ONLINE DATA VISUALIZATION

DR. DOMINIKUS BAUR
HTTPS://DO.MINIK.US

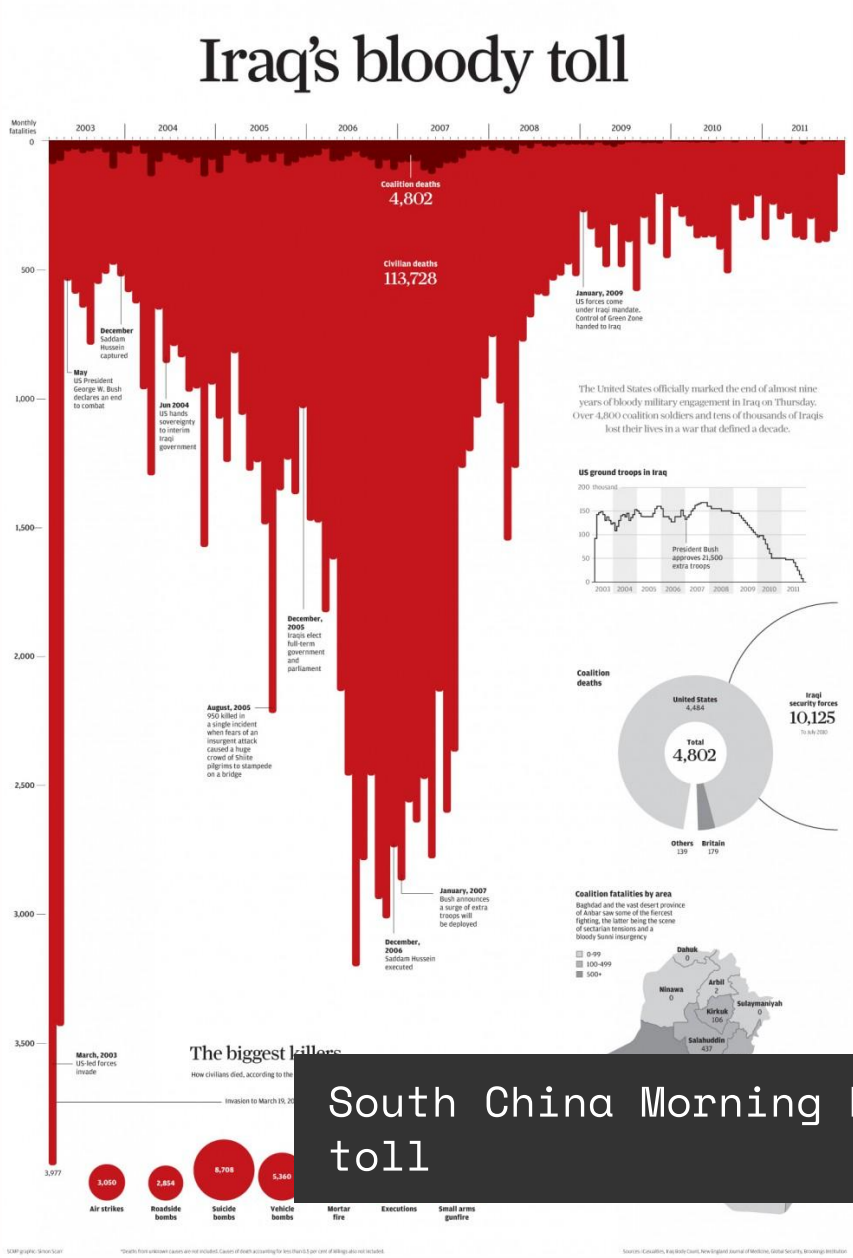
GRAPHICHUNTERS.NL

DAY 2

VISUALIZATION + COLOR

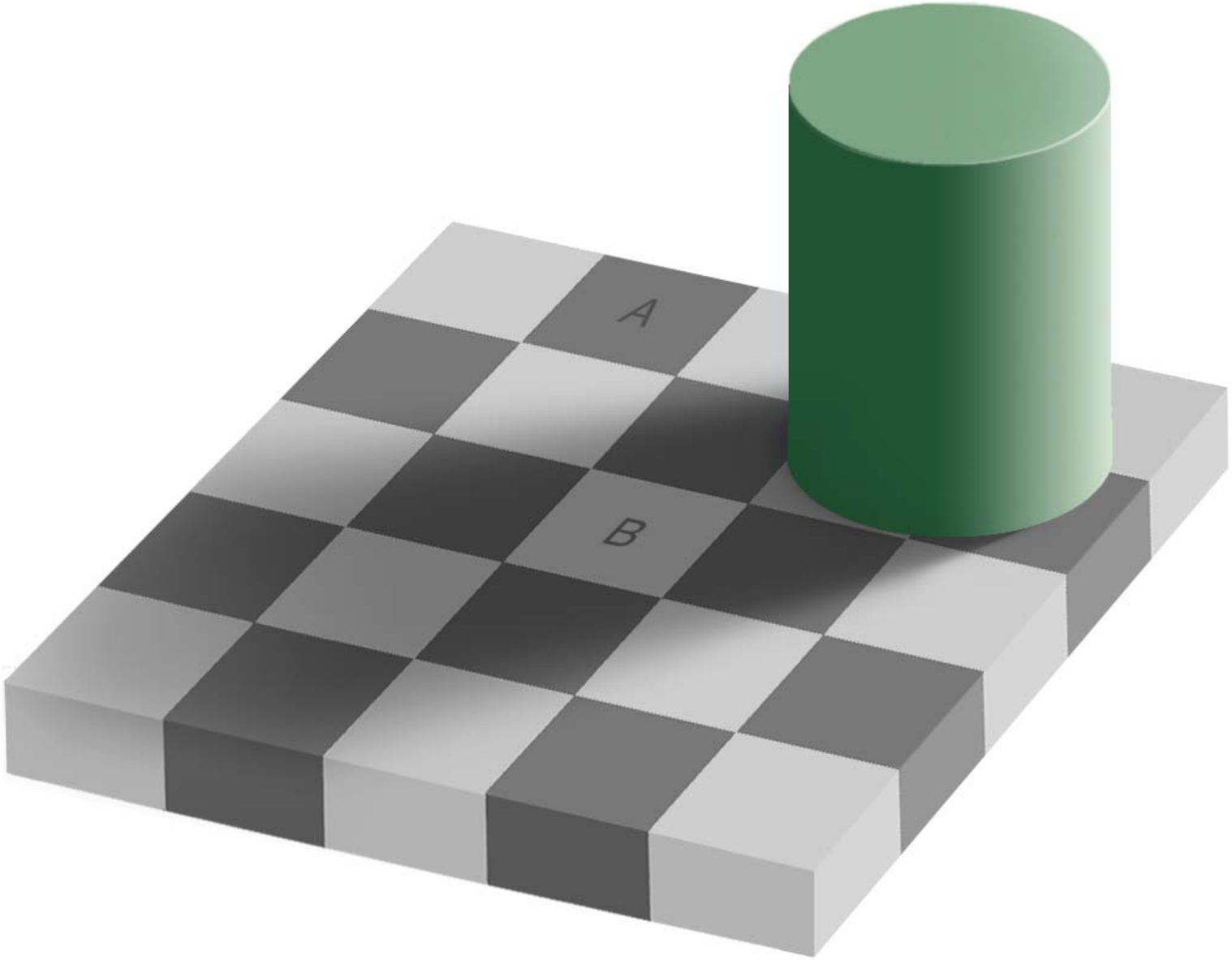
VISUALIZATION + COLOR

Colors are a critical, but a tough-to-do-right part of visualization.

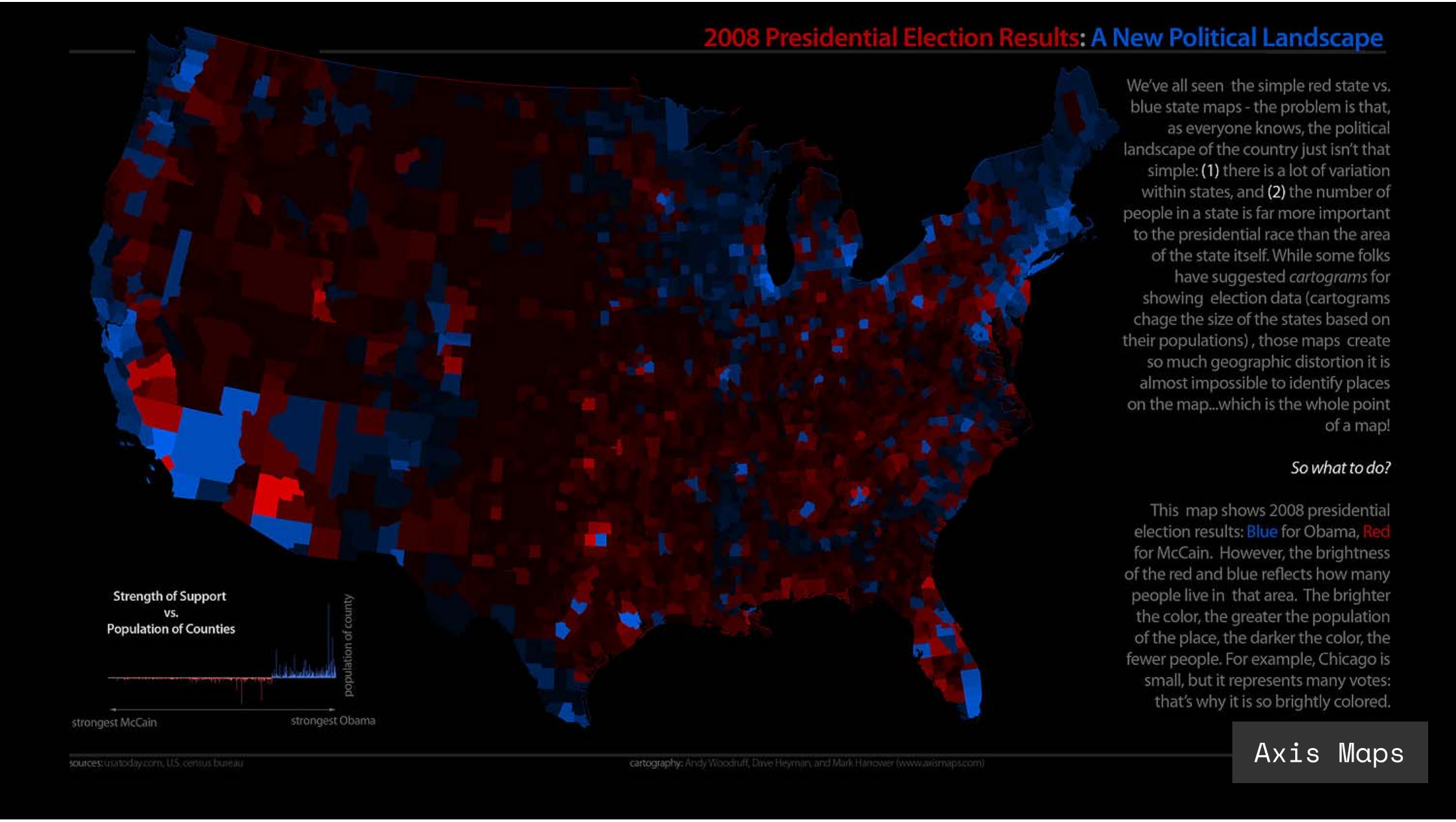


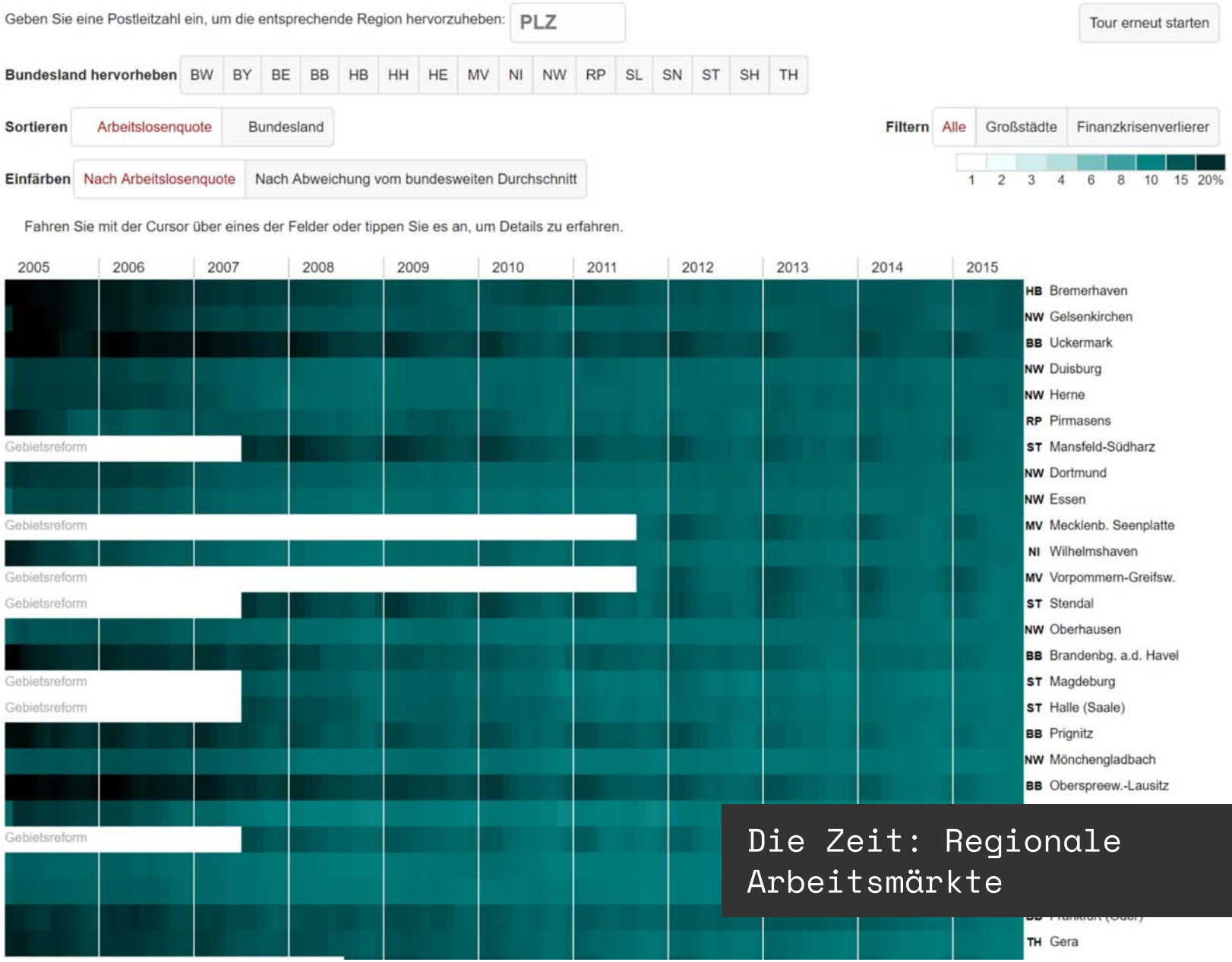
South China Morning Post: Iraq's bloody toll

