CSC 444:
Midterm Review

Carlos Scheidegger

Midterm Exam
University of Arizona, Department of Computer Science
CSC 444, Data Visualization, Fall 2018

October 28th, 2018 9:30-10:45AM

This is a closed-book exam. There are a total of 8 problems in the exam (each worth the same amount of credit), and you have 75 minutes to finish it. Make sure your copy of the exam includes all pages, and contact the proctor in case it does not.

It should take no more than 50 words to answer each problem. If you need more space, use the back of the page. Answer each question briefly and precisely, and justify your answers using the principles and concepts we discussed in class. In case no solution satisfies all constraints in the problem, describe those constraints, and present possible trade-offs.

Name:

Signature:

Problems	
1	:
2	:
3	:
4	:
5	:
6	:
7	:
8	:

Final grade:

D3: DATA-DRIVEN DOCUMENTS

The essential idea

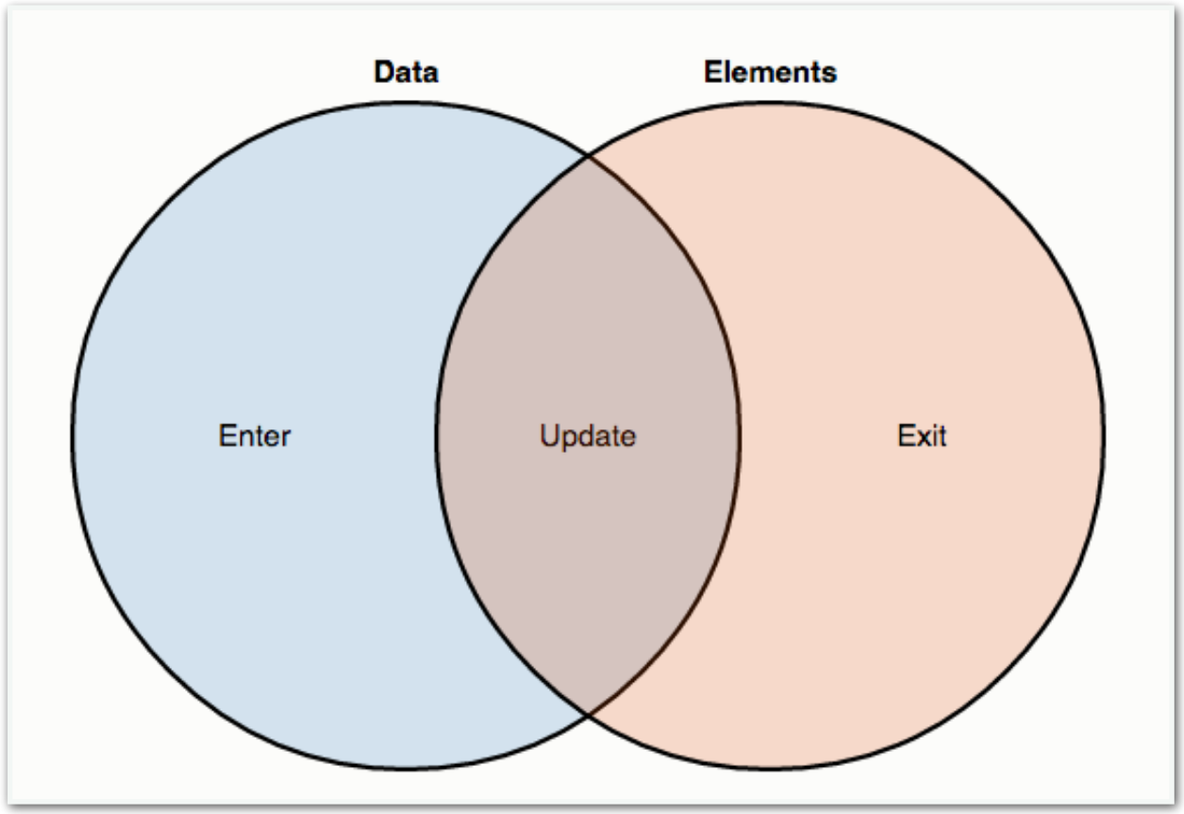
- D3 creates a two-way association between **elements of your dataset** and **entries in the DOM**
- D3 operates on **selections**
  - methods apply to all elements in the selection

Data Joins

- d3 associates data to a selection with the `data` method

```
d3.select(“svg”)
  .selectAll(“circle”)
  .data(inputData)
  .enter()
  .append(“circle”)
  .attr(“r”, function(d) {
   return d.age;
  });
```

