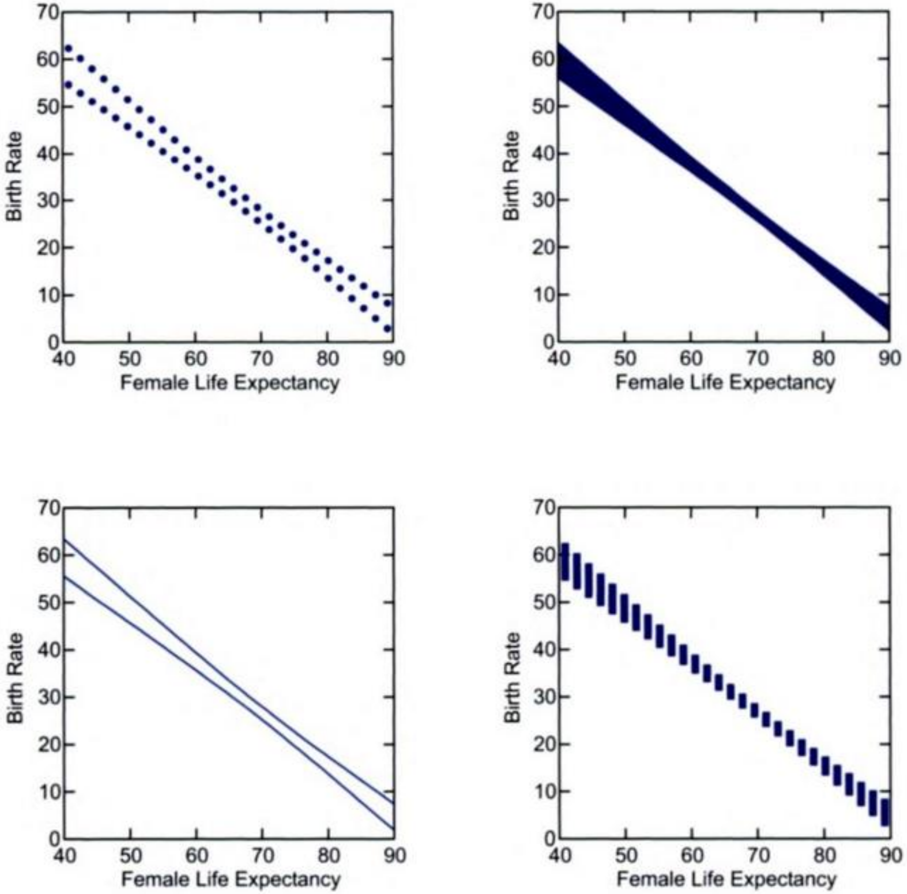
COMPOSABLE AND **GENERATIVE** LANGUAGES WAYS OF DESCRIBING A HUGE VARIETY OF CHART DESIGNS

ELEMENT: *point*(*position*(*region.confi.smooth.linear*(**female*birth**)))

ELEMENT: *line*(*position*(*region.confi.smooth.linear*(**female*birth**)))

ELEMENT: *area*(*position*(*region.confi.smooth.linear*(**female*birth**)))

ELEMENT: *interval*(*position*(*region.confi.smooth.linear*(**female*birth**)))



VizQL & POLARIS

Database Schema:
The user drags fields from the database schema to shelves to define the visual specification.

Layer Tabs:
Each layer has its own tab; different transformations and mappings can be specified for each layer.

Axis Shelves:
The fields placed here determine the structure of the table and the types of graphs in each table pane.

Context Menu:
The context menu provides access to the data transformation and interaction capabilities of Polaris such as sorting, filtering, and aggregation.

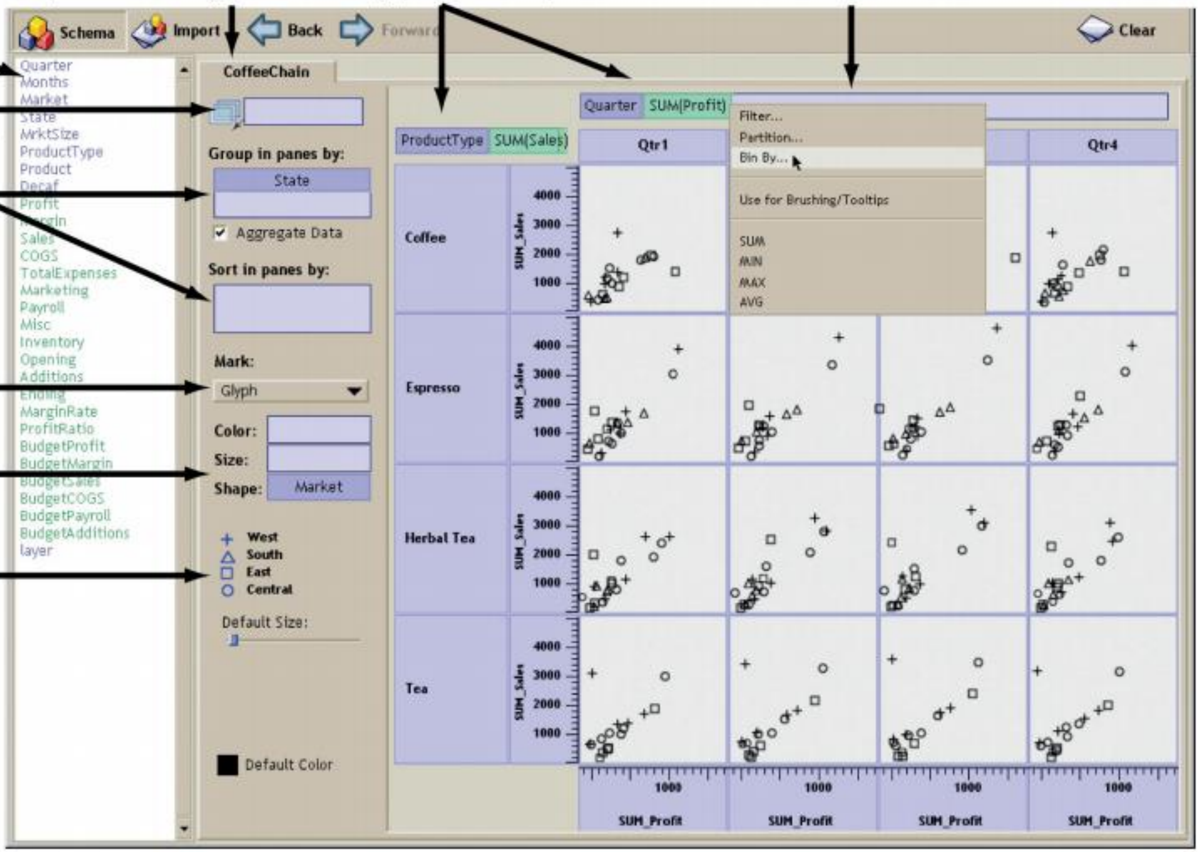
Layer Shelf:
The fields placed here determine how records are partitioned into layers.

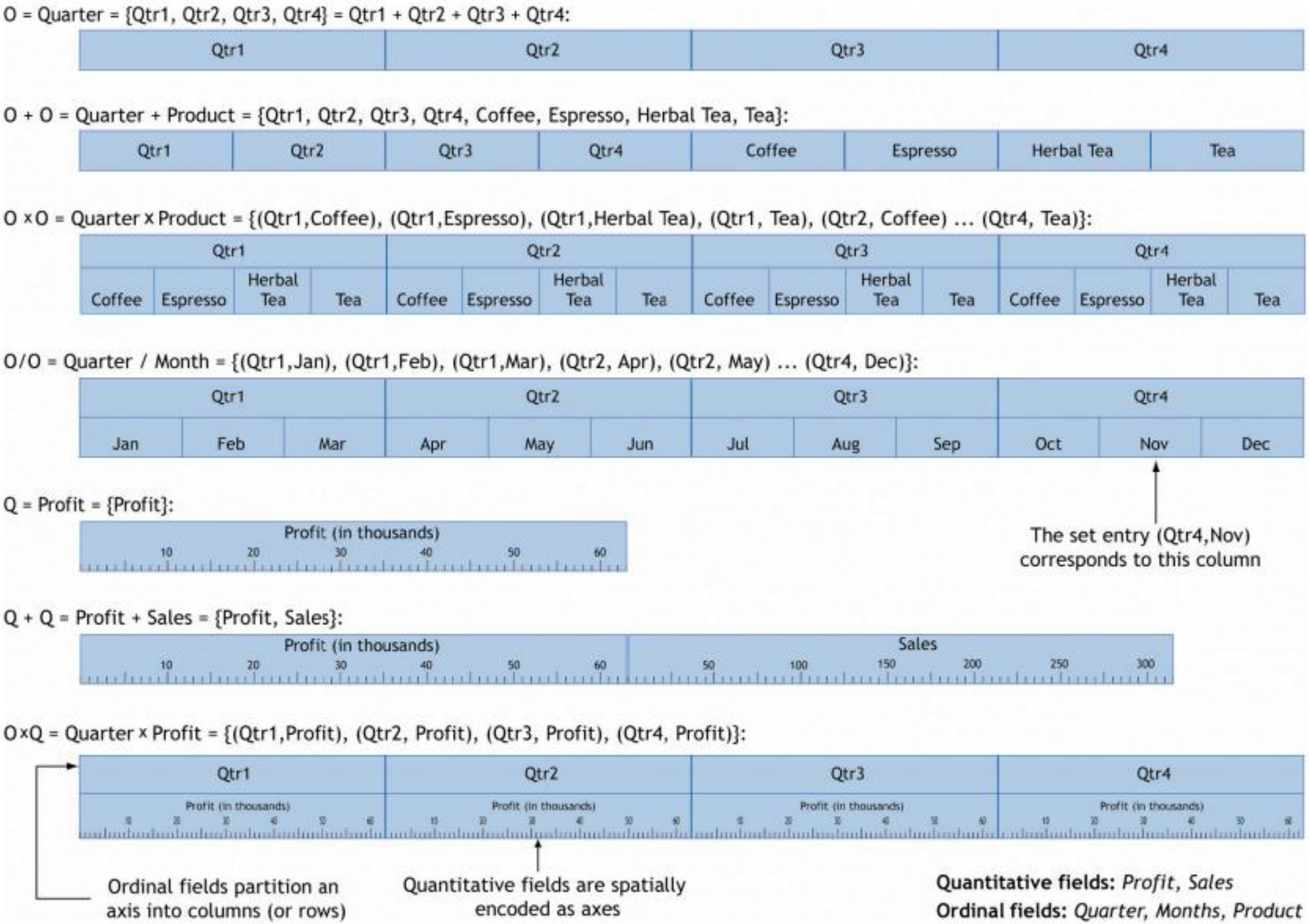
Grouping and Sorting Shelves:
The fields placed here determine how records are grouped and sorted within the table panes.

Mark Pulldown:
Relations in each pane are mapped to marks of the selected type.

Retinal Property Shelves:
The fields placed here determine how data is encoded in the retinal properties of the marks.

Legends:
Legends enable the user to see and modify the mappings from data to retinal properties.