Pitch Interactive: Out of Sight, Out of Mind

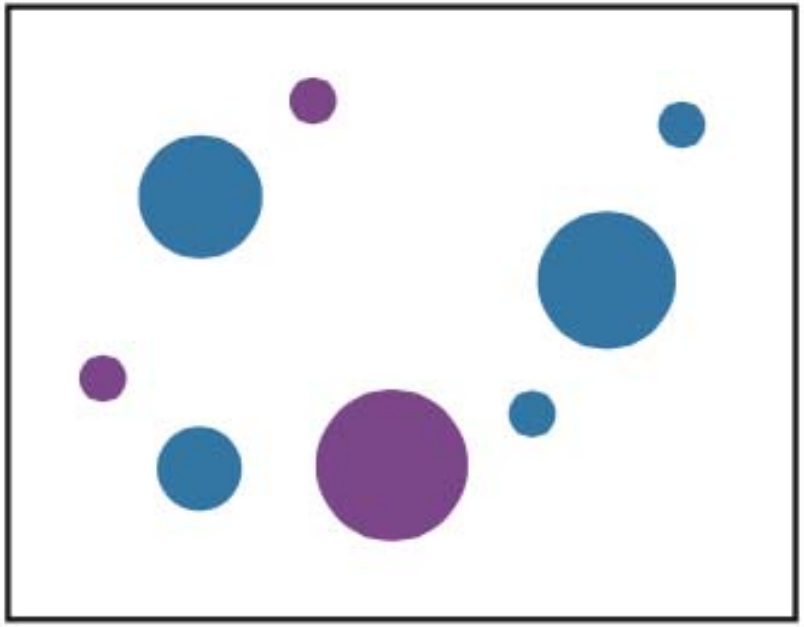


Wikipedia


2008 Presidential Election Results: A New Political Landscape
We've all seen the simple red state vs. blue state maps - the problem is that, as everyone knows, the political landscape of the country just isn't that simple: (1) there is a lot of variation within states, and (2) the number of people in a state is far more important to the presidential race than the area of the state itself. While some folks have suggested *cartograms* for showing election data (cartograms chage the size of the states based on their populations) , those maps create so much geographic distortion it is almost impossible to identify places on the map...which is the whole point of a map!
So what to do?
This map shows 2008 presidential election results: Blue for Obama, Red for McCain. However, the brightness of the red and blue reflects how many people live in that area. The brighter the color, the greater the population of the place, the darker the color, the fewer people. For example, Chicago is small, but it represents many votes: that's why it is so brightly colored.
Strength of Support
vs.
Population of Counties
population of county
strongest McCain
strongest Obama
sources: usatoday.com, U.S. census bureau
cartography: Andy Woodruff, Dave Heyman, and Mark Harrower (www.axismaps.com)
Axis Maps

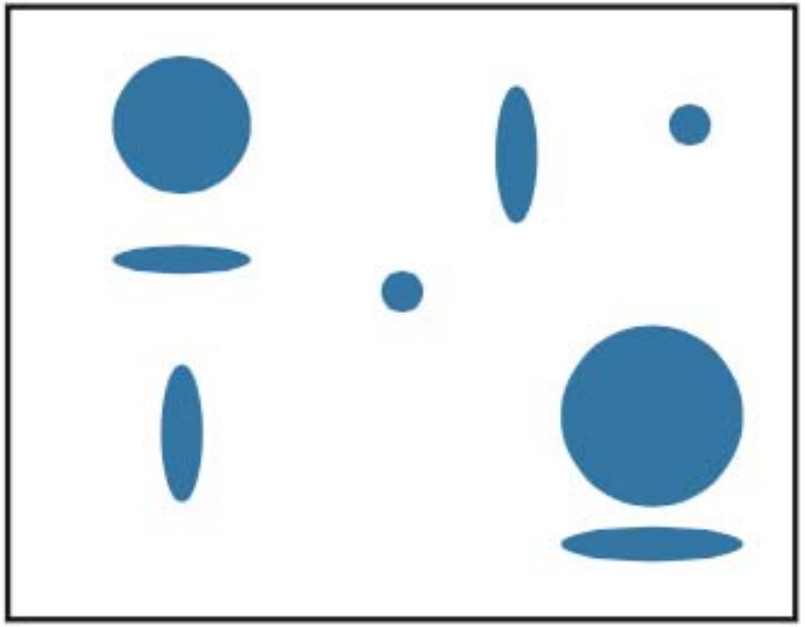

Geben Sie eine Postleitzahl ein, um die entsprechende Region hervorzuheben: PLZ
Tour erneut starten
Bundesland hervorheben BW BY BE BB HB HH HE MV NI NW RP SL SN ST SH TH
Sortieren Arbeitslosenquote Bundesland
Filtern Alle Großstädte Finanzkrisenverlierer
Einfärben Nach Arbeitslosenquote Nach Abweichung vom bundesweiten Durchschnitt
1 2 3 4 6 8 10 15 20%
Fahren Sie mit der Cursor über eines der Felder oder tippen Sie es an, um Details zu erfahren.
2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015
HB Bremerhaven
NW Gelsenkirchen
BB Uckermark
NW Duisburg
NW Herne
RP Pirmasens
ST Mansfeld-Südharz
NW Dortmund
NW Essen
MV Mecklenb. Seenplatte
NI Wilhelmshaven
MV Vorpommern-Greifsw.
ST Stendal
NW Oberhausen
BB Brandenbg. a.d. Havel
ST Magdeburg
ST Halle (Saale)
BB Prignitz
NW Mönchengladbach
BB Oberspreew.-Lausitz
Gebietsreform
Die Zeit: Regionale Arbeitsmärkte
TH Gera

Size
+ Hue (Color)



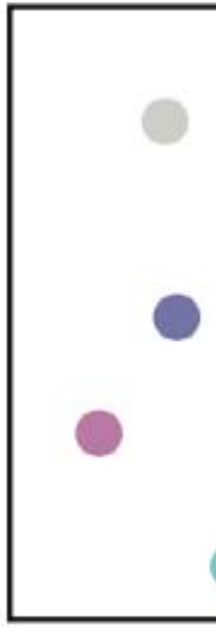
Some interference

Width
+ Height



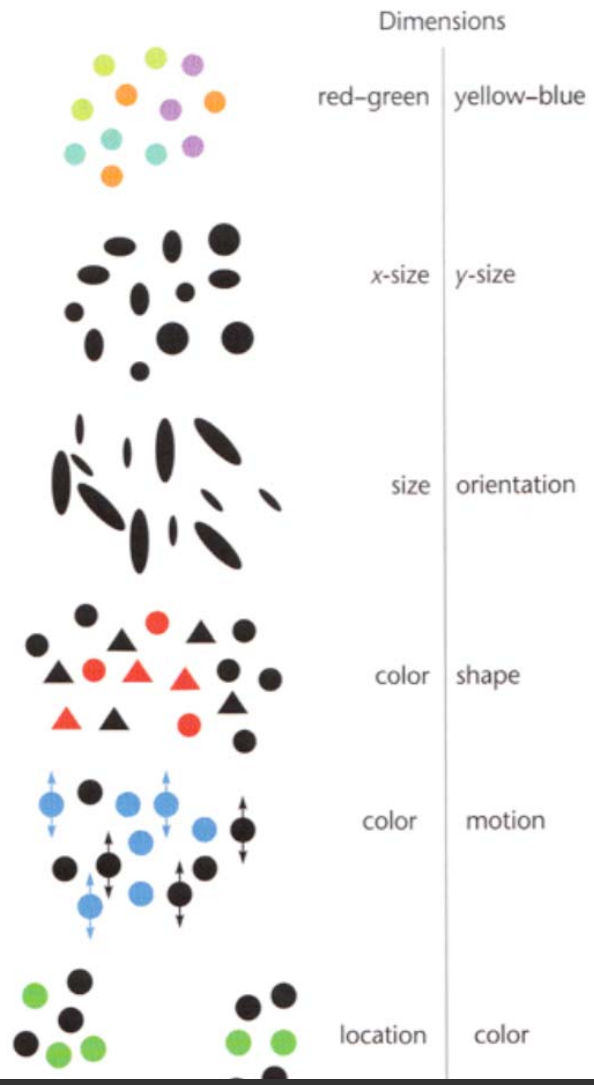
Some/significant

Red
+ Gre



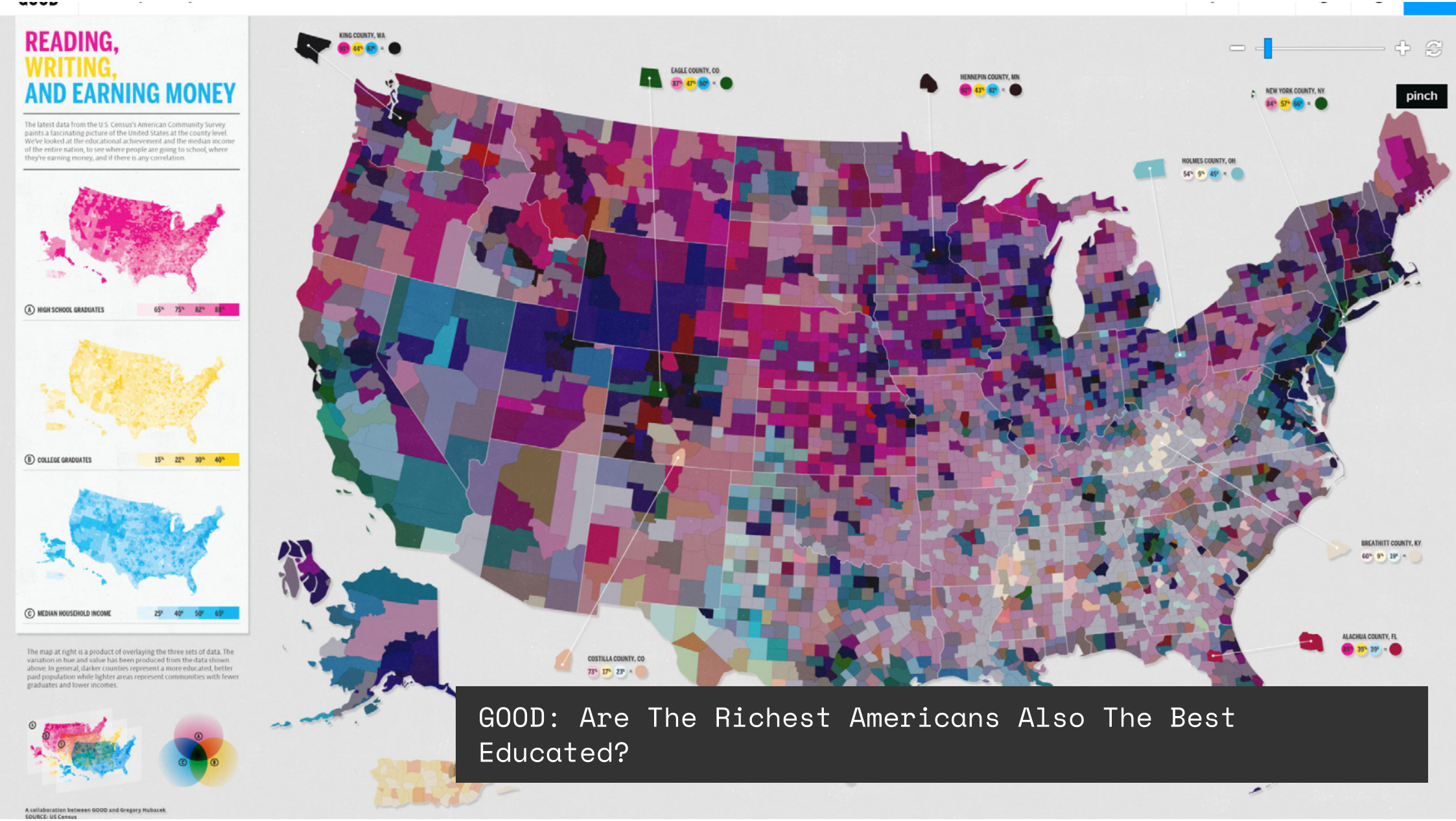
Majo

Tamara Munzner: Visualization Analysis And Design (p. 106)



Colin Ware: Visual Thinking for Design

Figure 5.31 Examples of glyphs coded according to two display attributes. At the top are more integral coding pairs. At the bottom are more separable coding pairs.


READING,
WRITING,
AND EARNING MONEY
The latest data from the U.S. Census's American Community Survey paints a fascinating picture of the United States at the county level. We've looked at the educational achievement and the median income of the entire nation, to see where people are going to school, where they're earning money, and if there is any correlation.
A HIGH SCHOOL GRADUATES
B COLLEGE GRADUATES
C MEDIAN HOUSEHOLD INCOME
The map at right is a product of overlaying the three sets of data. The variation in hue and value has been produced from the data shown above. In general, darker counties represent a more educated, better paid population while lighter areas represent communities with fewer graduates and lower incomes.
A collaboration between GOOD and Gregory Hubacek
SOURCE: US Census
KING COUNTY, WA
EAGLE COUNTY, CO
HENNEPIN COUNTY, MN
NEW YORK COUNTY, NY
HOLMES COUNTY, OH
BREATHITT COUNTY, KY
ALACHUA COUNTY, FL
COSTILLA COUNTY, CO
pinch
GOOD: Are The Richest Americans Also The Best Educated?