Join Selections




```
d3.select(“svg”)
  .selectAll(“circle”)
  .data(inputData)
  .enter()
  .append(“circle”)
  .attr(“r”, function(d) {
   return d.age;
  });
```

http://bost.ocks.org/mike/join/

Selection methods

- `selection.method(accessor)`
- `selection`: which elements to change
- `method`: what to change about elements
- `accessor`: which aspect of the data

```
d3.select(“svg”)
 .selectAll(“circle”)
 .data(inputData)
 .enter()
 .append(“circle”)
 .attr(“r”, function(d) {
  return d.age;
 });
```

Selection methods

- `selection.method(accessor)`
- `selection`: which elements to change
- `method`: what to change about elements
- `accessor`: which aspect of the data

```
d3.select(“svg”)
 .selectAll(“circle”)
 .data(inputData)
 .enter()
 .append(“circle”)
 .attr(“r”, function(d) {
  return d.age;
 });
```

Selection methods

- `selection.method(accessor)`
- `selection`: which elements to change
- `method`: what to change about elements
- `accessor`: which aspect of the data

```
d3.select(“svg”)
 .selectAll(“circle”)
 .data(inputData)
 .enter()
 .append(“circle”)
 .attr(“r”, function(d) {
  return d.age;
});
```

- Write a d3 statement to **select all circles** in this DOM

```
<svg id="svg">
  <g>
    <circle cx=300 cy=400 r=30 fill=red/>
    <circle cx=200 cy=30  r=50 fill=blue/>
    <circle cx=40  cy=20  r=60 fill=black/>
  </g>
</svg>
```

d3.select("#svg").selectAll("circle")

- Write a d3 statement to set the **radius** of **all red circles** to **40**

```
<svg id="svg">
  <g id="group1">
    <circle cx=300 cy=400 r=30 fill=blue/>
    <circle cx=200 cy=30  r=50 fill=blue/>
    <circle cx=40  cy=20  r=60 fill=blue/>
  </g>
  <g id="group2">
    <circle cx=300 cy=400 r=30 fill=red/>
    <circle cx=200 cy=30  r=50 fill=red/>
    <circle cx=40  cy=20  r=60 fill=red/>
  </g>
</svg>
```

- You have data stored in an array:

```
var data = [ { age: 5,  height: 3 },
             { age: 12, height: 30 },
             { age: 15, height: 40 } ];
```

- Create a list of rectangles inside the svg element, each bound to an element of data

```
<svg id="svg">
</svg>
```

- You have data stored in an array:

```
var data = [ { age: 5,  height: 3 },
             { age: 12, height: 30 },
             { age: 15, height: 40 } ];
```

- The variable sel currently holds a selection of three rectangles, each bound to an element of data. Write a d3 statement that **sets to red the fill color** of all rectangles bound to values with **age greater than 10**.

d3 scales

- scales encode transformations between different spaces
- `var scale = d3.scaleLinear();`
- `scale.domain([d1, d2])`: where the transformation comes from
    - `scale.domain()` returns the current domain of the scale
- `scale.range([t1, t2])`: where the transformation goes to
    - `scale.range()` returns the current range of the scale
- `scale(x)`: send x through transformation
    - `scale.invert(y)` returns an x such that scale(x) == y