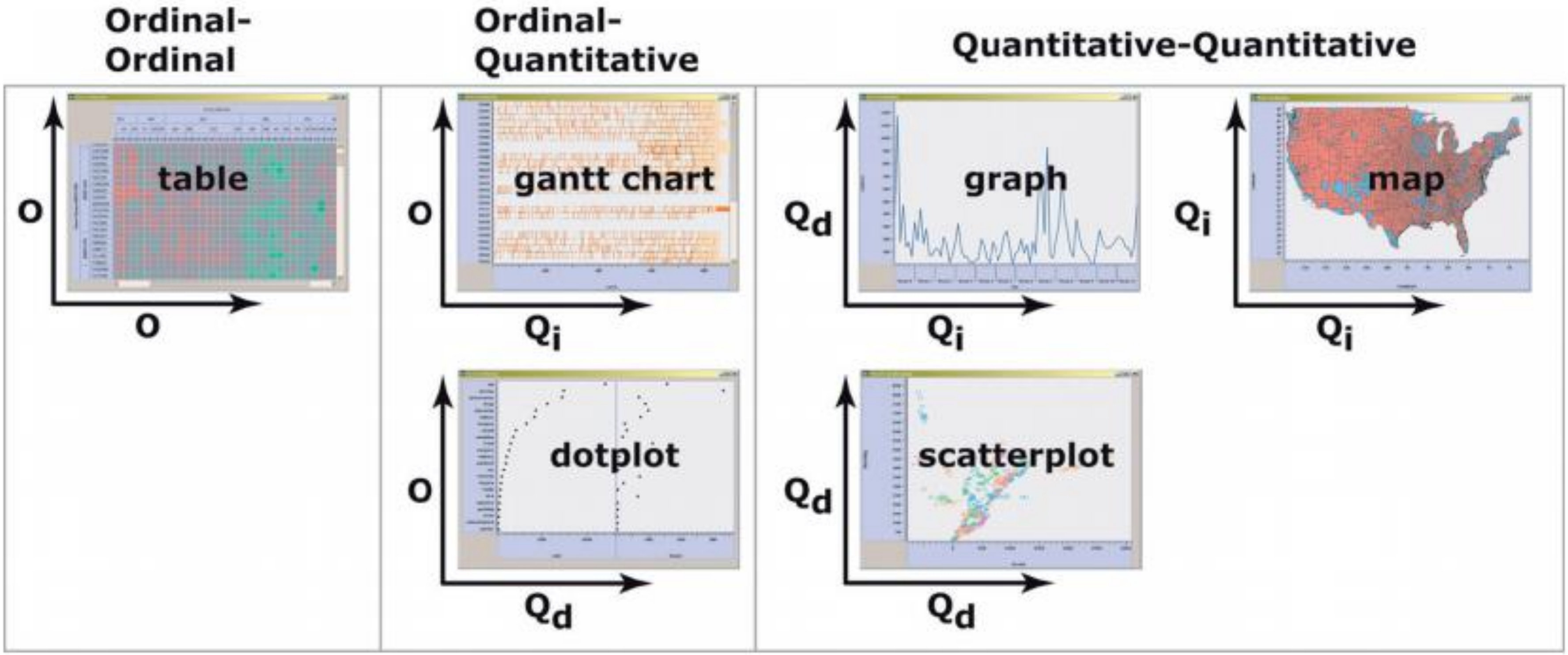

O = Quarter = {Qtr1, Qtr2, Qtr3, Qtr4} = Qtr1 + Qtr2 + Qtr3 + Qtr4:
Qtr1
Qtr2
Qtr3
Qtr4
O + O = Quarter + Product = {Qtr1, Qtr2, Qtr3, Qtr4, Coffee, Espresso, Herbal Tea, Tea}:
Qtr1
Qtr2
Qtr3
Qtr4
Coffee
Espresso
Herbal Tea
Tea
O ×O = Quarter x Product = {(Qtr1,Coffee), (Qtr1,Espresso), (Qtr1,Herbal Tea), (Qtr1, Tea), (Qtr2, Coffee) ... (Qtr4, Tea)}:
Qtr1
Qtr2
Qtr3
Qtr4
Coffee
Espresso
Herbal Tea
Tea
O/O = Quarter / Month = {(Qtr1,Jan), (Qtr1,Feb), (Qtr1,Mar), (Qtr2, Apr), (Qtr2, May) ... (Qtr4, Dec)}:
Qtr1
Qtr2
Qtr3
Qtr4
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Q = Profit = {Profit}:
Profit (in thousands)
10 20 30 40 50 60
Q + Q = Profit + Sales = {Profit, Sales}:
Profit (in thousands)
10 20 30 40 50 60
Sales
50 100 150 200 250 300
O×Q = Quarter x Profit = {(Qtr1,Profit), (Qtr2, Profit), (Qtr3, Profit), (Qtr4, Profit)}:
Qtr1
Qtr2
Qtr3
Qtr4
Profit (in thousands)
10 20 30 40 50 60

The set entry (Qtr4,Nov) corresponds to this column

Ordinal fields partition an axis into columns (or rows)

Quantitative fields are spatially encoded as axes

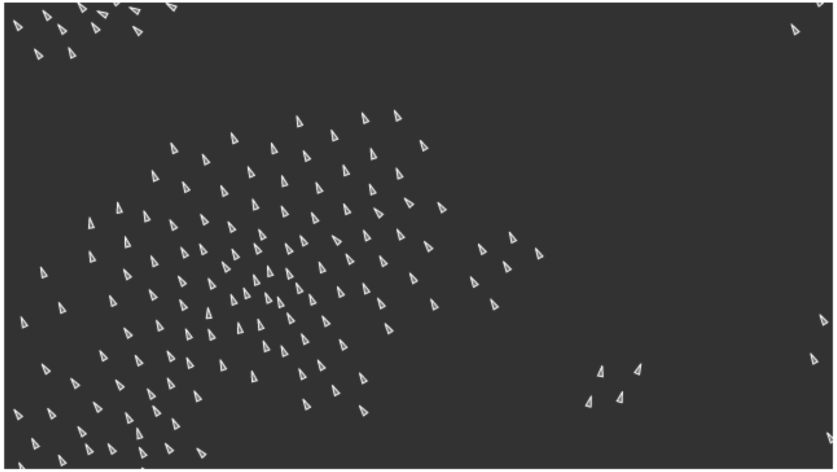
Quantitative fields: *Profit, Sales*
Ordinal fields: *Quarter, Months, Product*

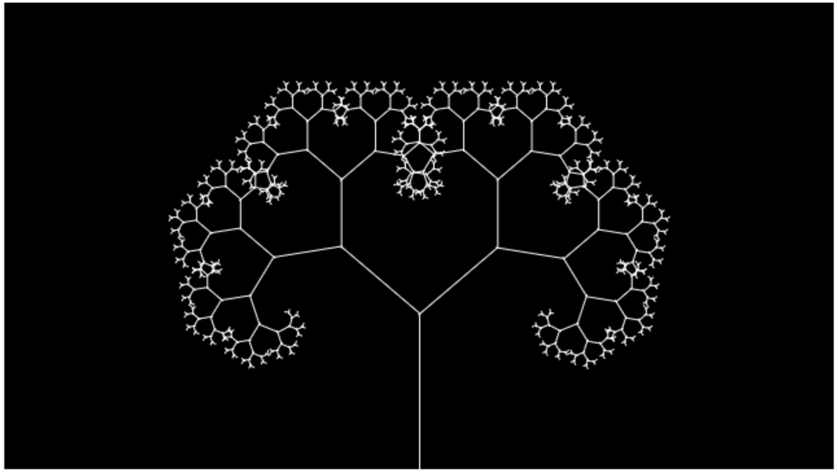

Ordinal-Ordinal
Ordinal-Quantitative
Quantitative-Quantitative
table
O
O
gantt chart
O
Q$_i$
dotplot
O
Q$_d$
graph
Q$_d$
Q$_i$
map
Q$_i$
Q$_i$
scatterplot
Q$_d$
Q$_d$

VISUALIZATION LANGUAGES AND TOOLKITS

DESKTOP

PROCESSING / P5.js



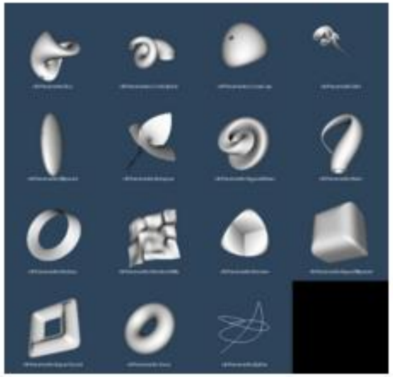


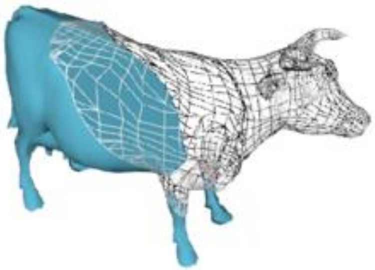
TARGETED AT ARTISTS & NON-EXPERTS

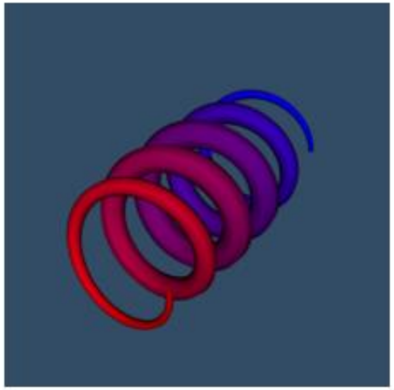
DESIGNED TO MAKE **DRAWING** & **INTERACTION** EASY

NO VISUALIZATION PRIMITIVES

VTK (VISUALIZATION TOOLKIT)





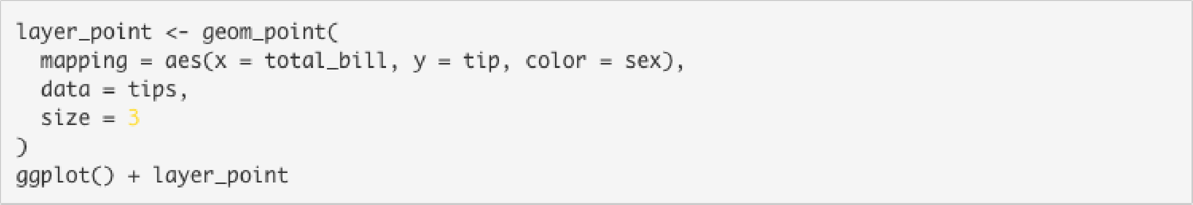


ESPECIALLY COMMON FOR SCIENTIFIC VISUALIZATION

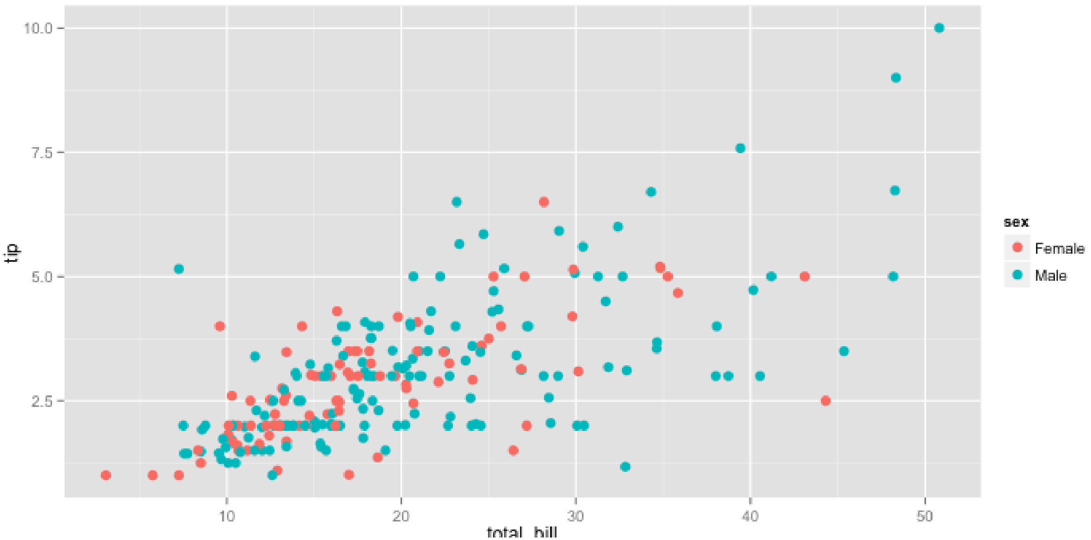
C++ (Python, Java, and Tcl WRAPPERS)

GGPLOT2

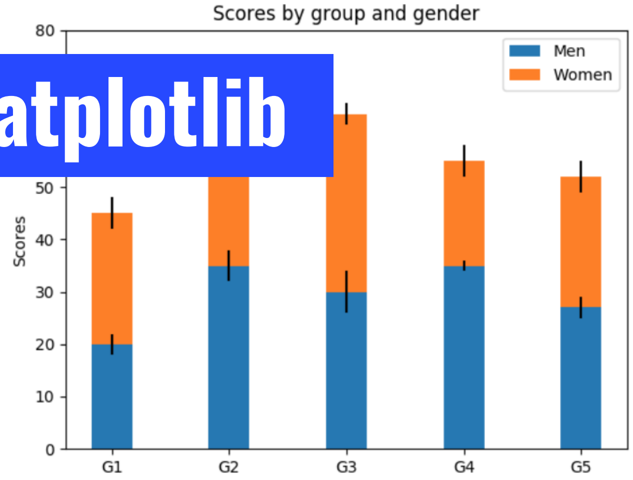
PLOTTING IN **R** BASED ON THE **GRAMMAR OF GRAPHICS**