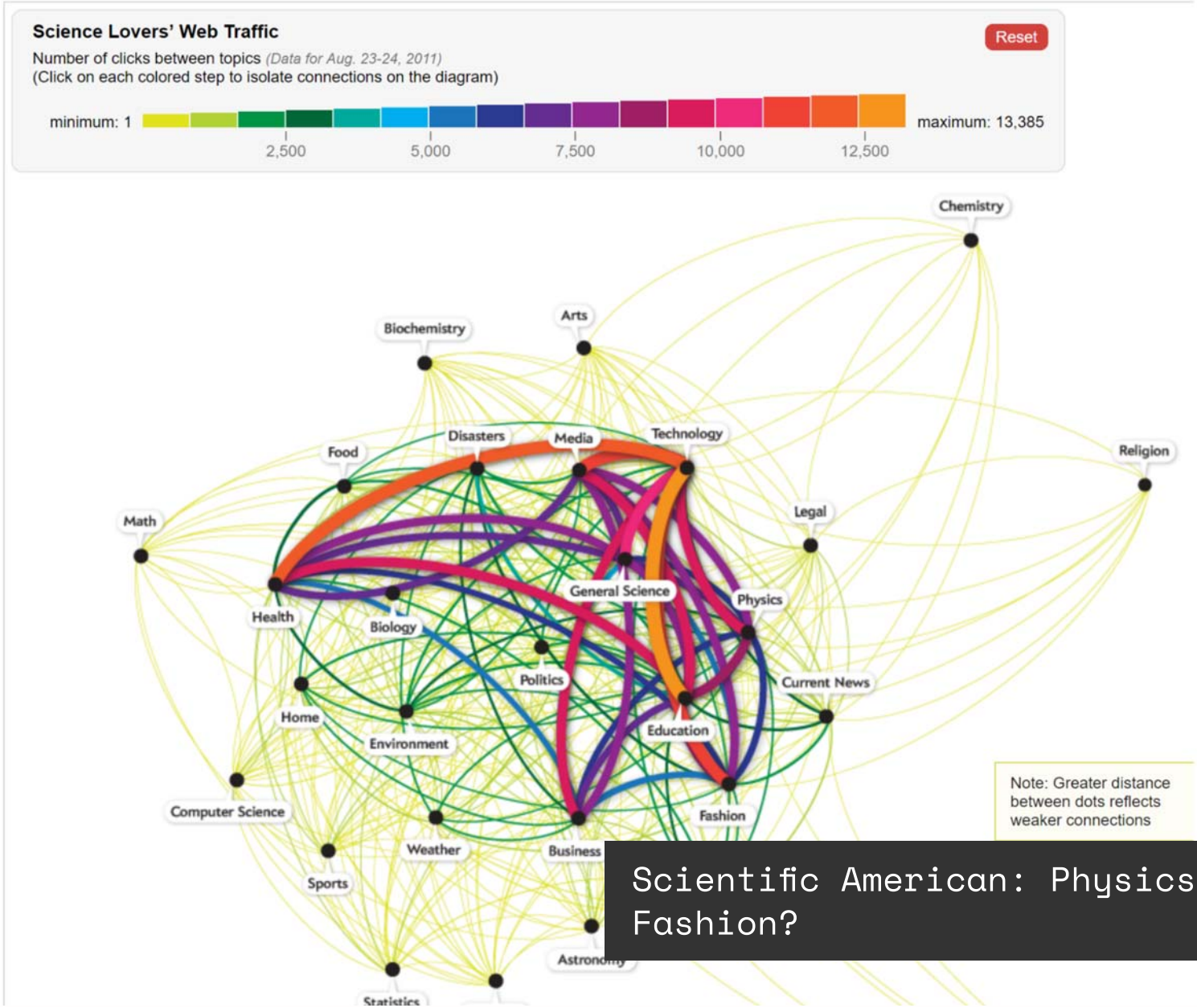
Cool idea, but:
It's not possible to mentally separate color channels!

Obama's Divided Nation

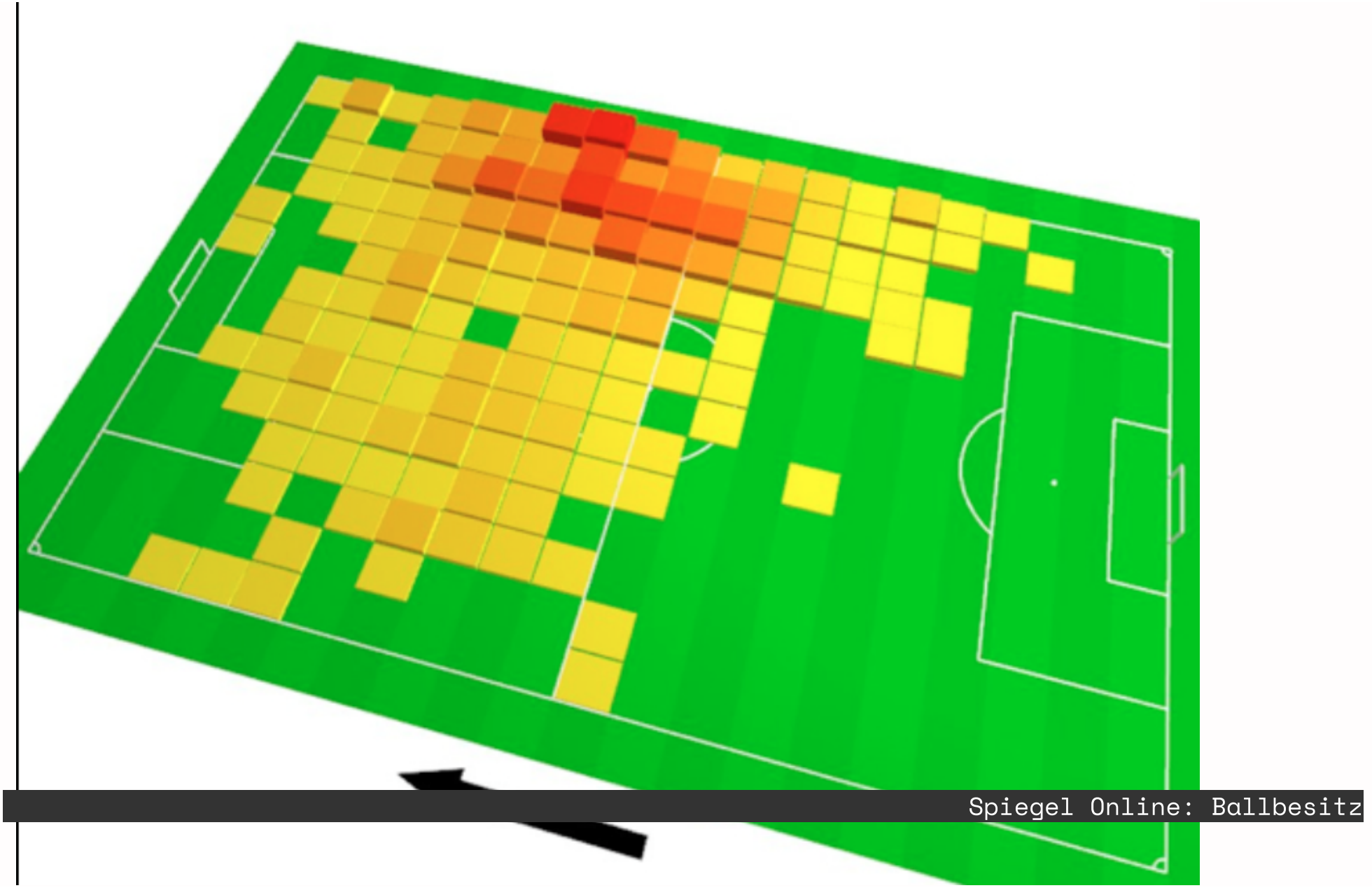
na presides over
a more divided
ny time in 50
y that was riven
racial lines gath-
008 to elect its
dent. That presi-
ur years dividing
the basis of eco-
he campaign re-
vealed no evi-
dence that Mr.
Obama will close
he chasm he has
reated between
is voters and
hose he attacked

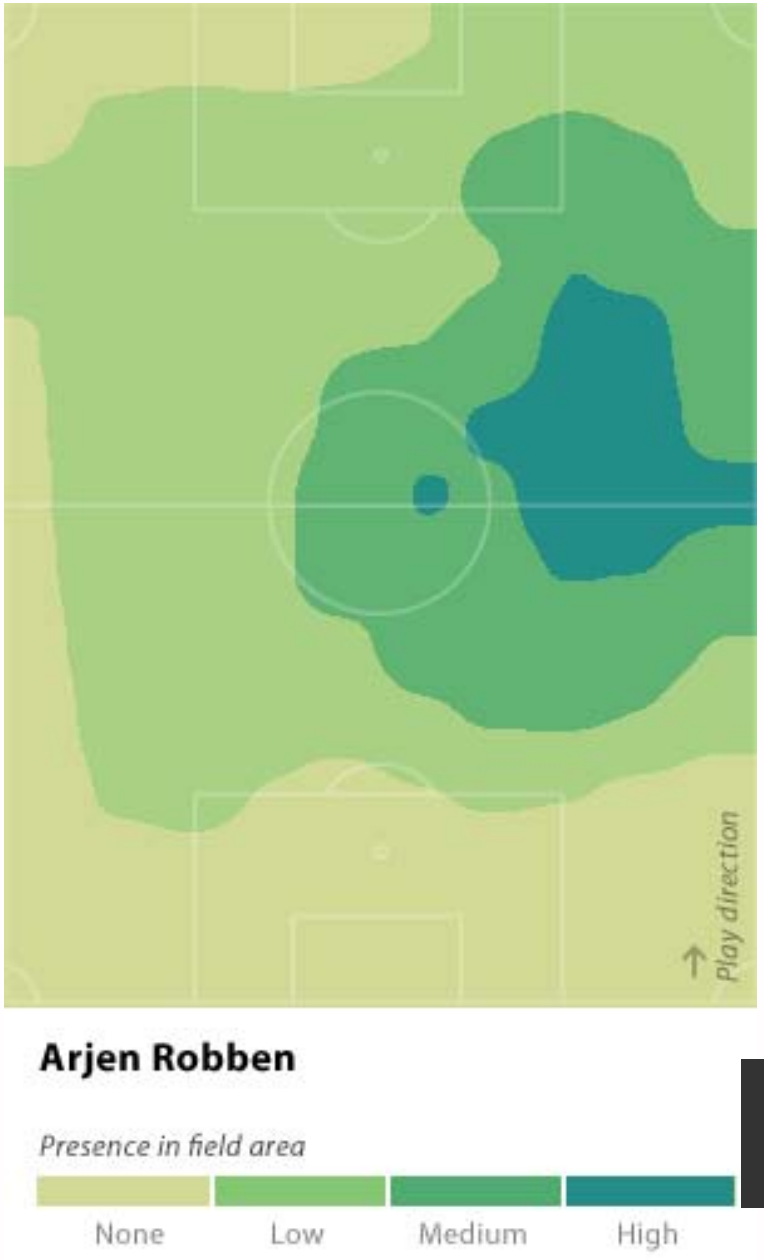
problem with pols,
verbally facile as M
that in crunch time
reverts to No. 1. E
that 9% of the elec
who to vote for ju
Tuesday; and an
42% said Mr. Ob
Sandy response—
tie photo-op—w
factor. Of those
voted for Mr.
Mr. Christie is
politico who is

Yes, Republ
across two pr
that there ar
how crudely



Scientific American: Physics Or Fashion?


Spiegel Online: Ballbesitz



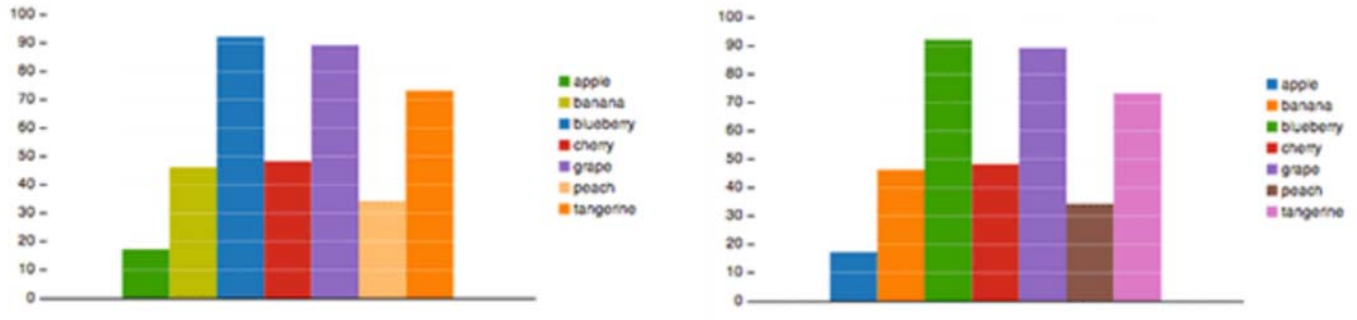
Moritz Stefaner:
Redesign

SEMANTICALLY RESONANT COLORS

STANFORD VIS GROUP HOME PAPERS PEOPLE VIDEO

Selecting Semantically-Resonant Colors for Data Visualization

Sharon Lin, Julie Fortuna, Chinmay Kulkarni, Maureen Stone, Jeffrey Heer



Bar charts depicting fictional fruit sales, each using the same backing color palette. The chart on the left uses our semantically-resonant assignment algorithm to pick colors that are representative of the data values. The chart on the right uses a default assignment that does not take color-concept associations into account.

ABSTRACT

We introduce an algorithm for automatic selection of semantically-resonant colors to represent data (e.g., using blue for data about "oceans", or pink for "love"). Given a set of categorical values and a target color palette, our algorithm matches each data value with a unique color. Values are mapped to colors by collecting representative images, analyzing image color distributi
balances the probabi
chosen semantically-
algorithm selects col
across a variety of data categories.

Stanford Vis Group: Selecting Semantically-Resonant Colors

TIPS

Get it right in Black + White.

Use color mostly for categories or highlights.

If used for quantities use mostly variation in brightness.

If used for diverging scales, blend over a neutral color.

TOOLS

TOOLS

ColorBrewer

TOOLS

Color Blindness Simulator

TOOLS

Adobe Color CC

TOOLS

IWantHue

MORE ON COLOR:

Stephen Few: Practical Rules for Using Color in Charts

Lisa Rost: Your Friendly Guide to Colors in Data Visualisation

COLOR IN D3

COLOR IN D3

d3-color

COLOR IN D3

```
d3.schemeCategory10
d3.schemeCategory20
d3.schemeCategory20b
d3.schemeCategory20c
```

```
d3-
scale#schemeCategory10
```

COLOR IN D3

You can find predefined color scales in d3-scale-chromatic.

```
<script src="https://d3js.org/d3-scale-chromatic.v1
```

D3 INTERACTION

D3 INTERACTION

d3 comes with varied support for interaction: event handlers and more specific tools for brushing, zooming...

BASIC INTERACTION

BASIC INTERACTION

d3's *.on* function ties DOM-event handlers to selections.

```
var circles = svg.selectAll('circle')
  .on('mouseover', function(d){
    console.log(this);
    console.log(d);
  });
```

BASIC INTERACTION

Let's try it on: code/d3-interaction/01_bitcoin_vis.html (or use your own!)

First, console.log elements that have been interacted with. Then, highlight them.

BASIC INTERACTION

Useful DOM-events:

Mozilla: DOM-events

- click
- mouseover
- mousemove
- mouseout
- touchstart
- touchmove
- touchend

BASIC INTERACTION

Sometimes very useful: *pointer-events: none* (CSS)

TOOLTIPS

Most common type of interaction: tooltips!
Not directly supported in d3, but easy to build yourself:

d3noob: Simple d3.js tooltips

Or use qTip2 or one of the other 75.000 tooltip libraries out there...