d3 scales

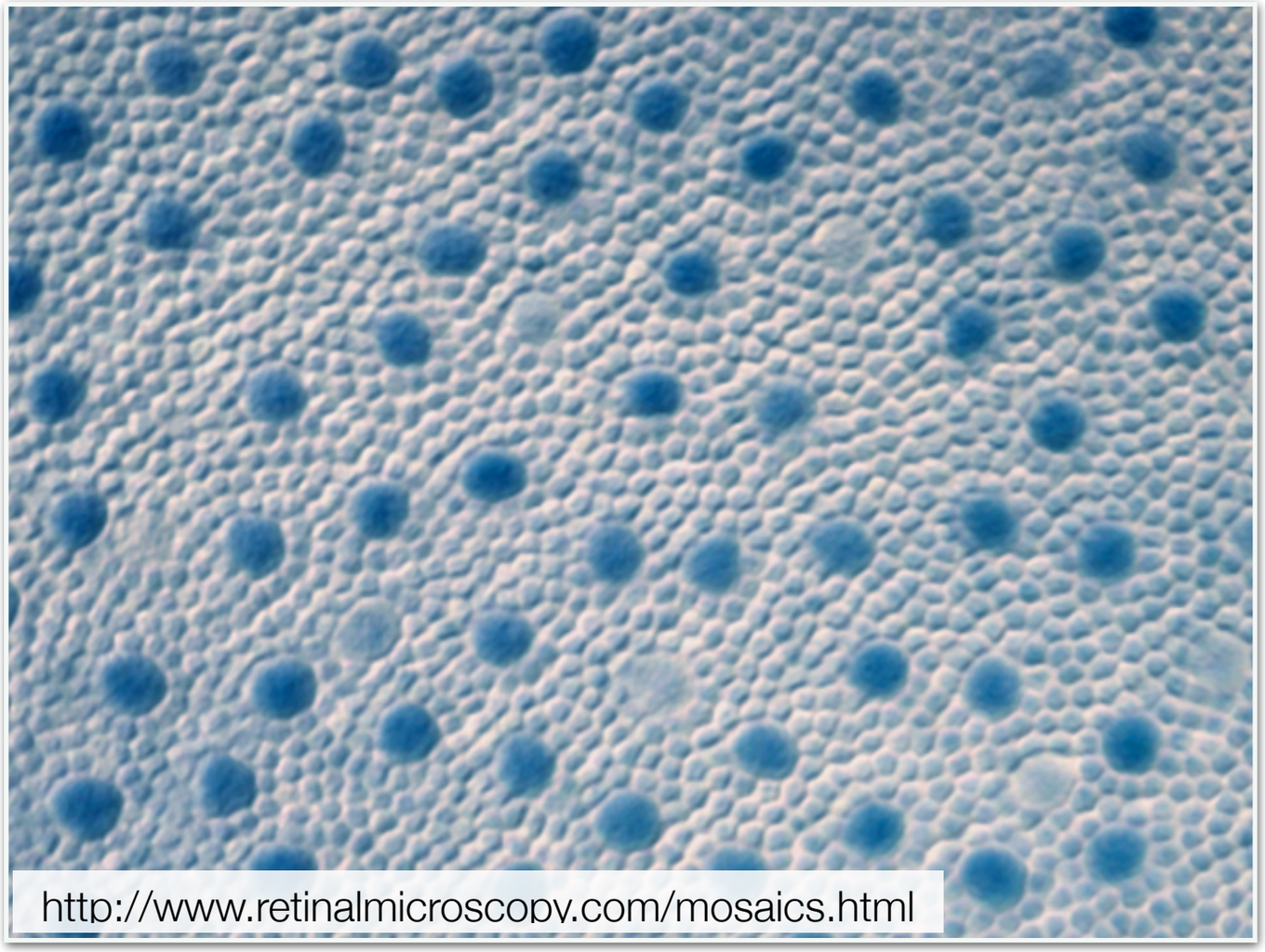
```
var scale = d3.scaleLinear()
.domain([10, 30]).range([100, 200]);
```