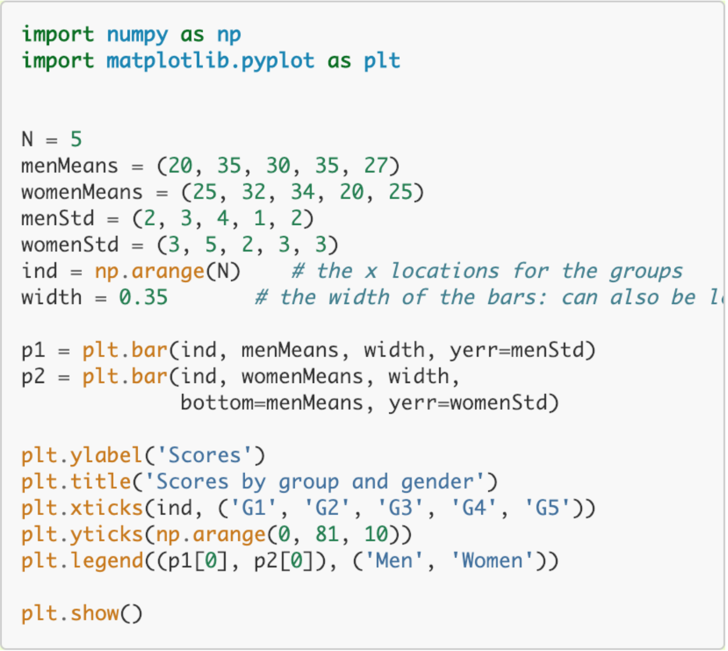


```
layer_point <- geom_point(
  mapping = aes(x = total_bill, y = tip, color = sex),
  data = tips,
  size = 3
)
ggplot() + layer_point
```



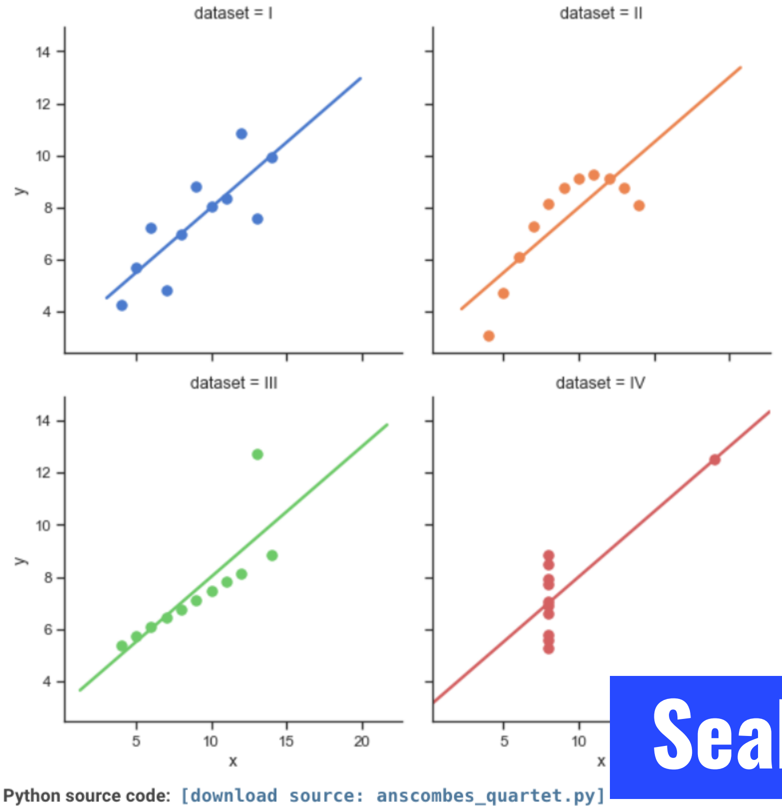
Matplotlib



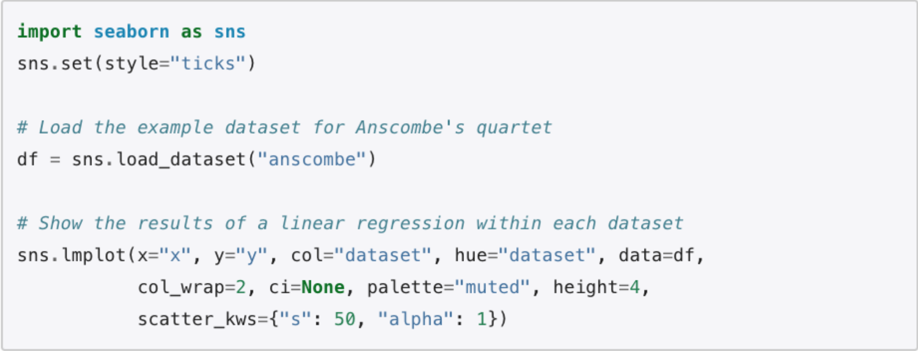


```python
import numpy as np
import matplotlib.pyplot as plt


N = 5
menMeans = (20, 35, 30, 35, 27)
womenMeans = (25, 32, 34, 20, 25)
menStd = (2, 3, 4, 1, 2)
womenStd = (3, 5, 2, 3, 3)
ind = np.arange(N)    # the x locations for the groups
width = 0.35       # the width of the bars: can also be l

p1 = plt.bar(ind, menMeans, width, yerr=menStd)
p2 = plt.bar(ind, womenMeans, width,
             bottom=menMeans, yerr=womenStd)

plt.ylabel('Scores')
plt.title('Scores by group and gender')
plt.xticks(ind, ('G1', 'G2', 'G3', 'G4', 'G5'))
plt.yticks(np.arange(0, 81, 10))
plt.legend((p1[0], p2[0]), ('Men', 'Women'))

plt.show()
```

Seaborn



Python source code: [download source: anscombes_quartet.py]



```python
import seaborn as sns
sns.set(style="ticks")

# Load the example dataset for Anscombe's quartet
df = sns.load_dataset("anscombe")

# Show the results of a linear regression within each dataset
sns.lmplot(x="x", y="y", col="dataset", hue="dataset", data=df,
           col_wrap=2, ci=None, palette="muted", height=4,
           scatter_kws={"s": 50, "alpha": 1})
```

VISUALIZATION LANGUAGES AND TOOLKITS

WEB

WHY DEVELOP FOR THE WEB?

NOW THE **DOMINANT PLATFORM FOR VIS CONSUMPTION**

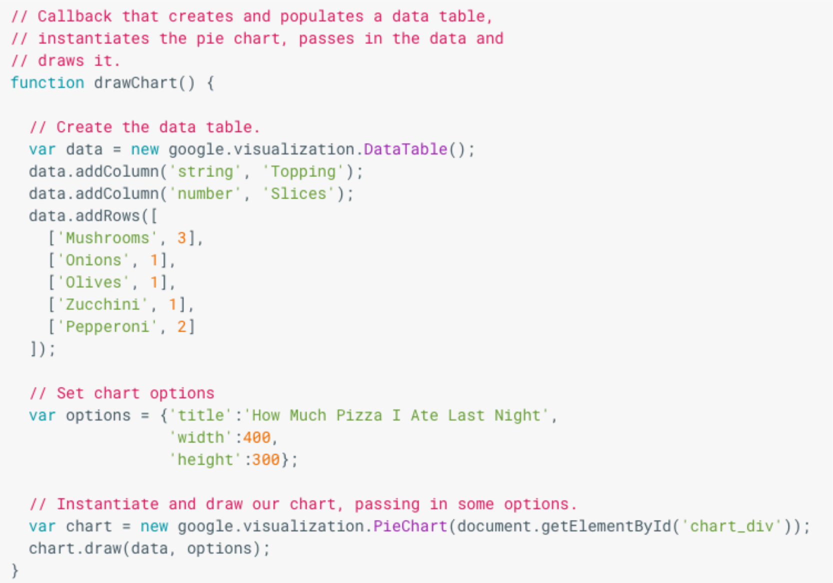
INTEGRATE VISUALIZATIONS INTO WEB PAGES AND APPLICATIONS

LEVERAGE **OTHER HTML5/JS LIBRARIES** AND TOOLS

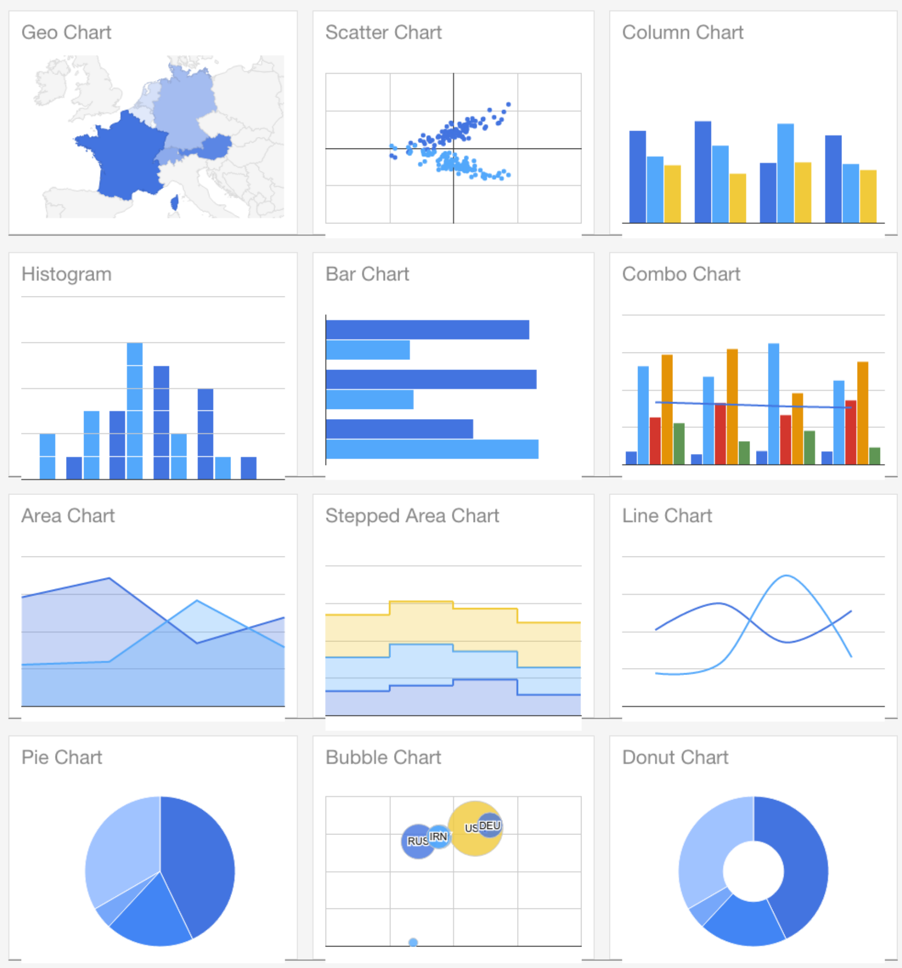
DEBUG AND TUNE IN THE BROWSER

GOOGLE CHARTS

EASY TO INSERT **PREDEFINED CHARTS TYPES** INTO PAGES AND STYLE THEM

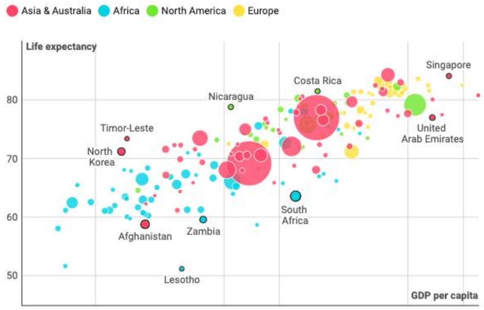


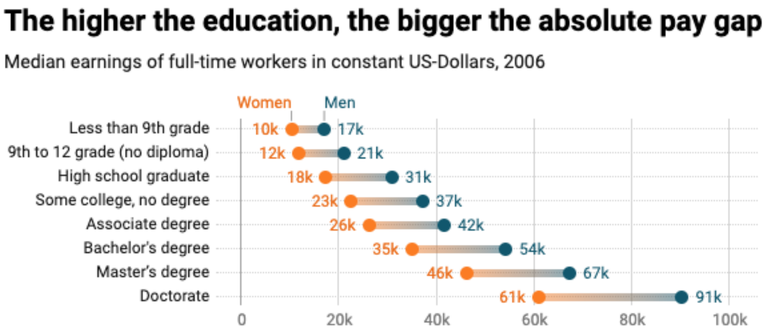
```
// Callback that creates and populates a data table,
// instantiates the pie chart, passes in the data and
// draws it.
function drawChart() {

  // Create the data table.
  var data = new google.visualization.DataTable();
  data.addColumn('string', 'Topping');
  data.addColumn('number', 'Slices');
  data.addRows([
    ['Mushrooms', 3],
    ['Onions', 1],
    ['Olives', 1],
    ['Zucchini', 1],
    ['Pepperoni', 2]
  ]);

  // Set chart options
  var options = {'title':'How Much Pizza I Ate Last Night',
                 'width':400,
                 'height':300};

  // Instantiate and draw our chart, passing in some options.
  var chart = new google.visualization.PieChart(document.getElementById('chart_div'));
  chart.draw(data, options);
}
```