MORE ON TRANSITIONS

MORE ON TRANSITIONS

Transitions can be tweaked using:

```
var t = d3.transition();
// add initial delay:
t.delay(500);
// change duration:
t.duration(5000);
// add event handlers:
t.on("end", function(d){ ... });
```

MORE ON TRANSITIONS

Want to be really fancy? Change the easings on the transitions!

```
d3.transition().ease(d3.easeBounceInOut)
```

d3-ease

d3: Life of a transition

BASIC INTERACTION

Let's try it on: code/d3-interaction/02_more_transitions.html

ADVANCED INTERACTION

ADVANCED INTERACTION

d3 has several specific *behaviors* supporting brushing, dragging, ...

DRAG

DRAG

Read up on d3-drag and d3.event, then use the basic d3-template from code/d3-basics/template.html to:

Draw some random circles on screen. Then make them draggable.

ZOOM

ZOOM

d3's zoom makes handling zoom+pan easier.

d3-zoom

ZOOM

Let's look at examples:

Mike Bostock: Pan & Zoom II

Mike Bostock: Pan & Zoom Axes

BRUSHES

BRUSHES

Brushes let you select parts of a chart:

Selfiecity: Selfiexploratory

BRUSHES

d3 (of course) has a helper function for that: *d3.brush()*.

d3-brush

BRUSHES

Go back to your circle-example from before. Instead of dragging them, add a brush and highlight them when they're inside the brush.

BRUSHES

Another cool brushing demo:

Mike Bostock: Mona Lisa Histogram

COMBINING BEHAVIORS

You can also combine d3 behaviors:

Mike Bostock: Brush & Zoom

(even more advanced - enjoy :)

MORE TOOLS:

d3-annotation by Susie Lu

D3 SHAPES

D3 SHAPES

d3 comes with various shape generators for arcs, pies, lines, areas, ...

Basically everything that's missing from SVG.

LINES

LINES

SVG paths:

```
<path d="M 10 10 L 300 200 l -50 -50"></path>
```

LINES

d3's line generator can be used to generate those nasty *d* attributes.

```
var line = d3.line();

svg.append('path')
  .attr('d', line(data));
```

LINES

Have a look at the docs:

d3-shape#lines

... and turn our bitcoin plot into a line chart!

CURVES

CURVES

SVG paths can also be used to draw curves:

```
<path d="M10 80 Q 95 10 180 80"
  stroke="black" fill="transparent"/>
```

CURVES

d3 - again - has some handy generators for curves/arcs: *d3.arc()*

```
var arc = d3.arc();

arc({
  innerRadius: 0,
  outerRadius: 100,
  startAngle: 0,
  endAngle: Math.PI / 2
});
```

d3-shape#arc

CURVES

Use the d3-template and d3.arc to generate random arcs across a site!

CANVAS

CANVAS

As a small aside: let's talk about rendering technologies.

CANVAS

Canvas is a bitmap-based alternative to SVG's vector format. It is not scalable and opaque.
But it's usually faster!

CANVAS

```
var canvas = document.querySelector('canvas');
var ctx = canvas.getContext('2d');
```

CANVAS

Canvas uses sequential commands with an implicit state for drawing:

```
ctx.fillStyle = "rgb(200,0,0)";
ctx.fillRect(10, 10, 55, 50);
```

Mozilla:
CanvasRenderingContext2D

CANVAS

Examples:

Stroke rect

Drawing text

Drawing paths

CANVAS IN D3

CANVAS IN D3

Since v4.0, canvas is a common option in d3-rendering functions.

Use the *.context()* function for shape generators to set a canvas context to automatically render to.

CANVAS IN D3

```
var canvas = document.querySelector('canvas');
var ctx = canvas.getContext('2d');
var line = d3.line().context(ctx);

ctx.beginPath();
line(data);
...
```

CANVAS IN D3

Use canvas instead of SVG to draw the bitcoin line chart or the random arcs!

PIE CHARTS

PIE CHARTS

Let's do everyone's favorite: the pie chart.

d3 has a pie generator (*d3.pie()*) whose result can be plugged into d3.arc.

d3-shape#pie

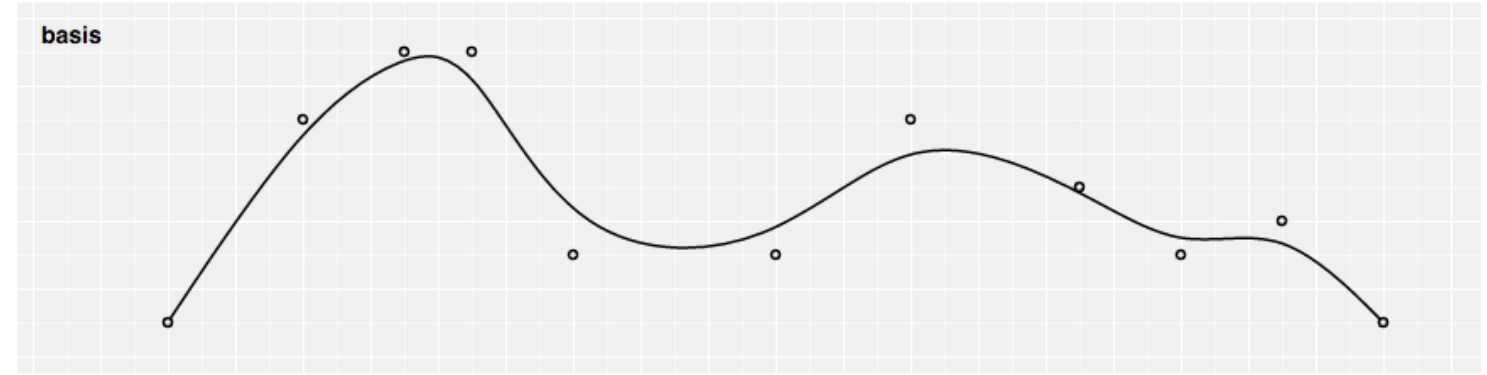
PIE CHARTS

Let's build a pie chart!

CURVES

CURVES

d3 has several curve generators:



d3-shape#curves

CURVES

You can plug them into the other line- and area-generators:

```
d3.line()
...
.curve(d3.curveBasis);
```

d3-shape#curves

AREA

AREA

Generate areas with *d3.area()*:

```
var area = d3.area()
  .x( ... )
  .y(250)  // baseline
  .y1( ... )
```

SHAPES + TRANSITIONS

SHAPES + TRANSITIONS

The *d* attribute of paths can also be animated via d3 transitions!

Have a look at code/d3-shapes/01_shape_transitions.html.

SHAPES + TRANSITIONS

Try building the flowers from the OECD Better-Life-Index based on code/d3-shapes/02_bli_flowers.html.

OECD Better-Life-Index

MORE

Mike Bostock: Introducing d3-shape

Andy Shora: Tweening Custom Shapes and Paths in d3.js

D3 LAYOUTS

D3 LAYOUTS

d3 has several predefined layouts to help with generating charts.

d3 Gallery

D3 LAYOUTS

All of your favorites should be there:

- cluster
- tree
- treemap
- partition
- pack
- chord
- forces

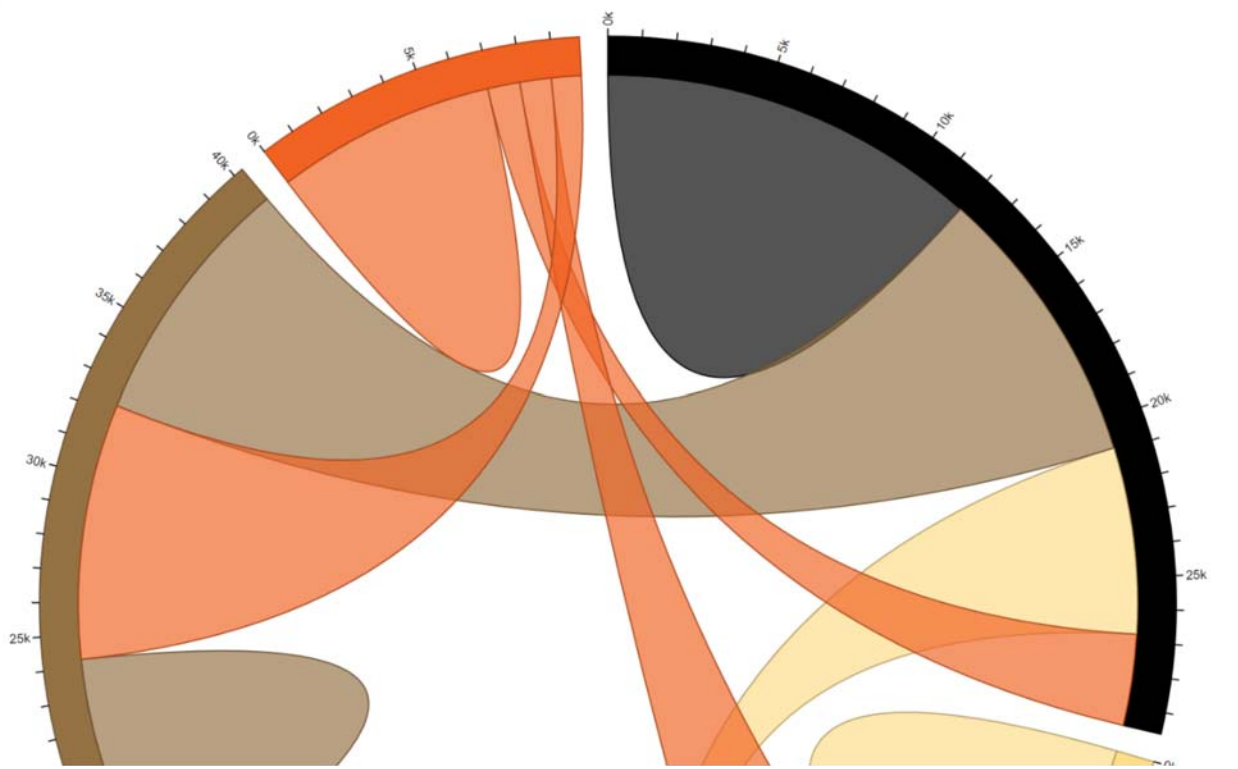
D3 LAYOUTS

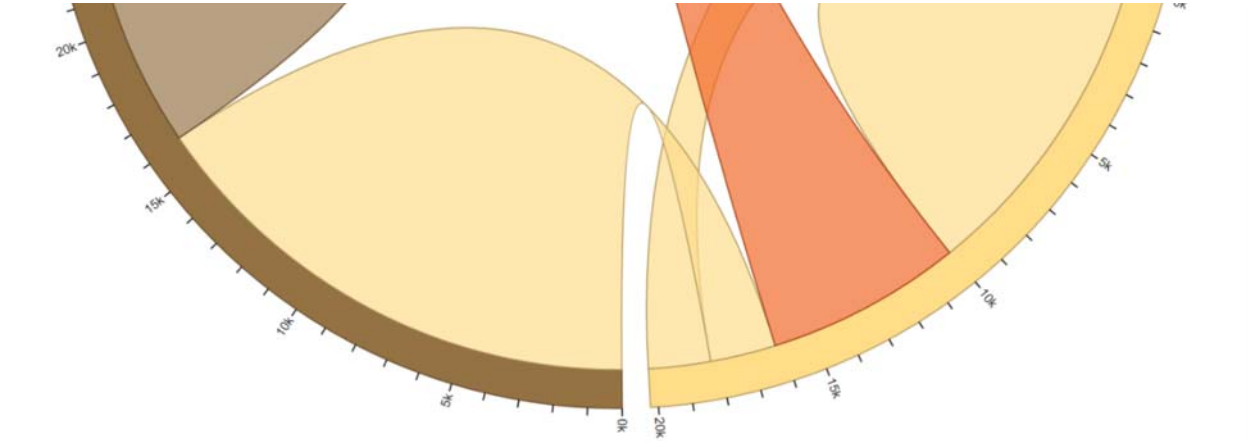
All layouts have in common that they're not drawing anything - they're only doing data manipulation to have an easier time drawing.

CHORD DIAGRAMS

CHORD DIAGRAMS

Chord layouts show object-to-object relationships.




20k
15k
10k
5k
0k
0k
5k
10k
15k
20k

CHORD DIAGRAMS

d3 provides *d3.chord* to calculate the outer ring segments.
d3.ribbon calculates the connecting ribbons between those segments.

CHORD DIAGRAMS

Let's play around with Mike Bostock's Chord Diagram example (in code/d3-layouts/01_bostock_chord.html)

Mike Bostock: Chord Diagram

HIERARCHIES

HIERARCHIES

Most layouts are based on some type of hierarchical data. More tree than list.

HIERARCHIES

Tabular data (such as CSVs) needs to be converted to a hierarchy (Object) format first, before the d3 hierarchical layouts can use it.

HIERARCHIES

You can use *d3.stratify* to do that:

```
var data = [{}, {}, {}, ...];

var root = d3.stratify()
  .parentId(function(d){ return d.parent; })
  (data);
```

HIERARCHIES

Try loading and stratifying the data in code/d3-layouts/02_hierarchical_data.html

HIERARCHIES

One special weirdness of *d3.stratify*: every parent node needs it own entry in the data!

```
id; region;
Netherlands; Europe;
Europe; ;     <-- required!
```

HIERARCHIES

Stratified data becomes a *d3.hierarchy* object:

- **.descendants()** - returns all children of a node as a flat array
- **.sum(value)** - calculates a node's value (see below)
- **.depth** - depth in the tree

d3-hierarchy#hierarchy

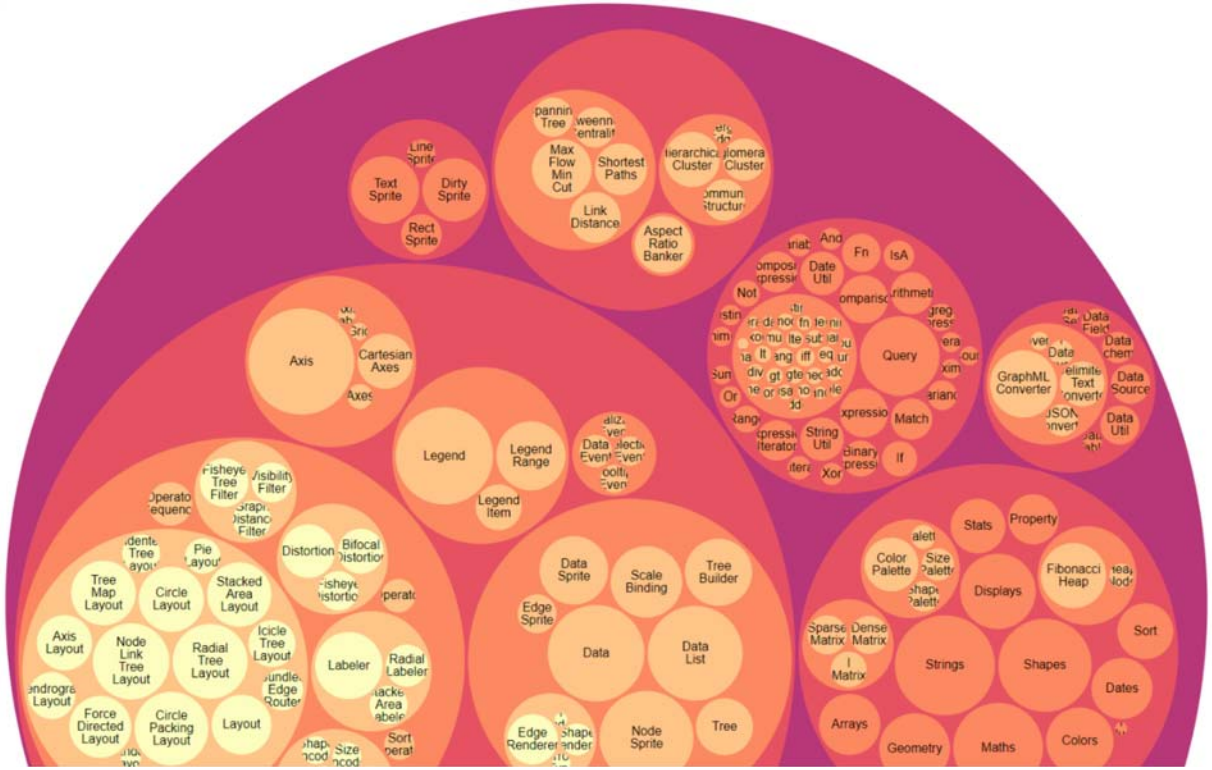
HIERARCHIES

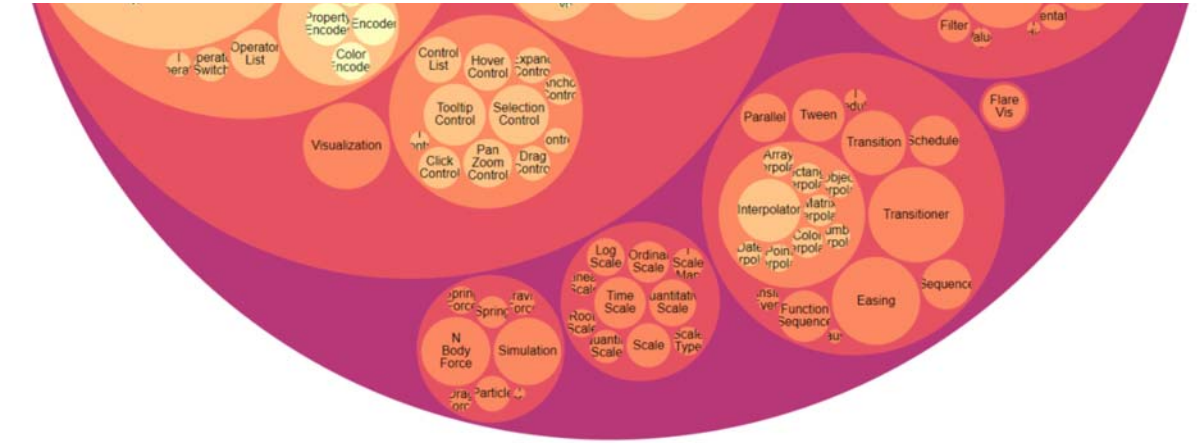
The good news: once the data has been stratified, drawing it is usually very straightforward!

CIRCLE PACKING

CIRCLE PACKING

To put that to the test: let's draw a circle packing viz based on the data!




Property Encoder
Encoder
Operator List
Color Encoder
Control List
Hover Control
Tooltip Control
Selection Control
Visualization
Click Control
Pan Zoom Control
Drag Control
Filter
Flare Vis
Parallel
Tween
Transition
Schedule
Interpolator
Transitioner
Easing
Sequence
Function Sequence
Log Scale
Ordinal Scale
Time Scale
Scale
N Body Force
Simulation

TREES & TREEMAPS

TREES & TREEMAPS

Let's have a look at a Mike Bostock example for drawing a tree based on *d3.tree*:

Mike Bostock: Tidy Tree

TREES & TREEMAPS

Change the world pack layout from before to a tree layout!

TREES & TREEMAPS

Finally, one last example: the treemap.

Mike Bostock: Treemap

MAPS

MAPS

One of the most common visualization tasks: putting stuff on a map!

PROJECTIONS

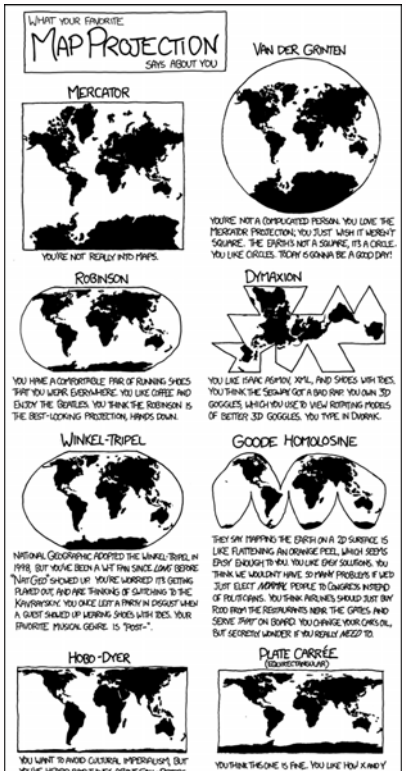
PROJECTIONS

The main challenge of that: the 3D-ness of the real world.

Projections are functions to map 3D coordinate to a 2D plane.

PROJECTIONS

And: there's no right or wrong one!
So we have a whole zoo of them!



XKCD: Map Projections


YOU'VE HEARD BAD THINGS ABOUT GALL-PETERS. YOU'RE CONFLICT-AVERSE AND BUY ORGANIC. YOU USE A RECENTLY-INVENTED SET OF GENDER-NEUTRAL PRONOUNS AND THINK SUPE WHAT THE WORLD NEEDS IS A REVOLUTION IN CONSCIOUSNESS.
A GLOBE!
YES, YOU'RE VERY CLEVER.
PEIRCE QUINCUNCIAL
YOU THINK SUPE WHEN WE LOOK AT A MAP, WHAT WE REALLY SEE IS OURSELVES. AFTER YOU FIRST SAW INCEPTION, YOU SAT SILENT IN THE THEMED BAR SIX HOURS. IT BREAKS YOU OUT TO REALIZE THAT EVERYONE AROUND YOU HAS A SKELETON INSIDE THEM. YOU HAVE REALLY LOOKED AT YOUR HANDS.
MAP TO LATITUDE AND LONGITUDE. THE OTHER PROJECTIONS OVERCOMPLICATE THINGS. YOU WANT ME TO STOP RAMBLING ABOUT MAPS SO YOU CAN ENJOY DINNER.
WATERMAN BUTTERFLY
REALLY? YOU KNOW THE WATERMAN? HAVE YOU SEEN THE PROF CHALL MAP ITS BASED— ...YOU HAVE A FRAMED REPRODUCTION AT HOME?! WHOA. ...LISTEN, FORGET THESE QUESTIONS. ARE YOU DOING ANYTHING TONIGHT?
GALL-PETERS
I HATE YOU.

PROJECTIONS

d3 has a whole spinoff library with tons of projections:

d3-geo-projection

PROJECTIONS

And it's relatively easy to use them - let's try it with code/maps/01_maps_d3.html.

PROJECTIONS

Many d3 map examples use TopoJSON - Mike Bostock's optimized JSON format for storing geo data.

```
Mike Bostock: How To Infer
Topology
```

MAPBOXGL

MAPBOXGL

d3 is great for projecting things - but we probably also want to show those nice map tiles like in Google Maps et al.

MAPBOXGL

Mapbox is a (commercial) way to show maps - but: very convenient to use, with a free plan.

Mapbox

MAPBOXGL

Mapbox provides several APIs - their fastest and latest is MapboxGL, based on WebGL.

MAPBOXGL

Let's build their Hello World example based on code/maps/mapbox-template.html

MapboxGL API

MapboxGL Examples

MAPBOXGL

Now: experiment with initial settings. Try to show satellite images, for example.

MapboxGL API

MapboxGL Examples

MAPBOXGL

While the basic example works with the default access token, make sure to use your own at some point: https://www.mapbox.com/studio/account/to

MAPBOX STUDIO

MAPBOX STUDIO

Mapbox comes with a cool web-based style editor: Mapbox Studio.

https://www.mapbox.com/studio/styles/

MAPBOX STUDIO

Use it to create your own style, then replace the default style from the example with it.

HEATMAPS

HEATMAPS

Creating heatmaps with MapboxGL is pretty easy - let's look at https://www.mapbox.com/mapbox-gl-js/example/heatmap-layer/

HEATMAPS

Choropleths are similarly simple - here's an example: https://www.mapbox.com/mapbox-gl-js/example/updating-choropleth/

SCROLLYTELLING

Nice example of scrollytelling: https://www.mapbox.com/mapbox-gl-js/example/scroll-fly-to/

GEOJSON

GEOJSON

The most common JSON format for geographic data is GeoJSON. Not as efficient as TopoJSON, but more widespread.

GEOJSON

GeoJSON can be directly plugged into Mapbox:

```
map.addLayer({
  'type': 'fill',
  'source': {
    'type': 'geojson',
    'data': GEOJSONDATA
  },
});
```

GEOJSON

Check out code/maps/03_mapbox_geojson.html for an example.

INTERACTION

INTERACTION

Mapbox comes with event handlers - very similar to d3 - and lots of infos in the event object:

```
map.on('click', function(e){
  console.log(e.point); // current on-screen positi
  console.log(e.lngLat); // longitude, latitude of
});
```

INTERACTION

Other useful event handlers:

```
// map has loaded:
map.on('load', function(){...});

// map starts moving:
map.on('movestart', function(){...});
// map stops moving:
map.on('moveend', function(){...});
```

PROJECTIONS

PROJECTIONS

MapboxGL *map* objects also support projections out of the box:

```
// convert lat/lng to screen coords:
map.project([-122.420679, 37.772537]);

// convert screen coords to lat/lng:
map.unproject([540, 288]);
```

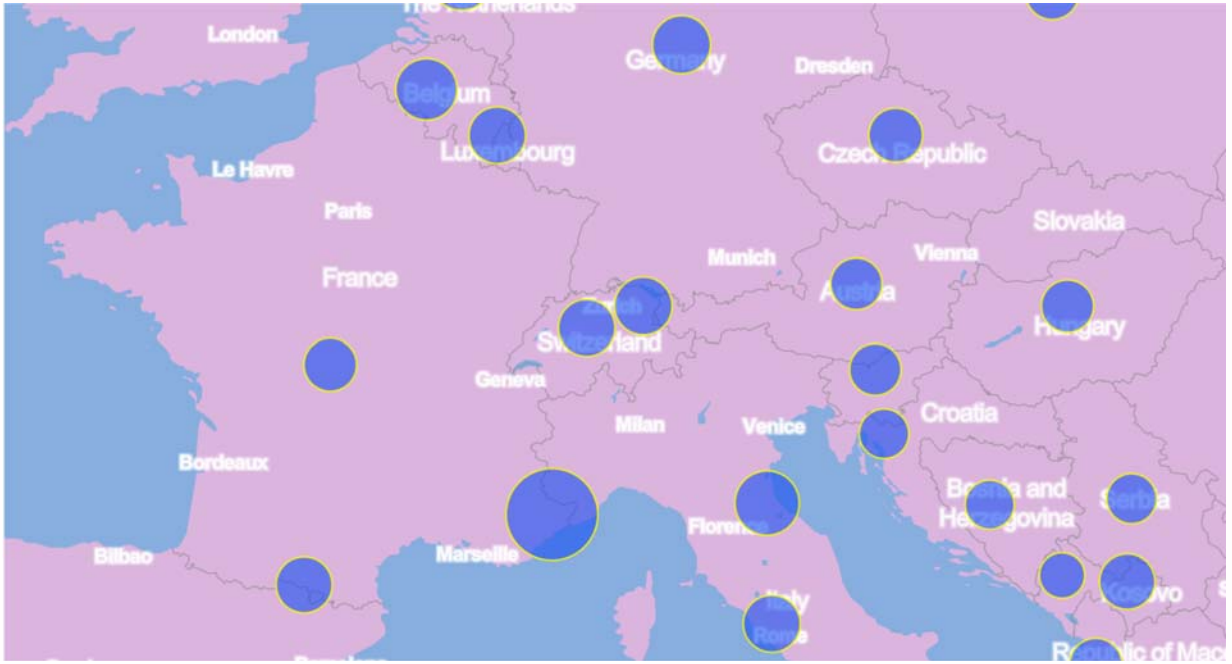
MAPBOX + D3

MAPBOX + D3

Use an SVG/Canvas overlay on top of the Mapbox element to display data using d3.

MAPBOX + D3

Start with code/maps/04_mapbox_d3.
Visualize population density as
circles on top of the map.




Spain
Valencia
Naples
Albania
Palermo
Catania
Algiers
Constantine
Tunis
Sousse
Oran
Fez
Tripoli
Misrata
Benghazi
Ghardaia
mapbox
© Mapbox © OpenStreetMap Improve this map

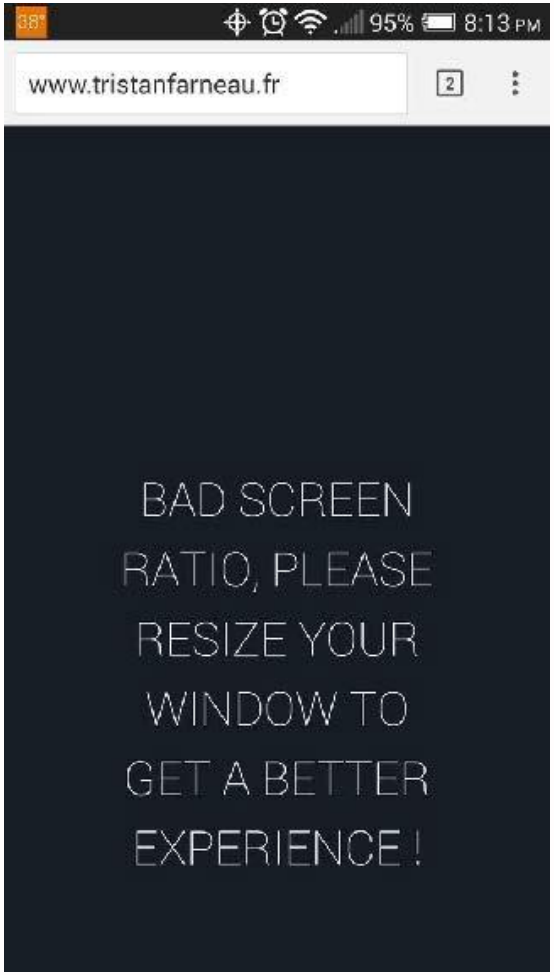
RESPONSIVE + PERFORMANCE


What a mobile website is supposed to do
Make this
Logo
Navigation
Sidebar
Content
readable on this
Navigation
Content
What every major news outlet is doing:
Turning this
Logo
Navigation
Sidebar
Content
The Oatmeal
Into this
HOLY SHITBUTTS!
YOU'VE BEEN SELECTED!
FREE IPHONE 8
IT WILL SLIM YOUR BELLYFAT!
CLICK THIS FUCKING AD
BECAUSE FUCK YOU, YOU'RE STUPID.
10 PHOTOS OF KATY PERRY DIGESTING ZOO ANIMALS
#7 will shock you!