DATAWRAPPER

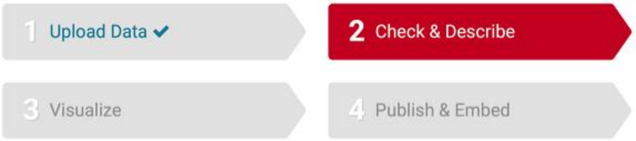
LOTS OF COMMON CHART TYPES WITH INTERACTIVITY VIA A **VISUAL INTERFACE**

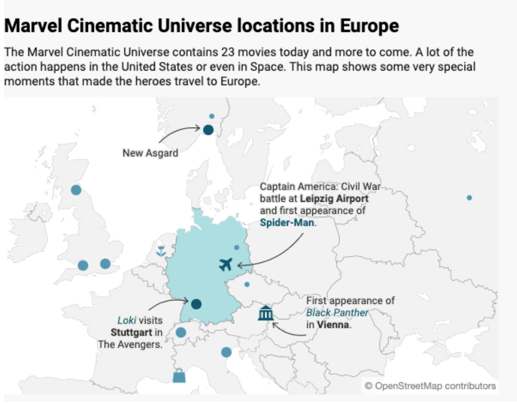




The higher the education, the bigger the absolute pay gap

Median earnings of full-time workers in constant US-Dollars, 2006



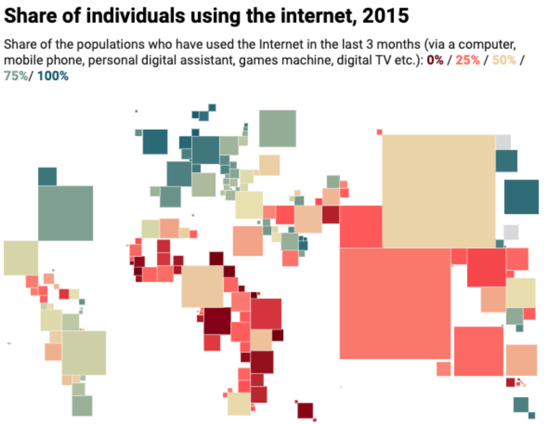


Marvel Cinematic Universe locations in Europe

The Marvel Cinematic Universe contains 23 movies today and more to come. A lot of the action happens in the United States or even in Space. This map shows some very special moments that made the heroes travel to Europe.

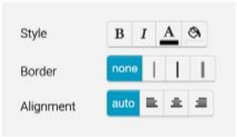
New Asgard
Captain America: Civil War battle at Leipzig Airport and first appearance of Spider-Man.
First appearance of Black Panther in Vienna.
Loki visits Stuttgart in The Avengers.

© OpenStreetMap contributors



Share of individuals using the internet, 2015

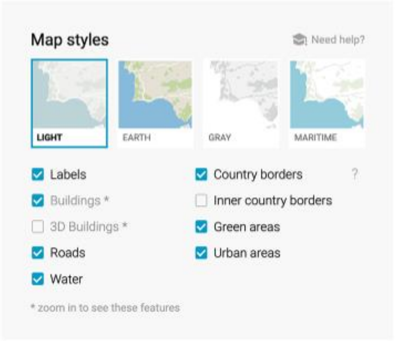
Share of the populations who have used the Internet in the last 3 months (via a computer, mobile phone, personal digital assistant, games machine, digital TV etc.): **0%** / 25% / 50% / 75%/ **100%**



Style B I A

Border none

Alignment auto



Map styles

☑ Labels

☑ Buildings *

☐ 3D Buildings *

☑ Roads

☑ Water

☑ Country borders

☐ Inner country borders

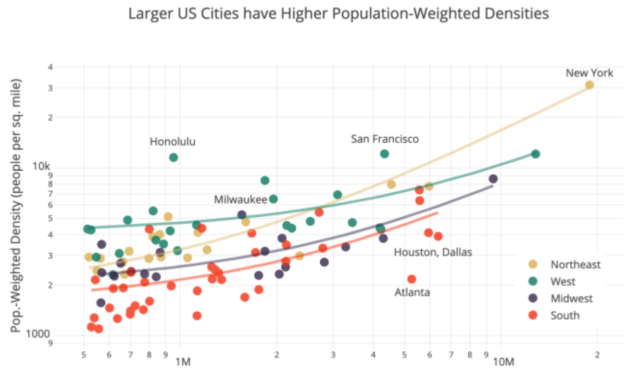
☑ Green areas

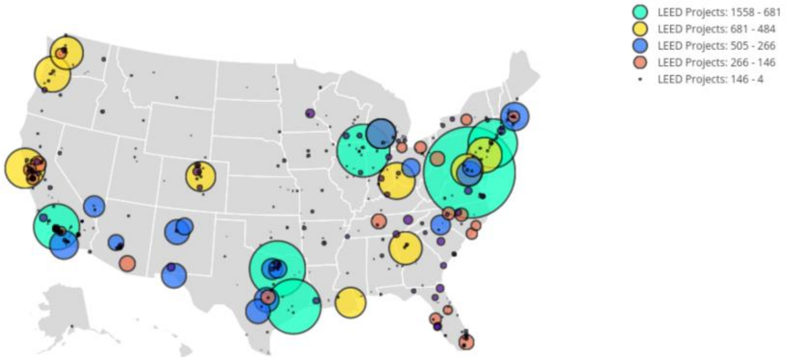
☑ Urban areas

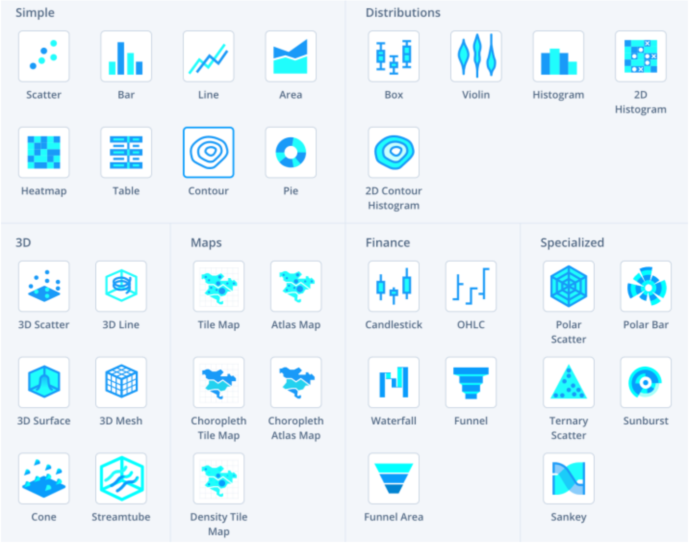
* zoom in to see these features

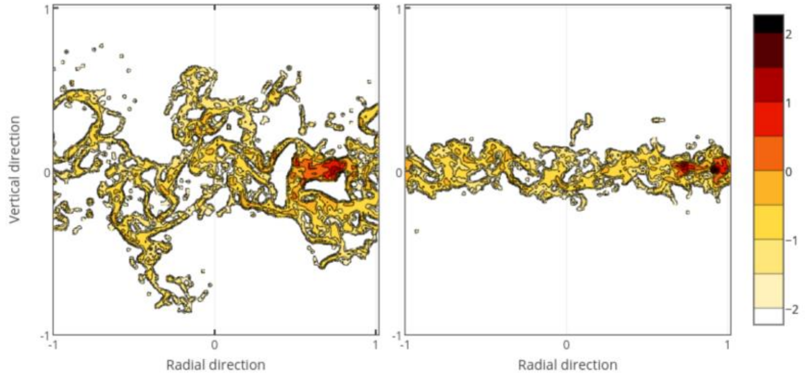
PLOT.LY

LARGE VOCABULARY OF CHART TEMPLATES + **VISUAL INTERFACE + SCRIPTING**









D3.JS

Data-Driven Documents

MIKE BOSTOCK

JEFF HEER

D3.JS

JAVASCRIPT / HTML5 / SVG / CSS

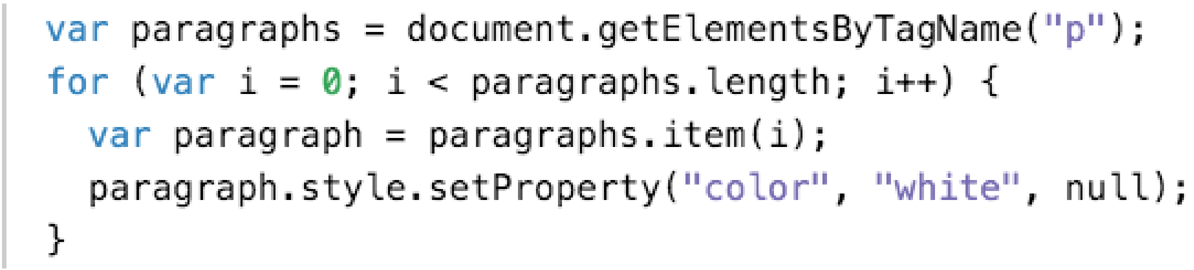
DYNAMIC DOCUMENT MANIPULATION AND **VISUALIZATION**

SUPPORT FOR **BINDING DATA TO ELEMENTS,**
HANDLING **SCALES** & **LAYOUTS, ANIMATION,**
AND MUCH MORE!

DECLARATIVE VISUALIZATION DESIGN

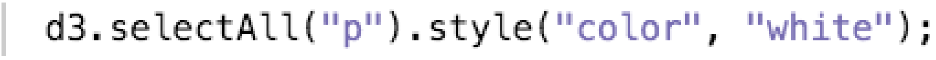
(SORT OF LIKE **JQUERY** FOR VISUALIZATION)

IMPERATIVE

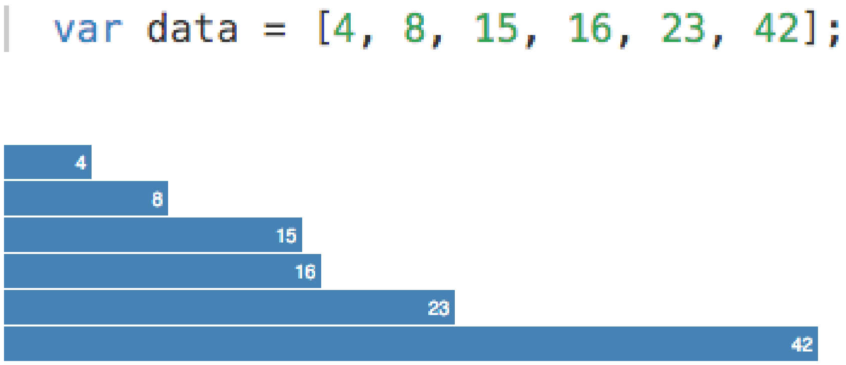


```
var paragraphs = document.getElementsByTagName("p");
for (var i = 0; i < paragraphs.length; i++) {
  var paragraph = paragraphs.item(i);
  paragraph.style.setProperty("color", "white", null);
}
```

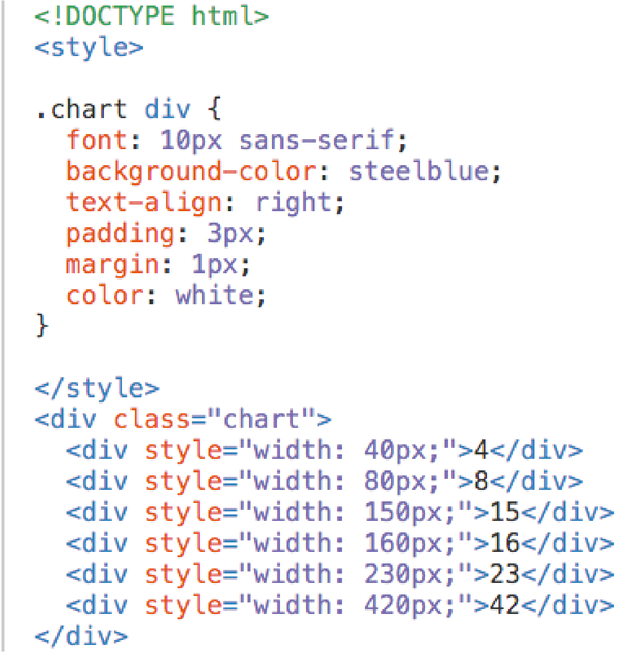
DECLARATIVE



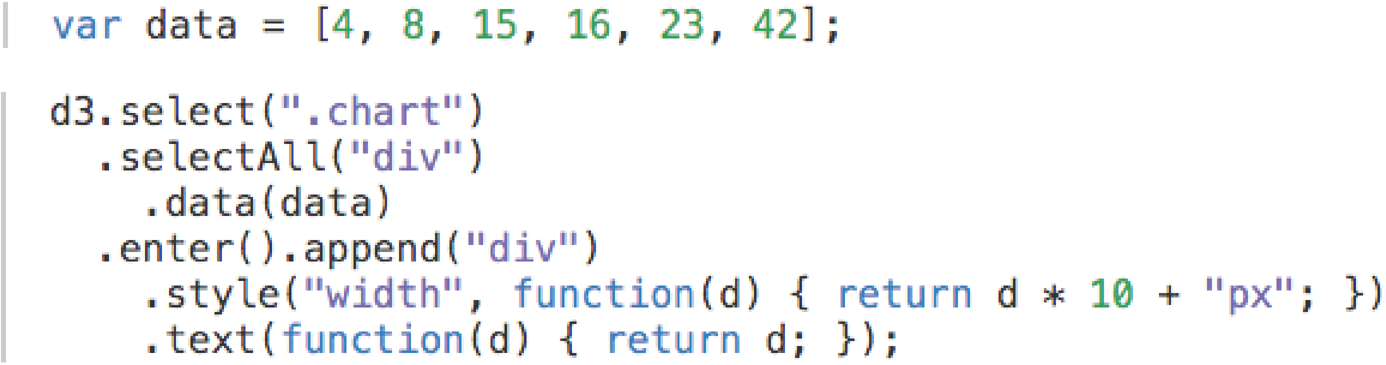
```
d3.selectAll("p").style("color", "white");
```



```
var data = [4, 8, 15, 16, 23, 42];
```

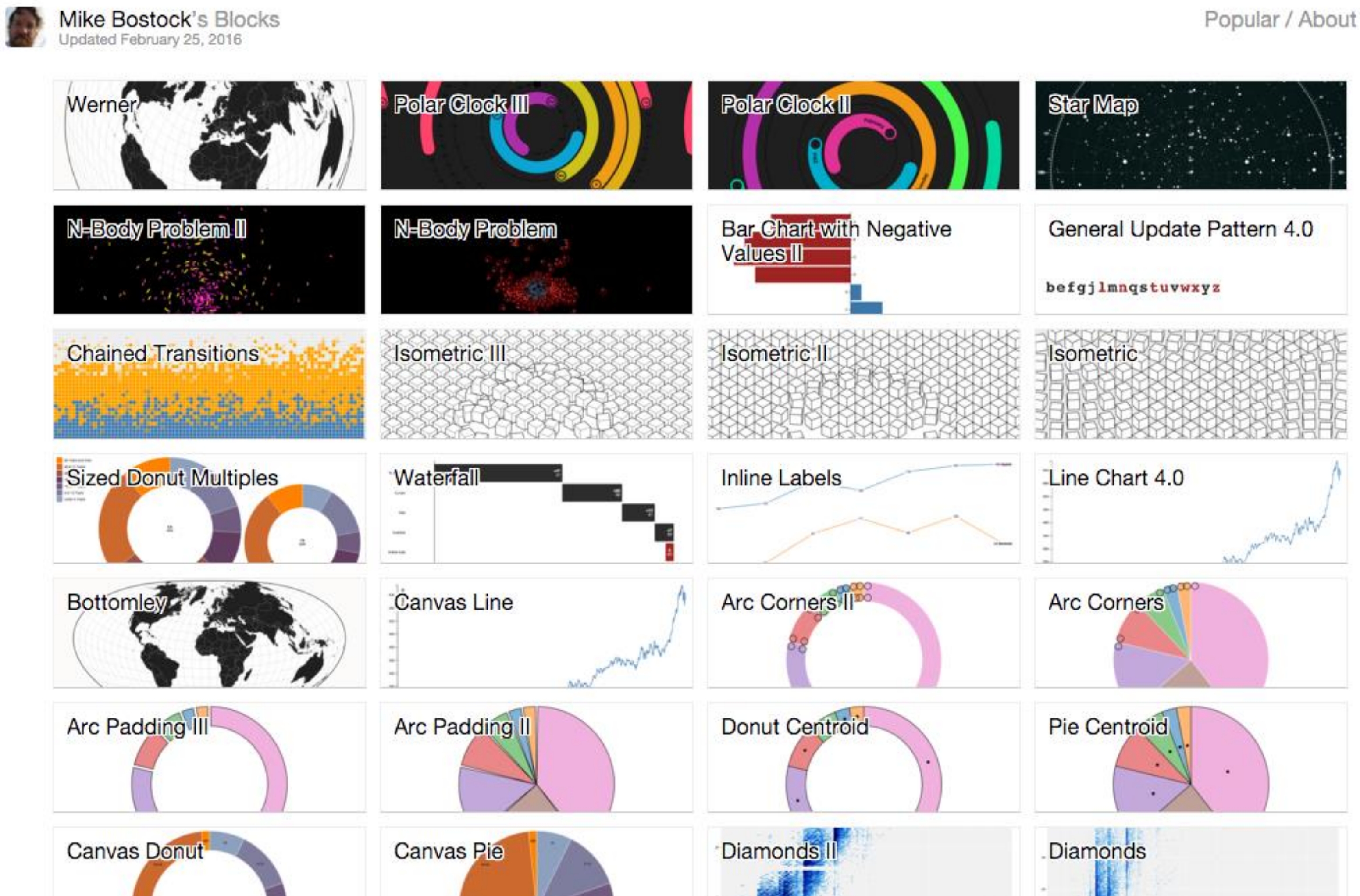


```
<!DOCTYPE html>
<style>

.chart div {
  font: 10px sans-serif;
  background-color: steelblue;
  text-align: right;
  padding: 3px;
  margin: 1px;
  color: white;
}

</style>
<div class="chart">
  <div style="width: 40px;">4</div>
  <div style="width: 80px;">8</div>
  <div style="width: 150px;">15</div>
  <div style="width: 160px;">16</div>
  <div style="width: 230px;">23</div>
  <div style="width: 420px;">42</div>
</div>
```



```
var data = [4, 8, 15, 16, 23, 42];

d3.select(".chart")
  .selectAll("div")
    .data(data)
  .enter().append("div")
    .style("width", function(d) { return d * 10 + "px"; })
    .text(function(d) { return d; });
```

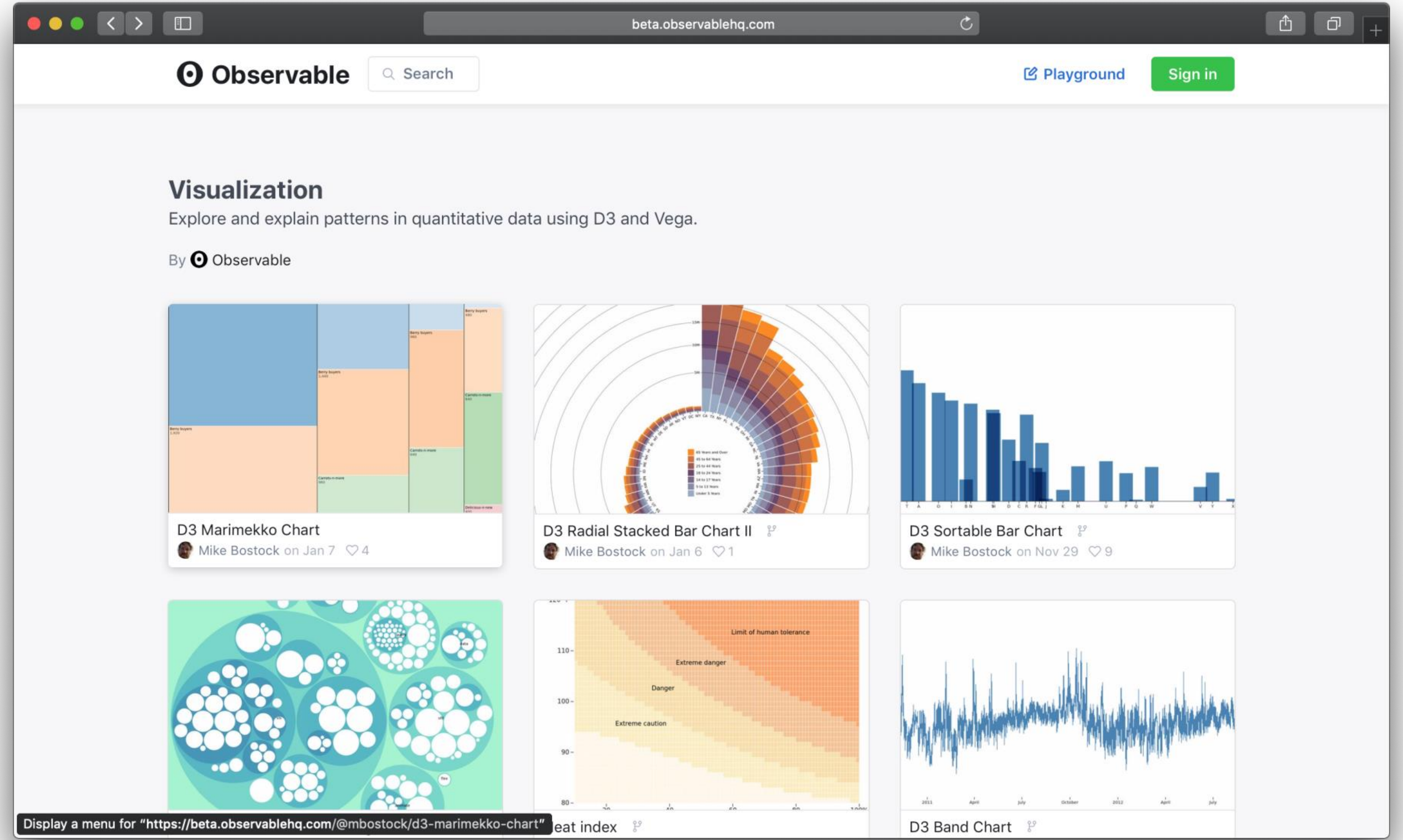
GREAT TUTORIALS & A HUGE LIBRARY OF EXAMPLES



Mike Bostock's Blocks
Updated February 25, 2016

Popular / About

Werner
Polar Clock III
Polar Clock II
Star Map
N-Body Problem II
N-Body Problem
Bar Chart with Negative Values II
General Update Pattern 4.0
Chained Transitions
Isometric III
Isometric II
Isometric
Sized Donut Multiples
Waterfall
Inline Labels
Line Chart 4.0
Bottomley
Canvas Line
Arc Corners II
Arc Corners
Arc Padding III
Arc Padding II
Donut Centroid
Pie Centroid
Canvas Donut
Canvas Pie
Diamonds II
Diamonds



Visualization

Explore and explain patterns in quantitative data using D3 and Vega.