The Oatmeal

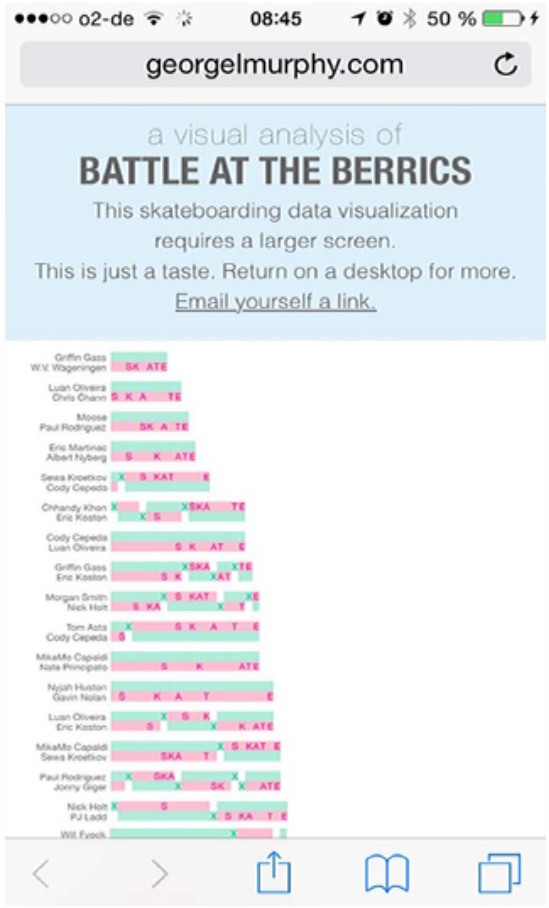
RESPONSIVE WEB DESIGN

RESPONSIVE WEB DESIGN



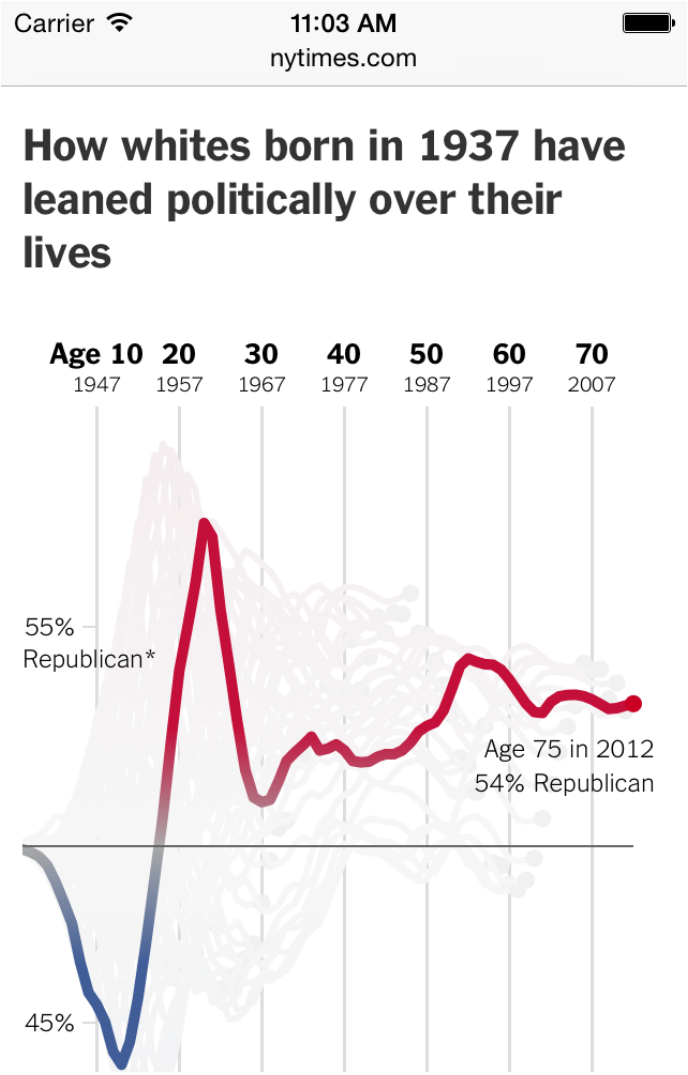
RESPONSIVE WEB DESIGN

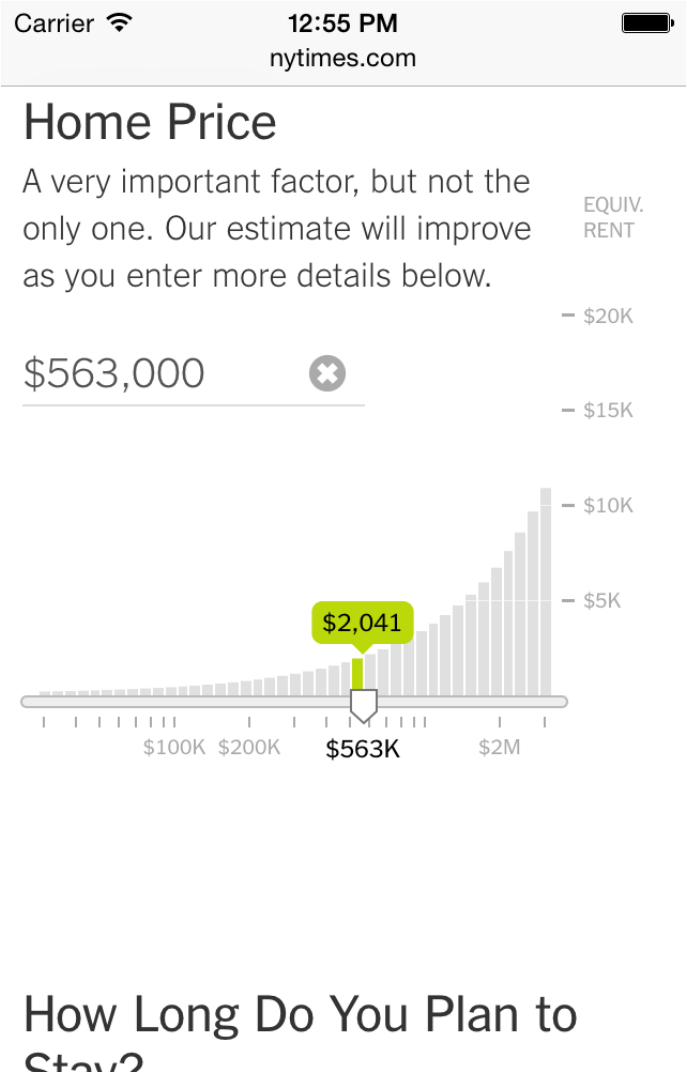
Email Fallback:


o2-de
08:45
50 %
georgelmurphy.com
a visual analysis of
BATTLE AT THE BERRICS
This skateboarding data visualization
requires a larger screen.
This is just a taste. Return on a desktop for more.
Email yourself a link.

RESPONSIVE VISUALIZATIONS

RESPONSIVE VISUALIZATIONS





If you can rent a similar home for less than **$2,041** per month, then renting is better.

RESPONSIVE VISUALIZATIONS

Bocoup: Mobile Vis

Sebastian Sadowski: Mobile Infovis

RESPONSIVE WEB DESIGN

Major form factors/devices:

- Desktops/Laptops
- Tablets
- Phones

RESPONSIVE WEB DESIGN

Differences:

- Screen size
- Input modality
- Sensors

RESPONSIVE WEB DESIGN

Differences:

- Environment
- Mindset
- Ease of use
- Context

RESPONSIVE WEB DESIGN

Responsive → Adaptive Web Design

Device-centric → User-centric

HTML/CSS TECHNIQUES

HTML/CSS TECHNIQUES

Use the right meta tags:

```
<meta
  name="viewport"
  content="width=device-width, initial-scale=1"
>
```

HTML/CSS TECHNIQUES

Use relative sizes instead of absolute ones:

```
#viz {
  width: 960px;
}
```

```
#viz {
  width: 80%;
}
```

HTML/CSS TECHNIQUES

Use (r)em instead of px/pt for font sizes:

```
.title {
  font-size: 2em;
}

body {
  font-size: 24px;
}
```

HTML/CSS TECHNIQUES

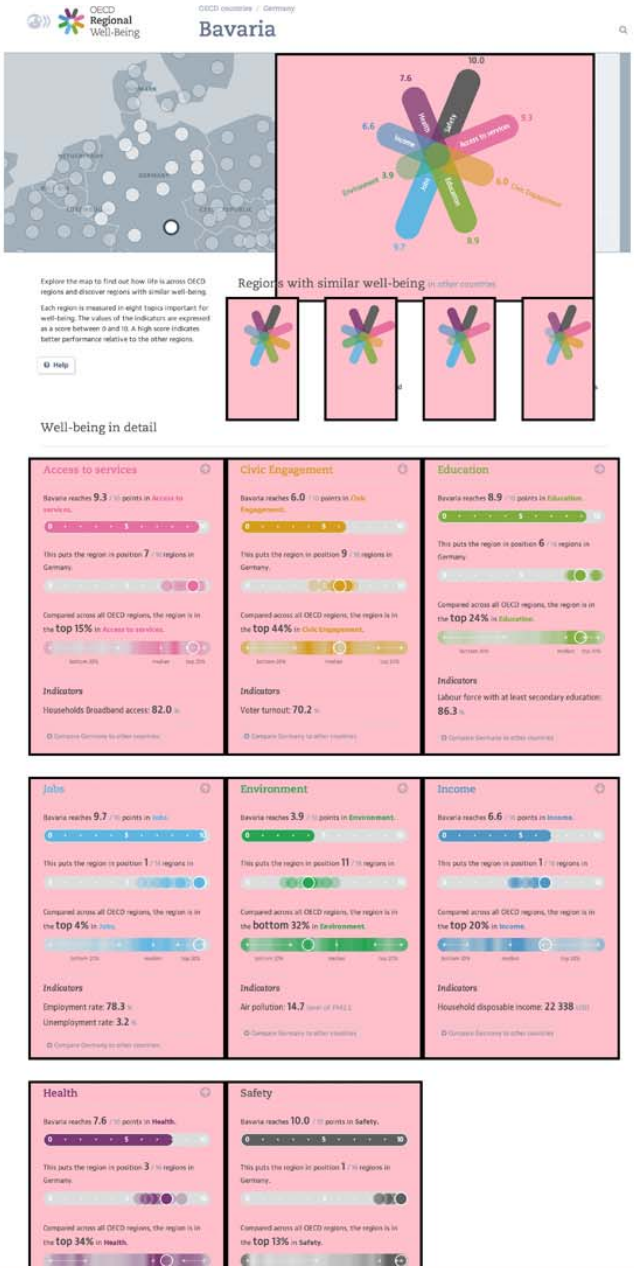
CSS media queries let you fine-tune for specific devices:

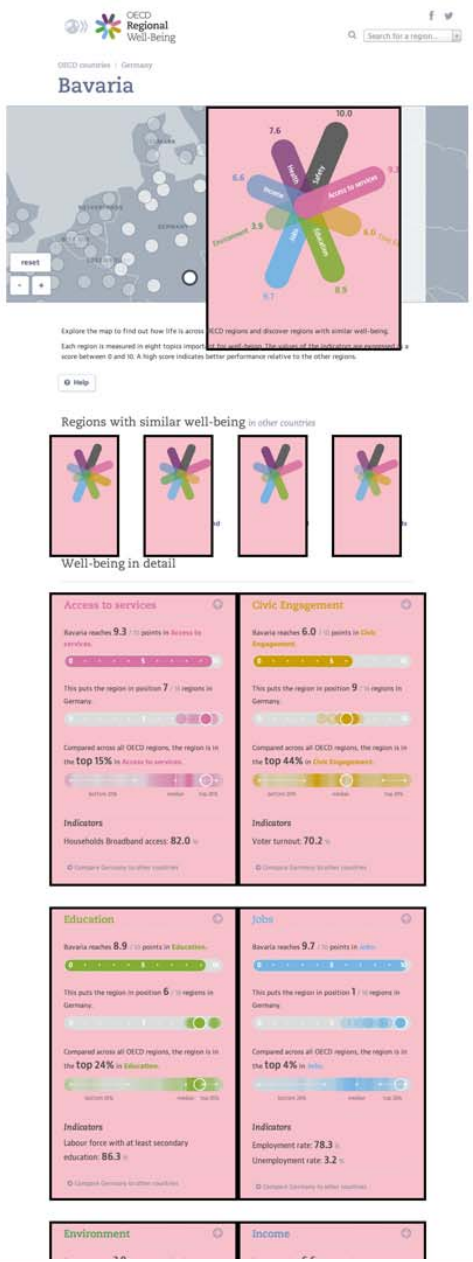
```
#nav {
  width: 100%;
}

@media screen and (min-width: 1368px) {
  #nav {
    float: left;
    width: 32.5%;
  }
}
```

GRIDS

GRIDS

Organizing content in boxes and placing them in a grid not only makes for a cleaner design, but also helps with adapting to different screen sizes.





OECD Regional Well-Being

GRIDS

There are various CSS frameworks supporting grids out there. The bigger ones are Twitter's Bootstrap and Zurb's Foundation.

GRIDS

Let's have a look at Bootstrap's grids:

Bootstrap grid examples

SVG TECHNIQUES

SVG TECHNIQUES

As the name implies, SVG are scalable, which makes them automatically look nice on various types of devices.