By Observable

D3 Marimekko Chart
Mike Bostock on Jan 7 ♡ 4

D3 Radial Stacked Bar Chart II
Mike Bostock on Jan 6 ♡ 1

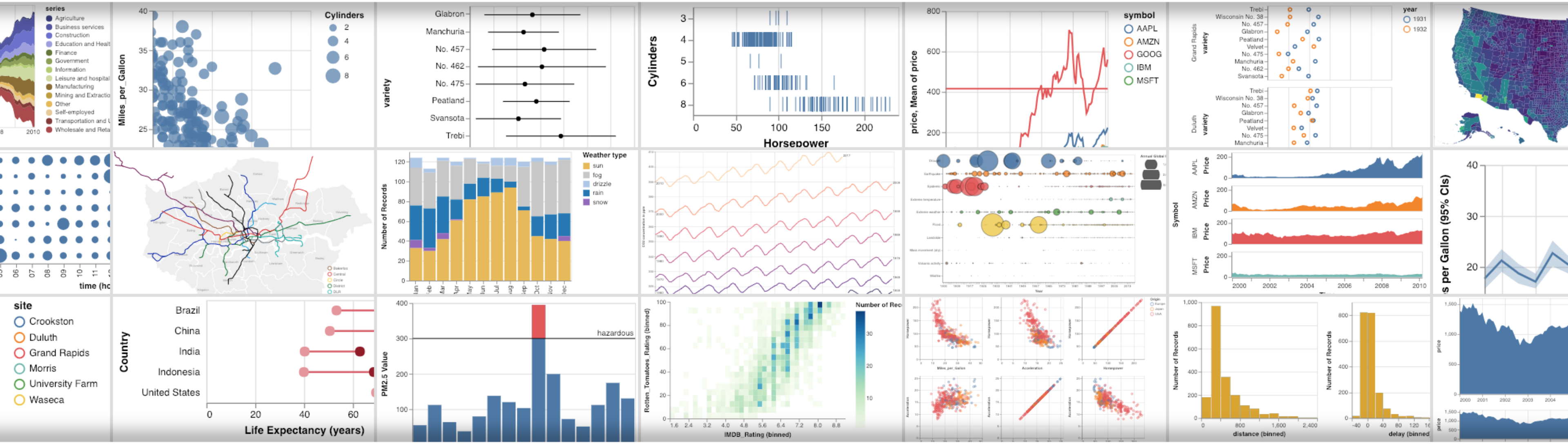
D3 Sortable Bar Chart
Mike Bostock on Nov 29 ♡ 9

eat index

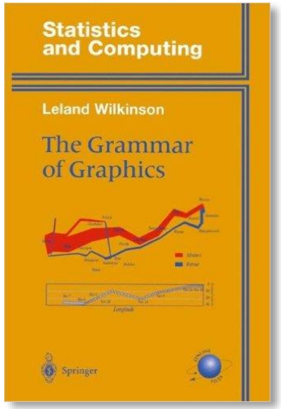
D3 Band Chart

Display a menu for "https://beta.observablehq.com/@mbostock/d3-marimekko-chart"

Vega-Lite – A Grammar of Interactive Graphics



Vega-Lite is a high-level grammar of interactive graphics. It provides a concise JSON syntax for rapidly generating visualizations to support analysis. Vega-Lite specifications can be compiled to Vega specifications.


Statistics and Computing

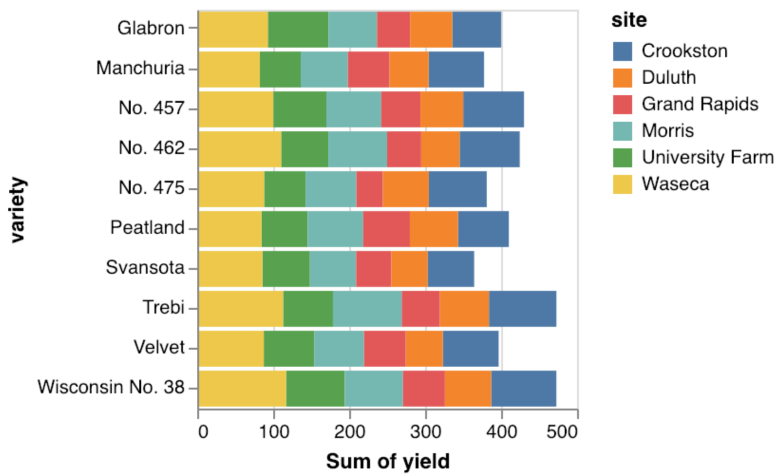
Leland Wilkinson

The Grammar of Graphics

Springer

A simple, powerful JSON syntax for authoring interactive visualizations inspired by Wilkinson's Grammar of Graphics.

Horizontal Stacked Bar Chart

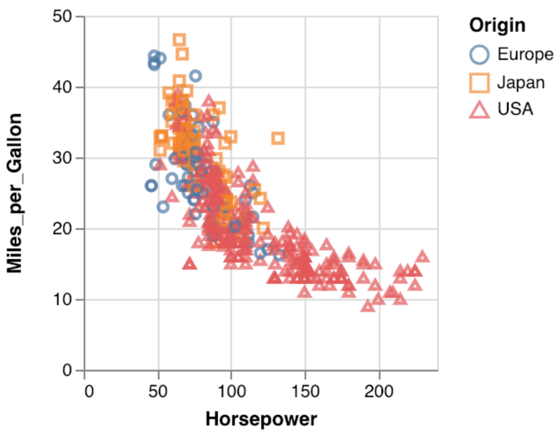


View this example in the online editor

Vega-Lite JSON Specification

```
{
  "$schema": "https://vega.github.io/schema/vega-lite/v3.json",
  "data": {"url": "data/barley.json"},
  "mark": "bar",
  "encoding": {
    "x": {"aggregate": "sum", "field": "yield", "type": "quantitative"}
    "y": {"field": "variety", "type": "nominal"},
    "color": {"field": "site", "type": "nominal"}
  }
}
```

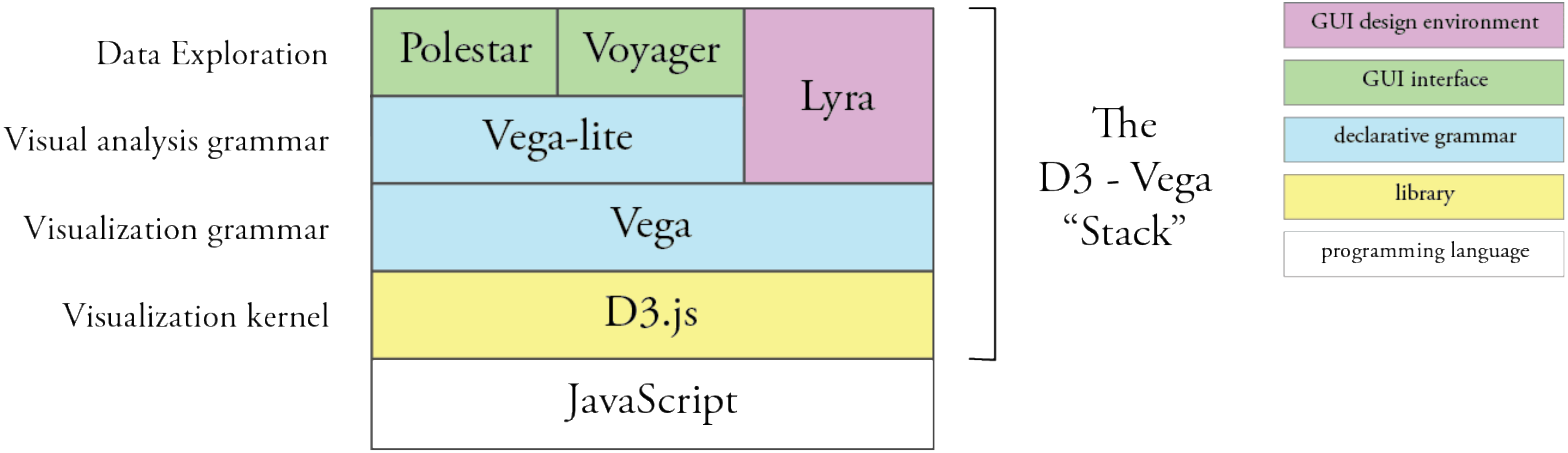
Colored Scatterplot



View this example in the online editor

Vega-Lite JSON Specification

```
{
  "$schema": "https://vega.github.io/schema/vega-lite/v3.json",
  "description": "A scatterplot showing horsepower and miles per
gallons.",
  "data": {"url": "data/cars.json"},
  "mark": "point",
  "encoding": {
    "x": {"field": "Horsepower", "type": "quantitative"},
    "y": {"field": "Miles_per_Gallon", "type": "quantitative"},
    "color": {"field": "Origin", "type": "nominal"},
    "shape": {"field": "Origin", "type": "nominal"}
  }
}
```



Data Exploration

Polestar

Voyager

Lyra

Visual analysis grammar

Vega-lite

Visualization grammar

Vega

Visualization kernel

D3.js

JavaScript

The D3 - Vega "Stack"

🟪 GUI design environment

🟩 GUI interface

🟦 declarative grammar

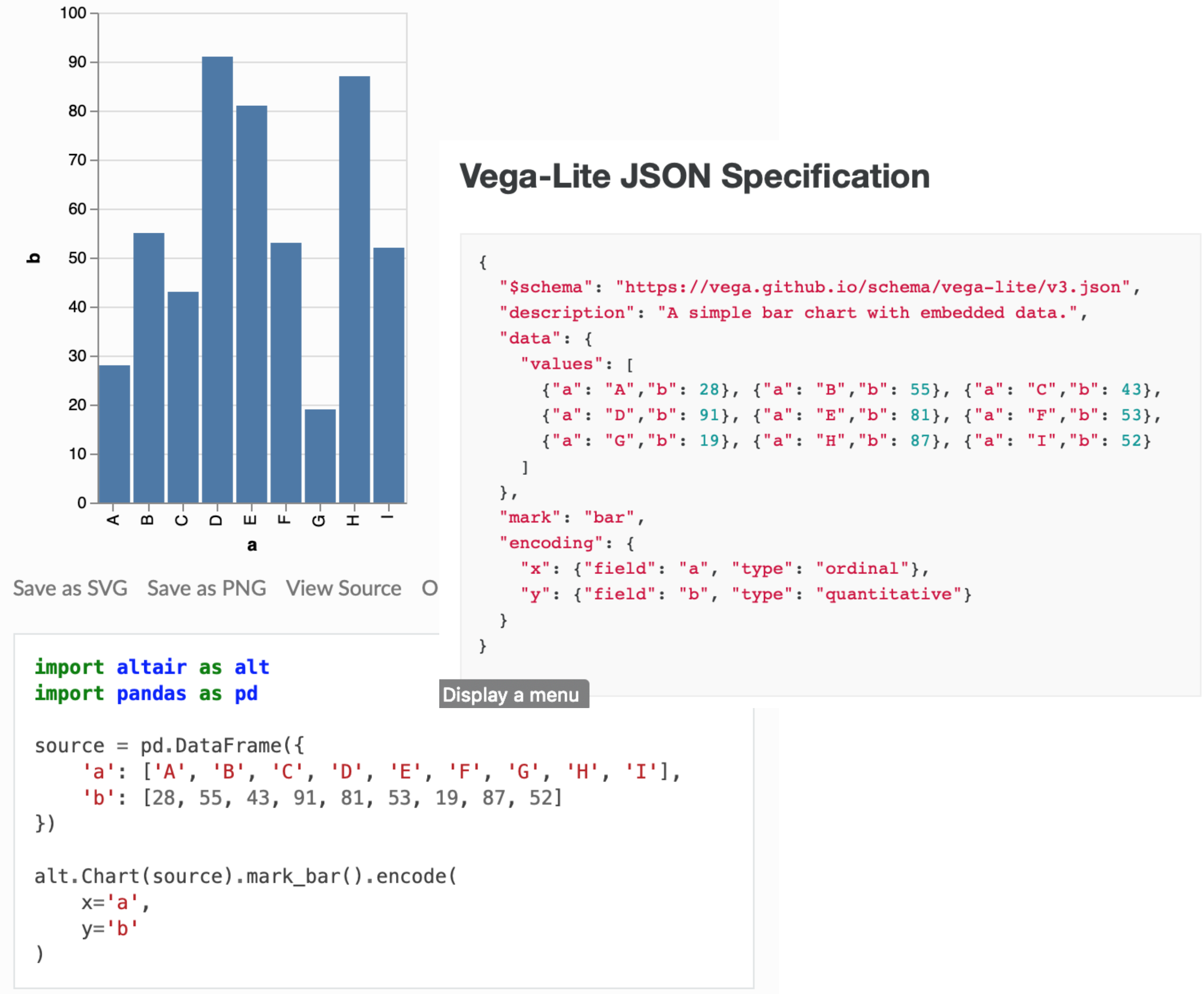
🟨 library

□ programming language

Altair

Python wrappers
for Vega-Lite!

Works with Pandas,
Jupyter, etc.



Vega-Lite JSON Specification

```json
{
  "$schema": "https://vega.github.io/schema/vega-lite/v3.json",
  "description": "A simple bar chart with embedded data.",
  "data": {
    "values": [
      {"a": "A","b": 28}, {"a": "B","b": 55}, {"a": "C","b": 43},
      {"a": "D","b": 91}, {"a": "E","b": 81}, {"a": "F","b": 53},
      {"a": "G","b": 19}, {"a": "H","b": 87}, {"a": "I","b": 52}
    ]
  },
  "mark": "bar",
  "encoding": {
    "x": {"field": "a", "type": "ordinal"},
    "y": {"field": "b", "type": "quantitative"}
  }
}
```

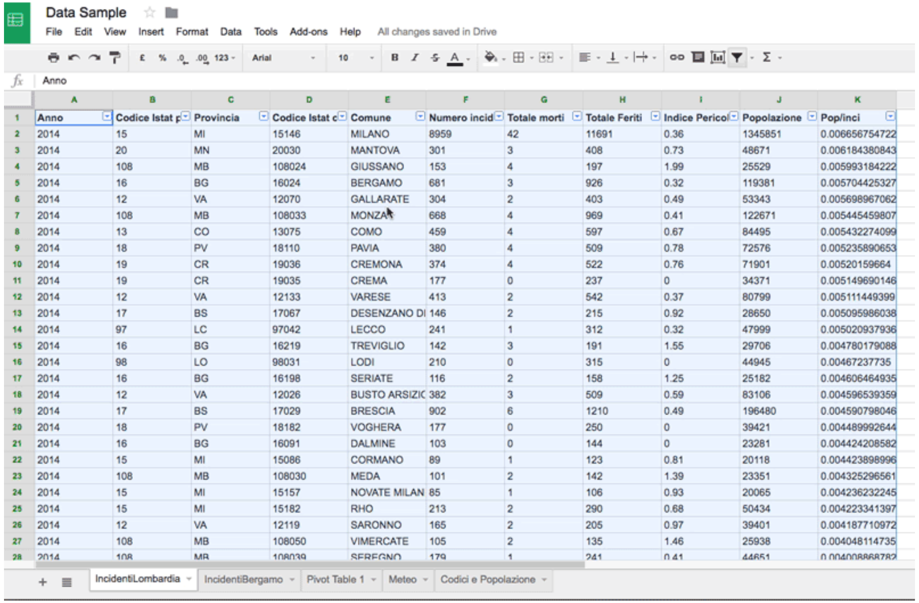
```python
import altair as alt
import pandas as pd

source = pd.DataFrame({
    'a': ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I'],
    'b': [28, 55, 43, 91, 81, 53, 19, 87, 52]
})

alt.Chart(source).mark_bar().encode(
    x='a',
    y='b'
)
```

OTHER USEFUL LIBRARIES

RAWGraphs

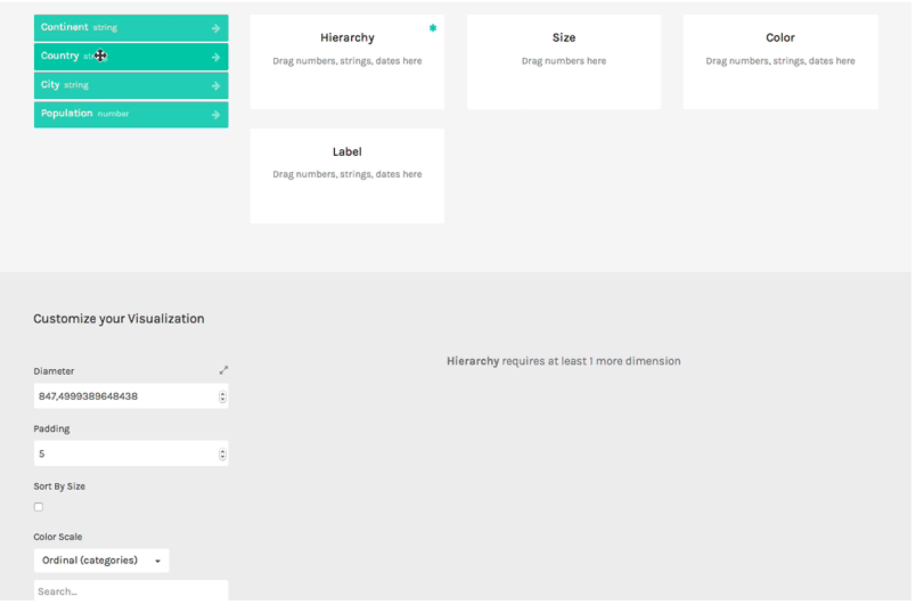
The missing link between spreadsheets and data visualization.

IEEE Visualization Dates - Google Sheets


Data Sample
File Edit View Insert Format Data Tools Add-ons Help All changes saved in Drive
£ % .0 .00 123 Arial 10 B I S A Σ

fx Anno

	A	B	C	D	E	F	G	H	I	J	K
1	Anno	Codice Istat p	Provincia	Codice Istat c	Comune	Numero incid	Totale morti	Totale Feriti	Indice Pericol	Popolazione	Pop/inci
2	2014	15	MI	15146	MILANO	8959	42	11691	0.36	1345851	0.006656754722
3	2014	20	MN	20030	MANTOVA	301	3	408	0.73	48671	0.006184380843
4	2014	108	MB	108024	GIUSSANO	153	4	197	1.99	25529	0.005993184222
5	2014	16	BG	16024	BERGAMO	681	3	926	0.32	119381	0.005704425327
6	2014	12	VA	12070	GALLARATE	304	2	403	0.49	53343	0.005698967062
7	2014	108	MB	108033	MONZA	668	4	969	0.41	122671	0.005445459807
8	2014	13	CO	13075	COMO	459	4	597	0.67	84495	0.005432274099
9	2014	18	PV	18110	PAVIA	380	4	509	0.78	72576	0.005235890653
10	2014	19	CR	19036	CREMONA	374	4	522	0.76	71901	0.00520159664
11	2014	19	CR	19035	CREMA	177	0	237	0	34371	0.005149690146
12	2014	12	VA	12133	VARESE	413	2	542	0.37	80799	0.005111449399
13	2014	17	BS	17067	DESENZANO DI	146	2	215	0.92	28650	0.005095986038
14	2014	97	LC	97042	LECCO	241	1	312	0.32	47999	0.005020937936
15	2014	16	BG	16219	TREVIGLIO	142	3	191	1.55	29706	0.004780179088
16	2014	98	LO	98031	LODI	210	0	315	0	44945	0.00467237735
17	2014	16	BG	16198	SERIATE	116	2	158	1.25	25182	0.004606464935
18	2014	12	VA	12026	BUSTO ARSIZIC	382	3	509	0.59	83106	0.004596539359
19	2014	17	BS	17029	BRESCIA	902	6	1210	0.49	196480	0.004590798046
20	2014	18	PV	18182	VOGHERA	177	0	250	0	39421	0.004489992644
21	2014	16	BG	16091	DALMINE	103	0	144	0	23281	0.004424208582
22	2014	15	MI	15086	CORMANO	89	1	123	0.81	20118	0.004423898996
23	2014	108	MB	108030	MEDA	101	2	142	1.39	23351	0.004325296561
24	2014	15	MI	15157	NOVATE MILAN	85	1	106	0.93	20065	0.004236232245
25	2014	15	MI	15182	RHO	213	2	290	0.68	50434	0.004223341397
26	2014	12	VA	12119	SARONNO	165	2	205	0.97	39401	0.004187710972
27	2014	108	MB	108050	VIMERCATE	105	2	135	1.46	25938	0.004048114735
28	2014	108	MB	108039	SEREGNO	179	1	241	0.41	44651	0.004008868782


Continent string
Country str
City string
Population number
Hierarchy
Drag numbers, strings, dates here
Size
Drag numbers here
Color
Drag numbers, strings, dates here
Label
Drag numbers, strings, dates here

Customize your Visualization

Diameter: 847,4999389648438

Padding: 5

Sort By Size ☐

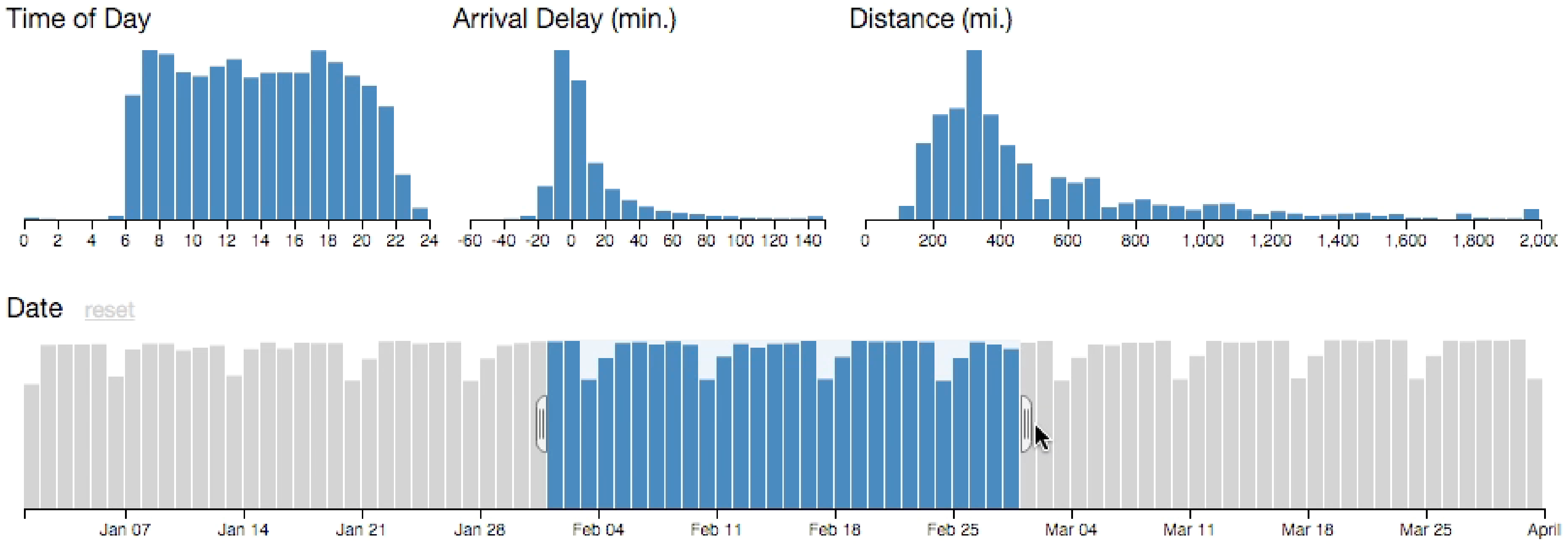
Color Scale: Ordinal (categories)