SVG TECHNIQUES

You can fine-tune the scaling using the SVG viewBox and preserveAspectRatio attributes:

```
<svg width="600" height="400"
  viewBox="0 0 200 200"
  preserveAspectRatio="xMinYMax meet" />
```

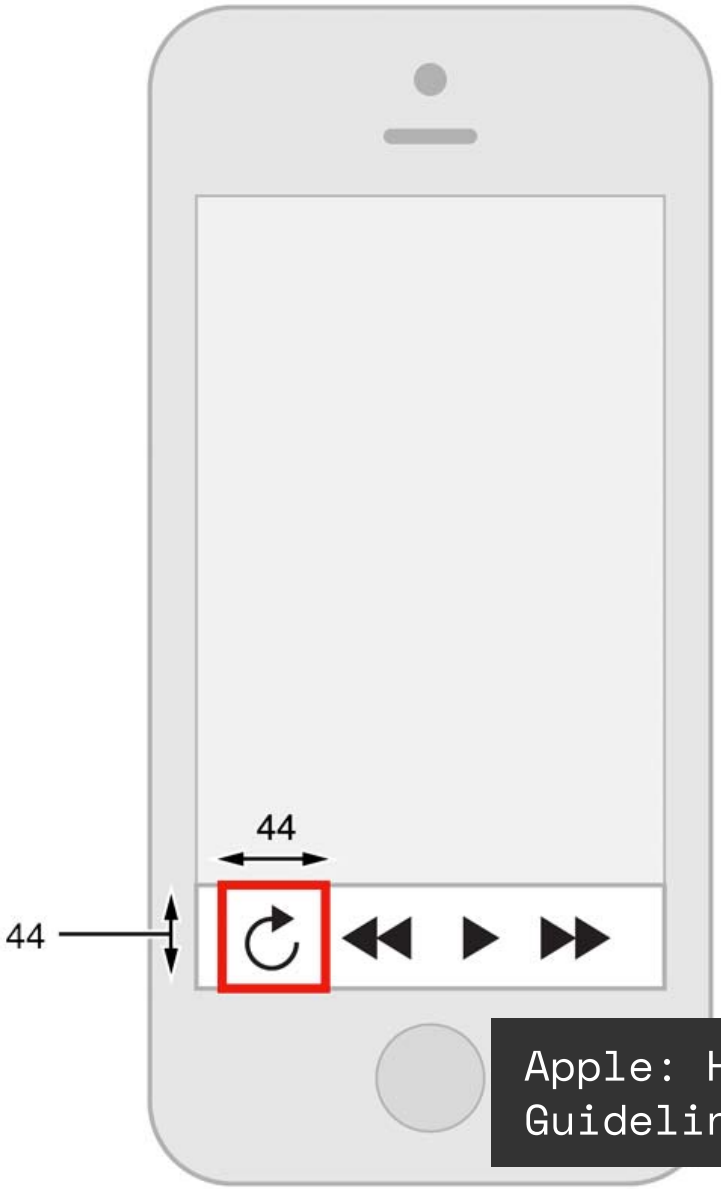
Let's look at code/responsive/01_svg_viewbox.html

INTERACTION

INTERACTION

Touch input has different event handlers. But they work just like their mouse counterparts.

```
circle.on('touchstart', ...)
  .on('touchmove', ...)
  .on('touchend', ...);
```



Apple: Human Interface Guidelines

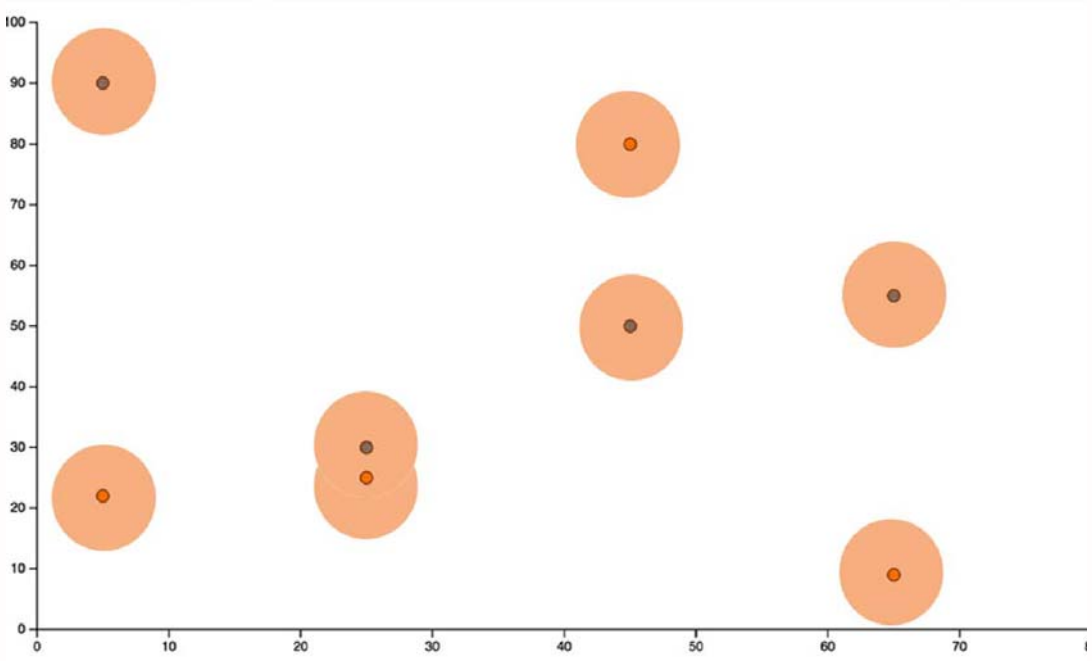
INTERACTION

Mobile browsers have to juggle touch-to-scroll with touch-to-interact. If you have a fullscreen page anyway, consider disabling scrolling:

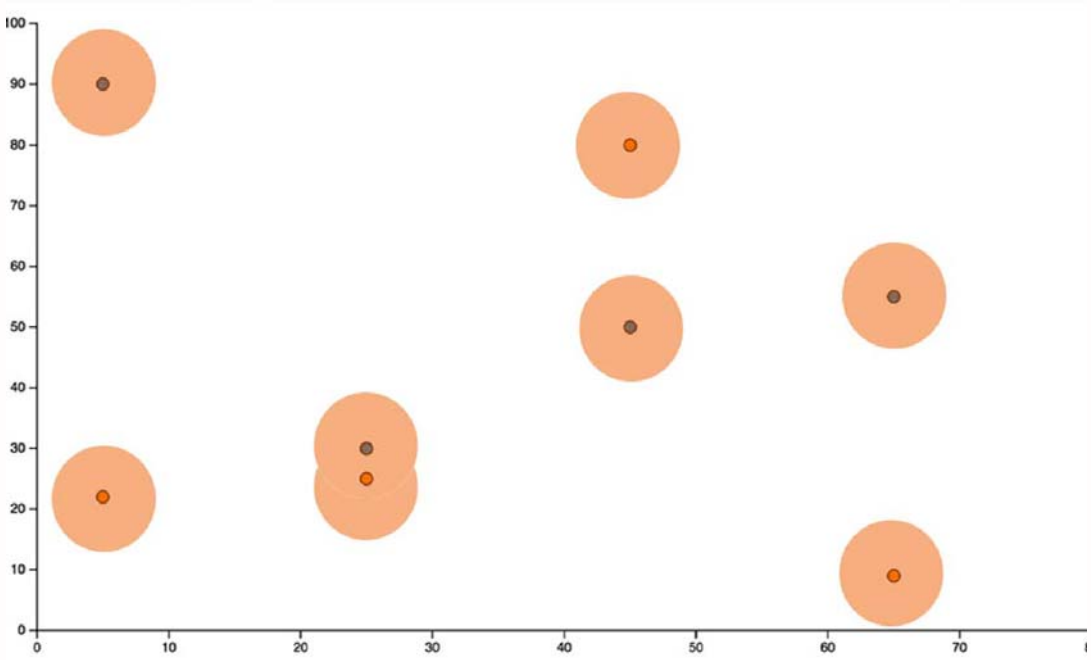
```
document.ontouchstart =
  function(e){
    e.preventDefault();
  };
```

INTERACTION

Consider adding enlarged (invisible) touch areas:



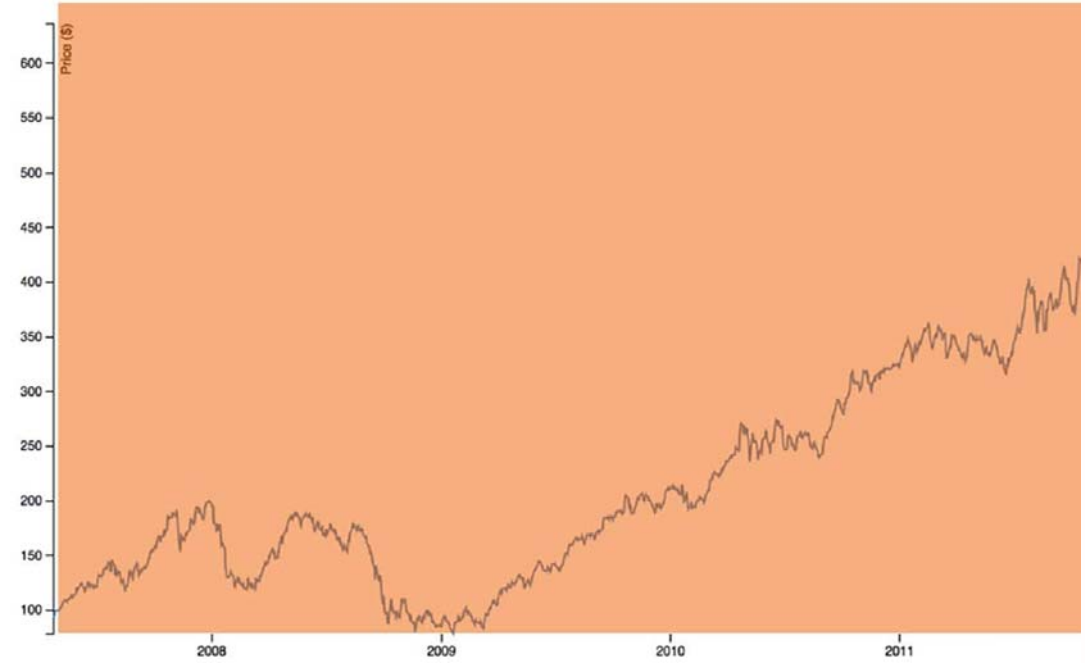
INTERACTION




```
.touch-help {
    opacity: 0;
    z-index: 99999;
}
```

INTERACTION

... or even take over interactivity completely:



INTERACTION


```
touchHelpers.on('touchstart', function(){
  console.log(d3.touches(this));
  // => [342.1, 283.5]
```

```
});
```

TOOLTIPS

TOOLTIPS

Tooltips are pretty useful for data exploration - we don't want to lose them on mobile!

TOOLTIPS

One solution: have a *position: fixed* tooltip at the bottom!

TOOLTIPS

Two-step interaction: tap once to open, tap again to click, tap anywhere to close.


```
OECD Regional Well-
Being
```

SENSORS

SENSORS

Mobile devices also have upsides:
cool sensors!

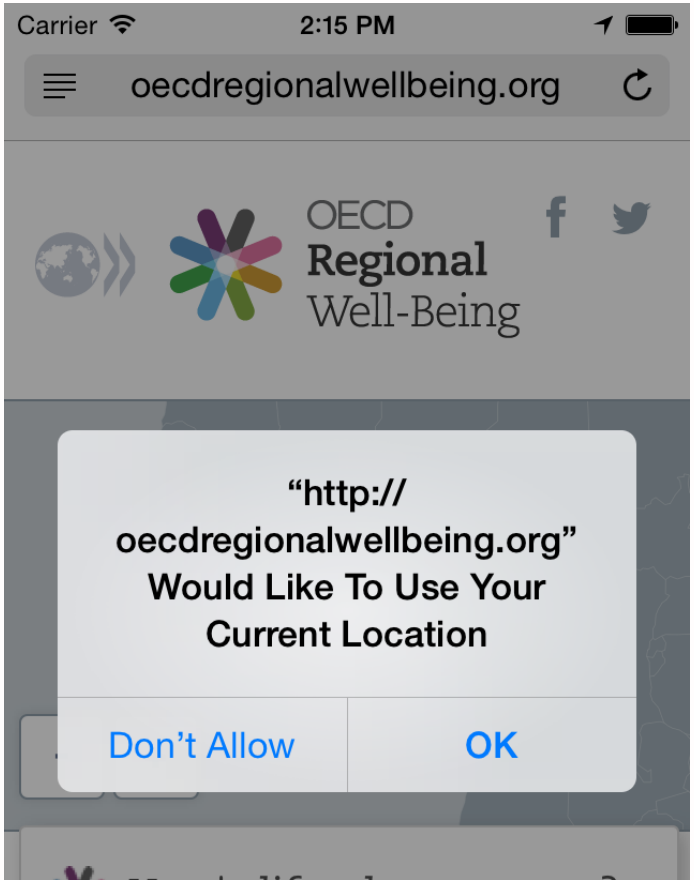
SENSORS

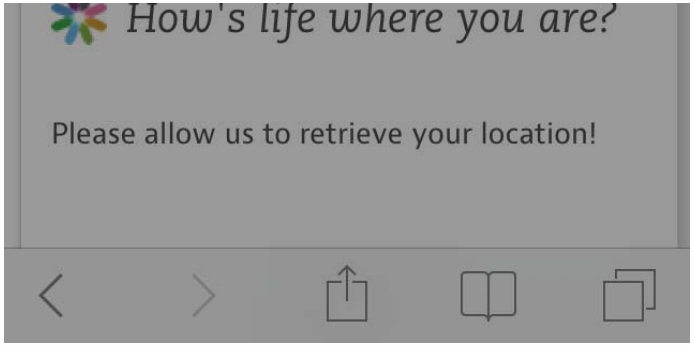
Geolocation:

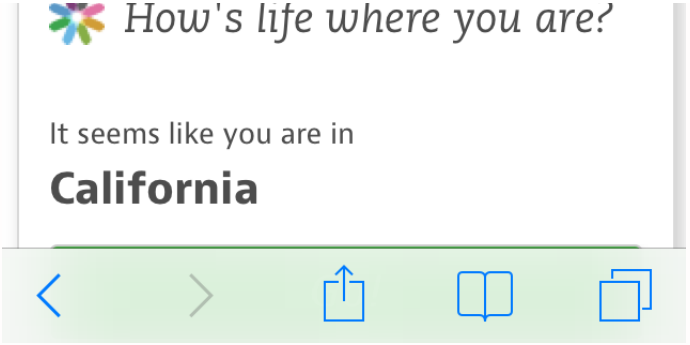
```
navigator.geolocation.getCurrentPosition(
      function(position){
         console.log(
            position.coords.latitude +
            "," + position.coords.longitude
         );
      }
   );
```

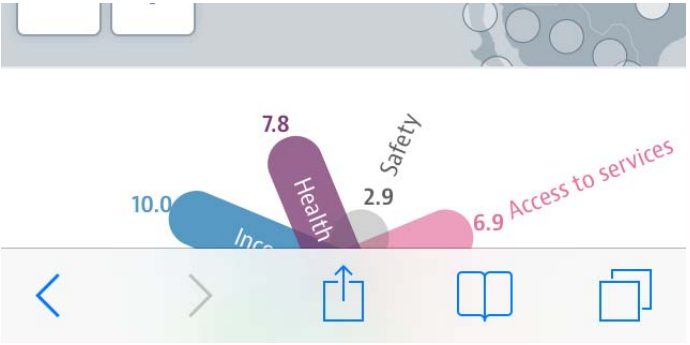
SENSORS

Geolocation:




How's life where you are?
Please allow us to retrieve your location!