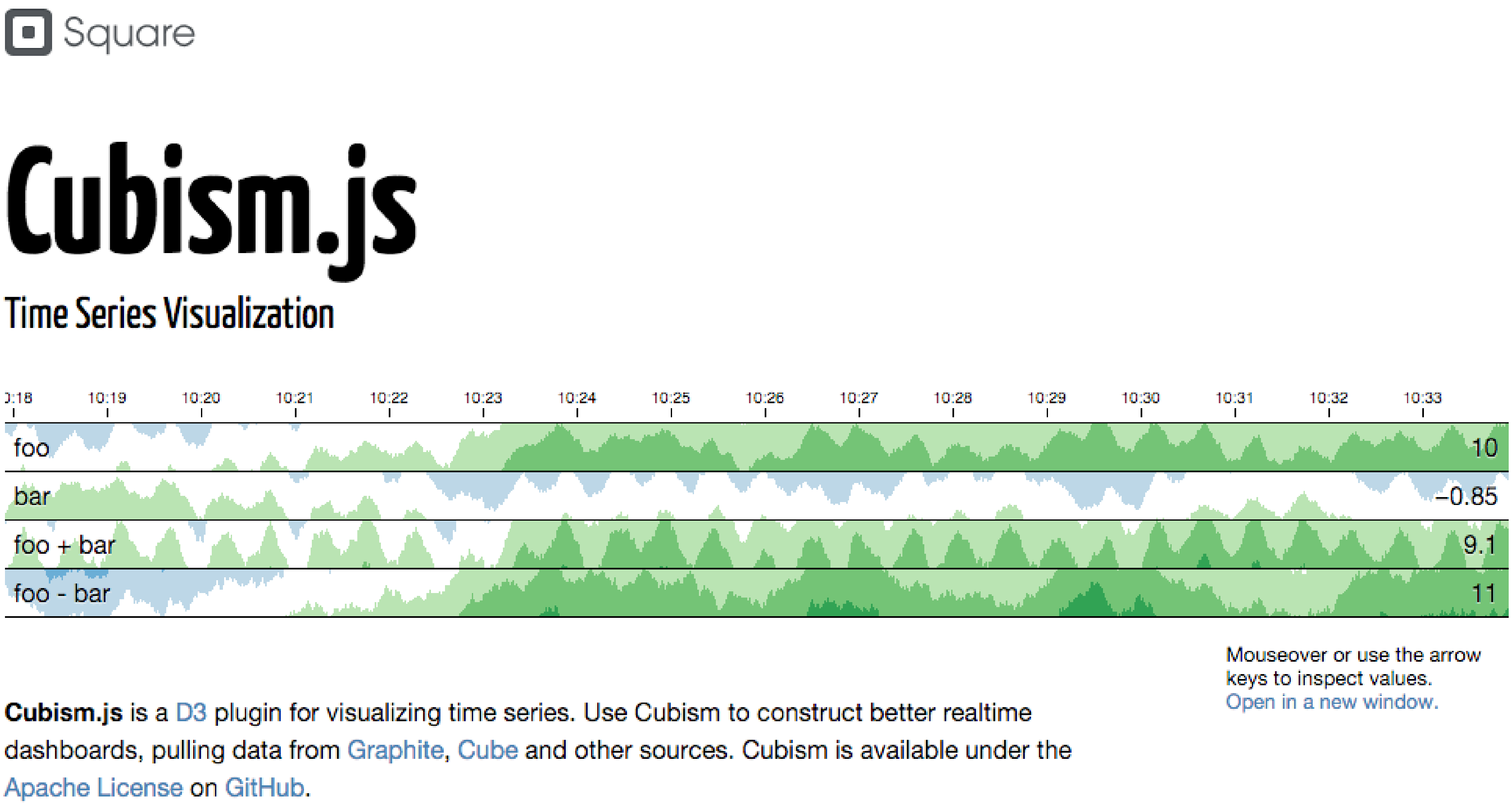
Search...

Hierarchy requires at least 1 more dimension

Square

Crossfilter





Square

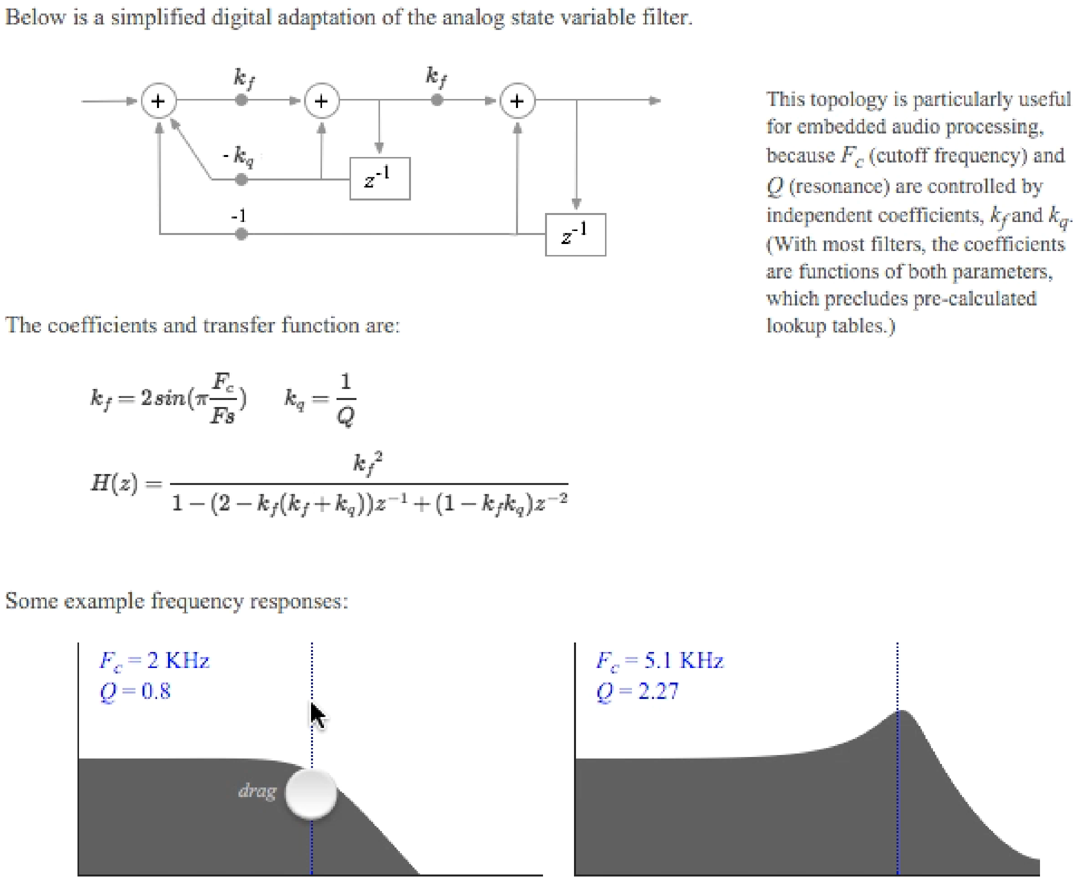
Cubism.js

Time Series Visualization

Mouseover or use the arrow keys to inspect values.
Open in a new window.

Cubism.js is a D3 plugin for visualizing time series. Use Cubism to construct better realtime dashboards, pulling data from Graphite, Cube and other sources. Cubism is available under the Apache License on GitHub.

Tangle http://worrydream.com/Tangle/



Below is a simplified digital adaptation of the analog state variable filter.

This topology is particularly useful for embedded audio processing, because F_c (cutoff frequency) and Q (resonance) are controlled by independent coefficients, k_f and k_q. (With most filters, the coefficients are functions of both parameters, which precludes pre-calculated lookup tables.)

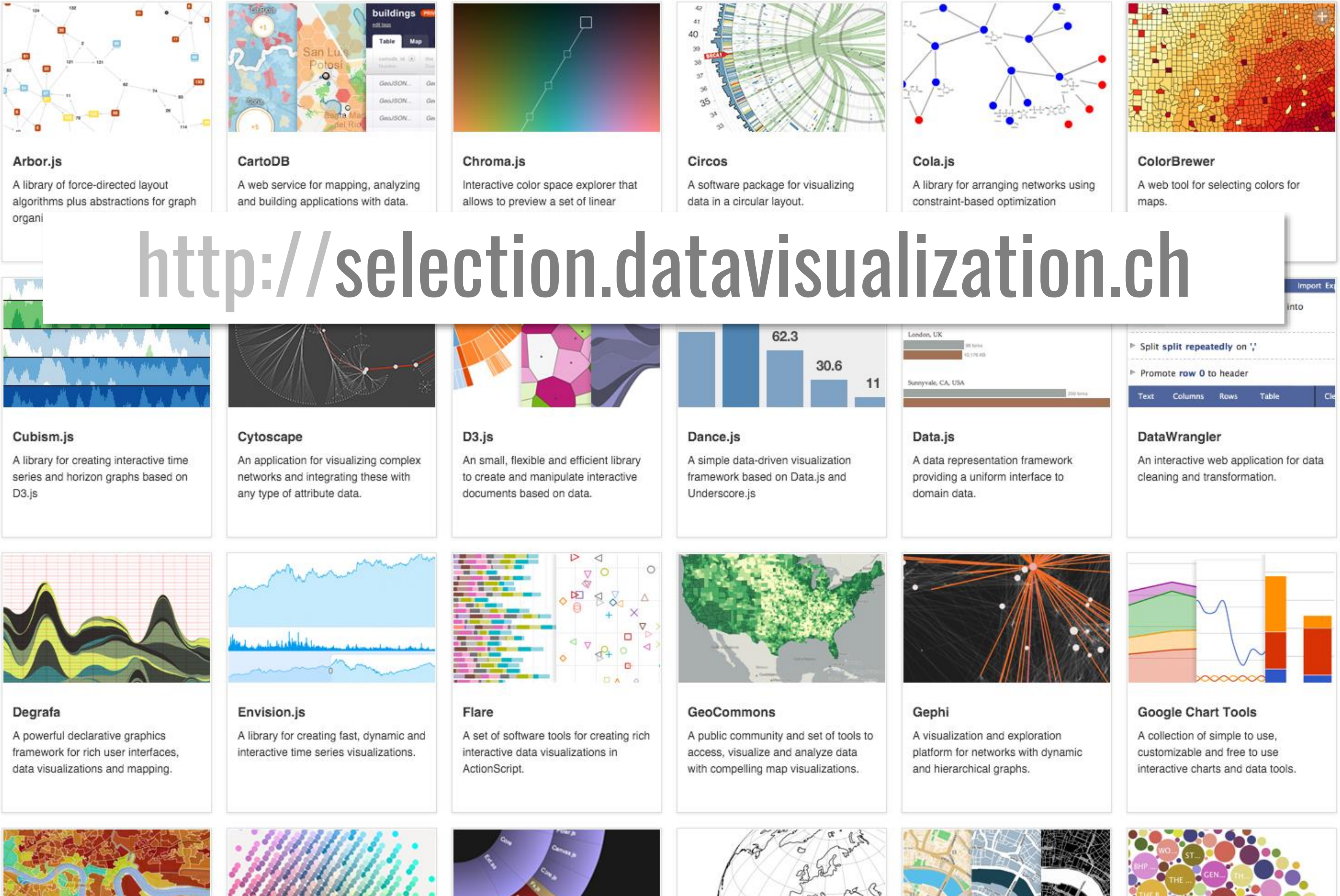
The coefficients and transfer function are:

$$k_f = 2 sin(\pi \frac{F_c}{Fs}) \qquad k_q = \frac{1}{Q}$$

$$H(z) = \frac{{k_f}^2}{1-(2-k_f(k_f+k_q))z^{-1}+(1-k_f k_q)z^{-2}}$$

Some example frequency responses:

http://selection.datavisualization.ch



Arbor.js

A library of force-directed layout algorithms plus abstractions for graph organi

CartoDB

A web service for mapping, analyzing and building applications with data.

Chroma.js

Interactive color space explorer that allows to preview a set of linear

Circos

A software package for visualizing data in a circular layout.

Cola.js

A library for arranging networks using constraint-based optimization

ColorBrewer

A web tool for selecting colors for maps.

Cubism.js

A library for creating interactive time series and horizon graphs based on D3.js

Cytoscape

An application for visualizing complex networks and integrating these with any type of attribute data.

D3.js

An small, flexible and efficient library to create and manipulate interactive documents based on data.

Dance.js

A simple data-driven visualization framework based on Data.js and Underscore.js

Data.js

A data representation framework providing a uniform interface to domain data.

DataWrangler

An interactive web application for data cleaning and transformation.

Degrafa

A powerful declarative graphics framework for rich user interfaces, data visualizations and mapping.

Envision.js

A library for creating fast, dynamic and interactive time series visualizations.

Flare

A set of software tools for creating rich interactive data visualizations in ActionScript.

GeoCommons

A public community and set of tools to access, visualize and analyze data with compelling map visualizations.

Gephi

A visualization and exploration platform for networks with dynamic and hierarchical graphs.

Google Chart Tools

A collection of simple to use, customizable and free to use interactive charts and data tools.

WHAT TO USE?

There are **many** different visualization tools available.

Need to balance tradeoffs:

- Expressiveness (Can I create the visualization I want/need?)
- Speed & Flexibility (How quickly can I generate, modify, and explore?)
- Reproducibility (Can I re-run the analysis? Re-generate the vis with new data?)
- Presentation (Can I style? Annotate? Share?)
- Interoperability (Can I integrate with other tools and applications?)

QUESTIONS?