How's life where you are?
It seems like you are in
California


7.8
Safety
Health
2.9
10.0
6.9
Access to services

SENSORS

Multitouch/Gestural interactions: Supported by most d3 behaviors out of the box.

Mike Bostock: Pan & Zoom IV

Mike Bostock: Multitouch II

SENSORS

You can also use a dedicated gesture library like QuoJS or HammerJS to support fancier interactions.

SENSORS

Gyroscopes/Accelerometers

```
window.addEventListener(
    "deviceorientation",
    function(orientation){
        console.log(orientation.alpha +
        "," + orientation.beta +
        "," + orientation.gamma
        );
    },
    true
 );
```

SENSORS

Check out code/responsive/02_bluemarble.html for a demo.

MORE

Bill Hinderman: Building Responsive Data Visualization for the Web

PERFORMANCE

PERFORMANCE

Speaking about responsiveness, oftentimes we bind redraw handler to the resize event:

```
window.on('resize', redraw);
```

PERFORMANCE

This can lead to A LOT of redraws. Consider using some way of debouncing:

```
window.on('resize', _.debounce(redraw, 300))
```

PERFORMANCE

Use *requestAnimationFrame* instead of *setTimeout*.

PERFORMANCE

Sometimes: canvas/WebGL can be faster than SVG (usually with really massive amounts of shapes/animations).

PERFORMANCE

Plus all the other details:

- put scripts, styles last
- minify your code/images
- ...

FORCES + VORONOI

NETWORKS

NETWORKS

You might have noticed that we've ignored one type of data completely so far: networks.

NETWORKS

Networks have their problems:

- visualizations are harder to read
- do not scale well
- harder to represent (tabular?)

NETWORKS

Network data consists of **nodes** that have **attributes** and are connected via **links** (which can also have attributes)

NETWORKS

It can be represented as a Javascript object thusly:

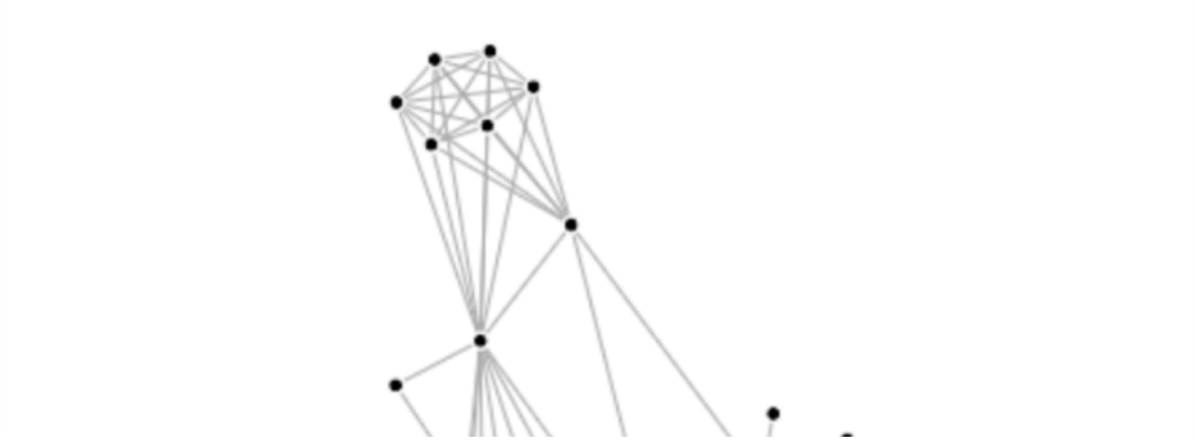
```
{
  "nodes": [{"id": "node1"}, ...],
  "links": [{"source": "node1", "target": "node2"}]
}
```

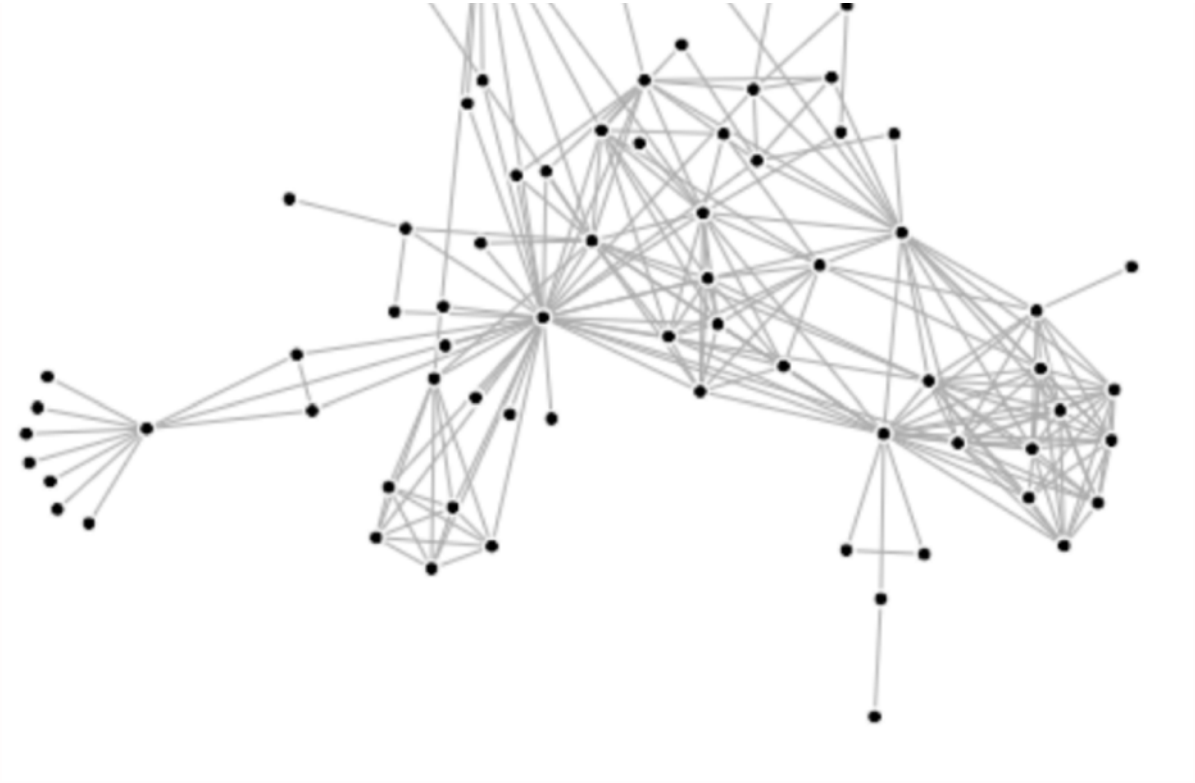
Les Miserables network

NODE-LINK DIAGRAMS

NODE-LINK DIAGRAMS

A common way to visualize networks are node-link diagrams. Each node becomes a dot, links become lines between dots. Easily turns into a hairball.





NODE-LINK DIAGRAMS

Major challenge of node-link diagrams: how to do the layout.
One common approach is the **force-directed layout**, a physics simulation where nodes repel each other, while being attracted trough their links.

NODE-LINK DIAGRAMS

d3 comes with a built-in force simulation in *d3.force*.

`d3-force`

NODE-LINK DIAGRAMS

To work with d3-force, use an array of nodes. The simulation adds the following attributes:

```
var nodes = [{
x: 0, // x-position
y: 0, // y-position
vx: 0, // x-velocity
vy: 0 // y-velocity
}]
```

NODE-LINK DIAGRAMS

You can also add *fx* and *fy* attributes to fix the node at position (fx, fy).

NODE-LINK DIAGRAMS

d3-force is flexible enough to support all kinds of physics simulations. You can add specific **forces** with *.force()*. d3 comes with several predefined forces such as centering, collision and links.

d3-force#forces

NODE-LINK DIAGRAMS

- **d3.forceLink** - forces working along links
- **d3.forceManyBody** - forces among all nodes (gravity)
- **d3.forceCenter** - drags nodes to the center
- **d3.forceCollision** - simulates collisions

NODE-LINK DIAGRAMS

Use the *tick* event of forceSimulation to redraw!

```
var redraw = function(){ ... }
var sim = d3.forceSimulation()
  .on('tick', redraw);
```

NODE-LINK DIAGRAMS

Let's build a node-link diagram based on the Les Miserables dataset (code/d3-forces/01_miserable.html)

LIFECYCLE OF SIMS

LIFECYCLE OF SIMS

Force simulations use an internal value called **alpha**. Every tick the value of alpha is reduced (set via *alphaDecay*). Once it's lower than *alphaMin*, the simulation stops.

LIFECYCLE OF SIMS

Use *.alpha()* and *.restart()* to restart the simulation:

```
sim.alpha(1);
sim.restart();
```

COLLISIONS

COLLISIONS

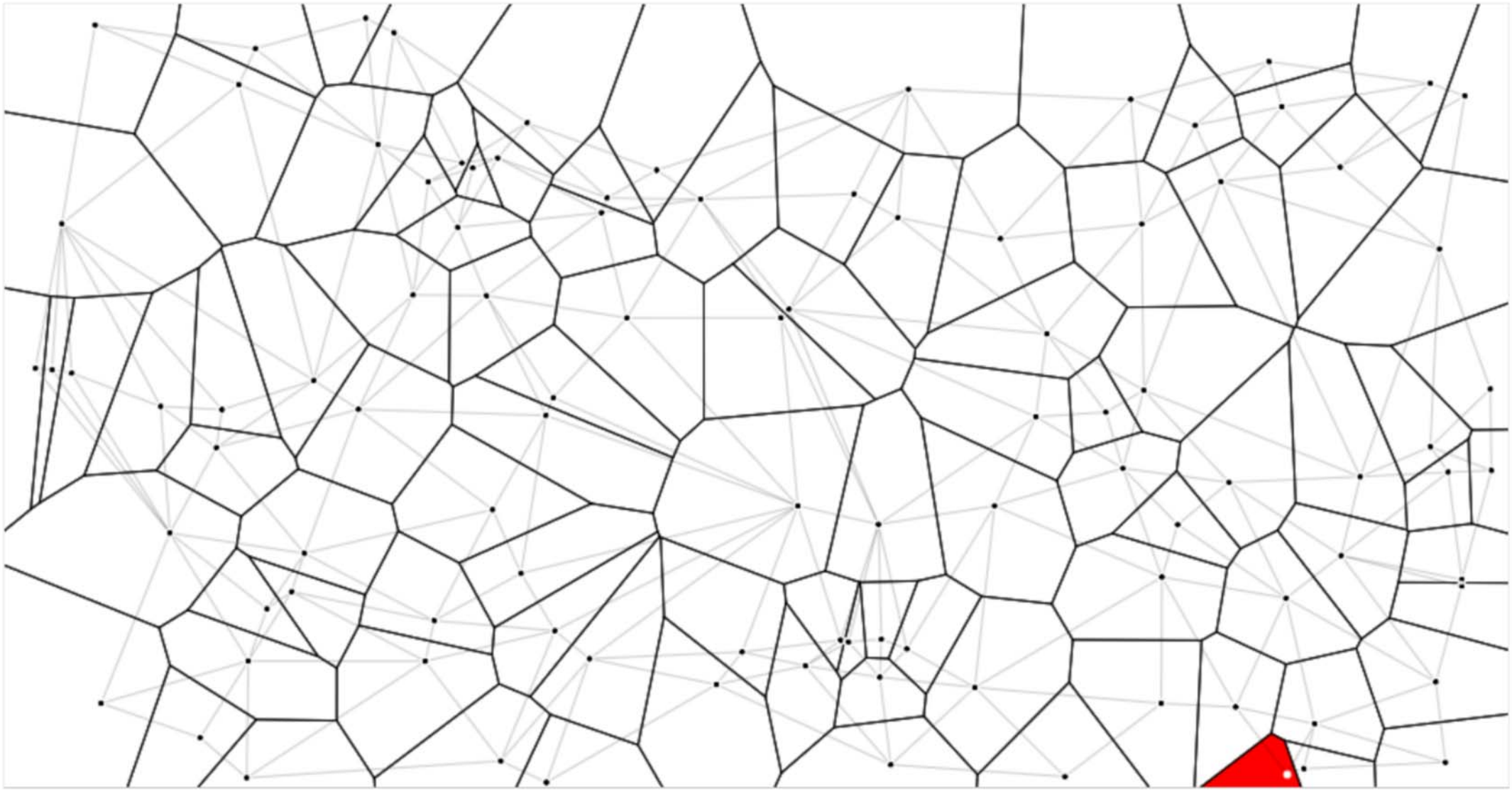
You can also use d3-force to simulate collisions. Have a look at d3-forceCollide, then create a bunch of random circles and experiment with collisions!

VORONOI

VORONOI

Voronoi diagrams describe a partitioning of a plane based on points so that everything in a Voronoi "cell" is closest to its seed point.

Mike Bostock: Canvas Voronoi



VORONOI

So what? Voronoi partitionings are great to improve interaction (e.g., for scatter plots). Make the Voronoi cells interactive instead of the (super small) points.

VORONOI

d3-voronoi calculates Voronoi cells based on a set of seed points:

```
var part = d3.voronoi()
  .extent([[0,0], [w,h]])
  (points);
```

VORONOI

Set accessor functions for x and y using *.x()* and *.y()*.

```
var part = d3.voronoi()
      .x(function(d){ return xScale(d.date); })
      .y(function(d){ return yScale(d.value); })
      (points);
```

VORONOI

d3-voronoi returns the result of the voronoization as:

- *.polygons()* - an array of polygons
- *.triangles()* - an array of (Delauney) triangles
- *.links()* - an array of triangle links

VORONOI

Build a scatterplot based on code/d3-forces/03_voronoi_scatter.html. Then, add highlighting based on a Voronoi diagram.

MORE

- Jim Vallandingham: Using and Abusing the Force

THANK YOU

DR. DOMINIKUS BAUR
HTTPS://DO.MINIK.US

GRAPHICHUNTERS.NL