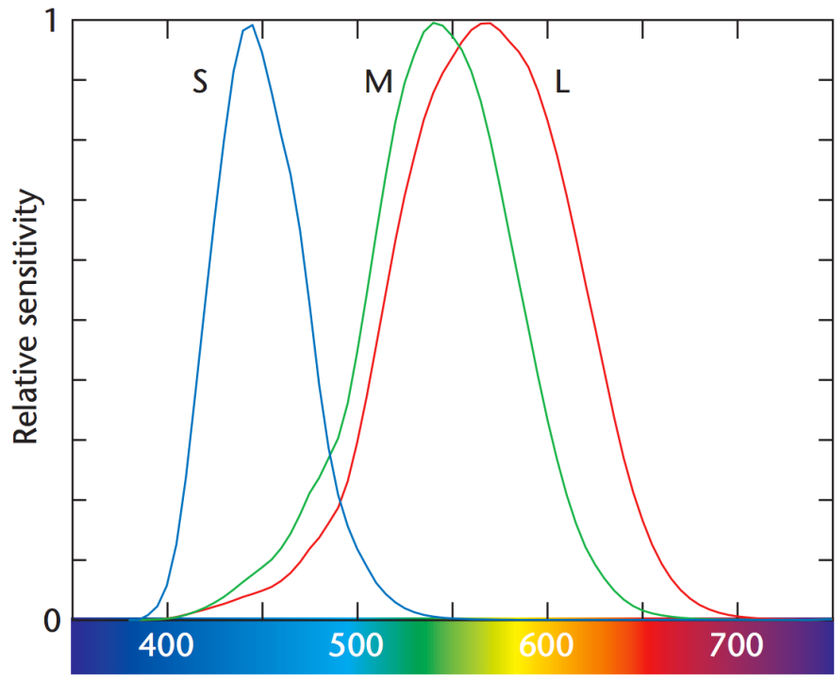
What's the result of

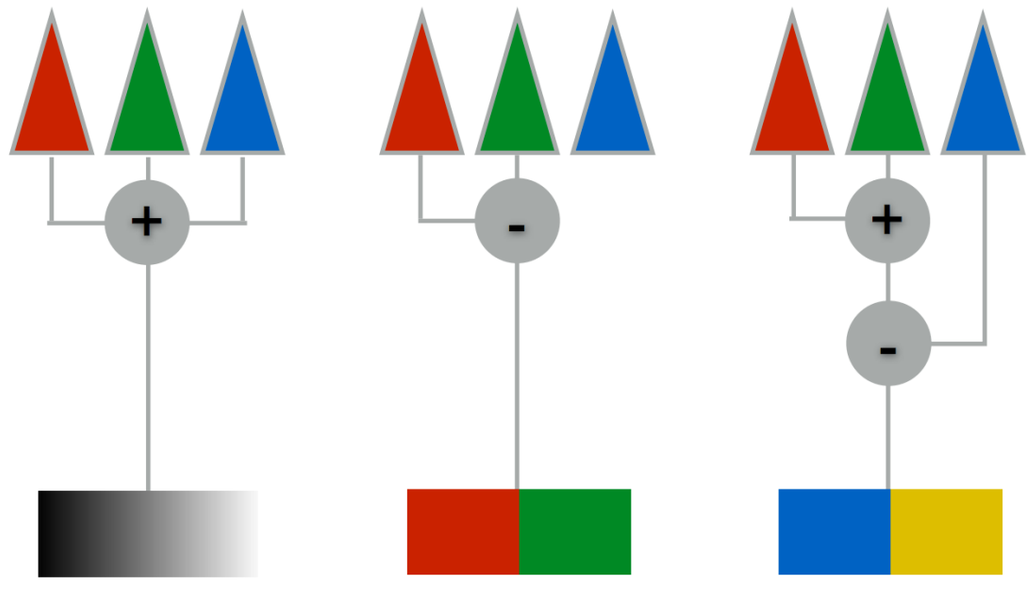
```
scale(20)?
scale(50)?
```

PRINCIPLES

Color Vision

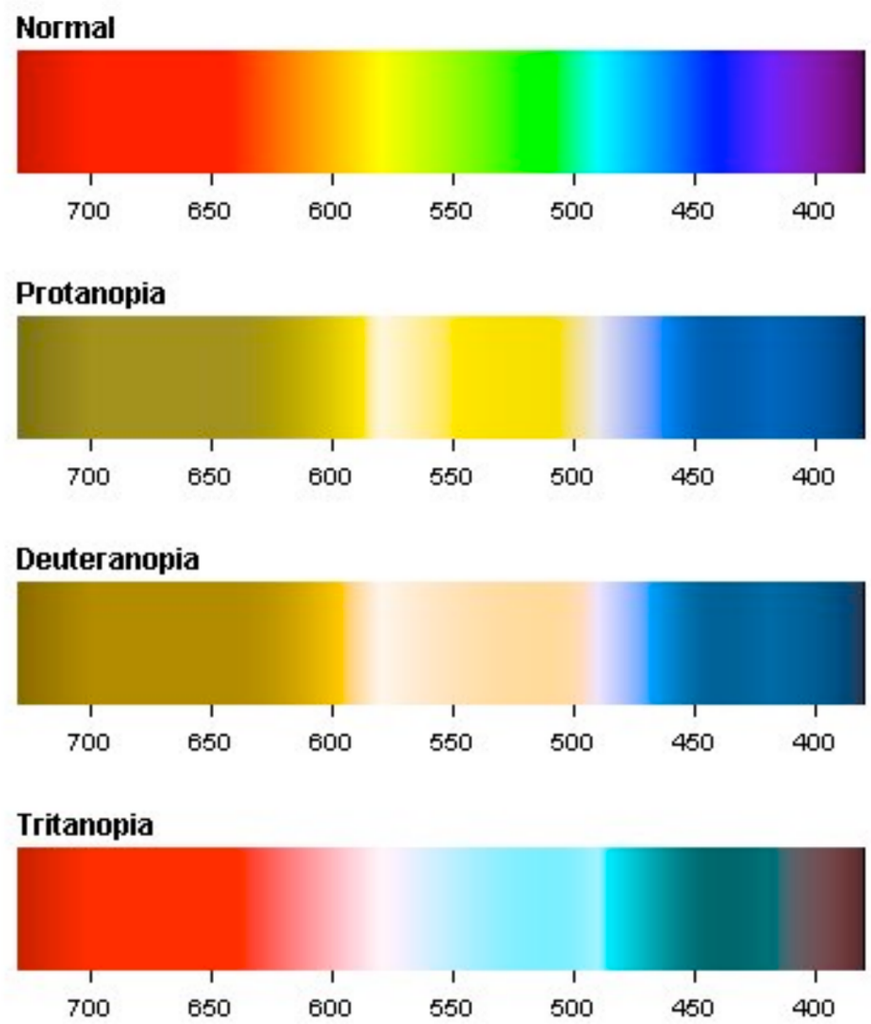


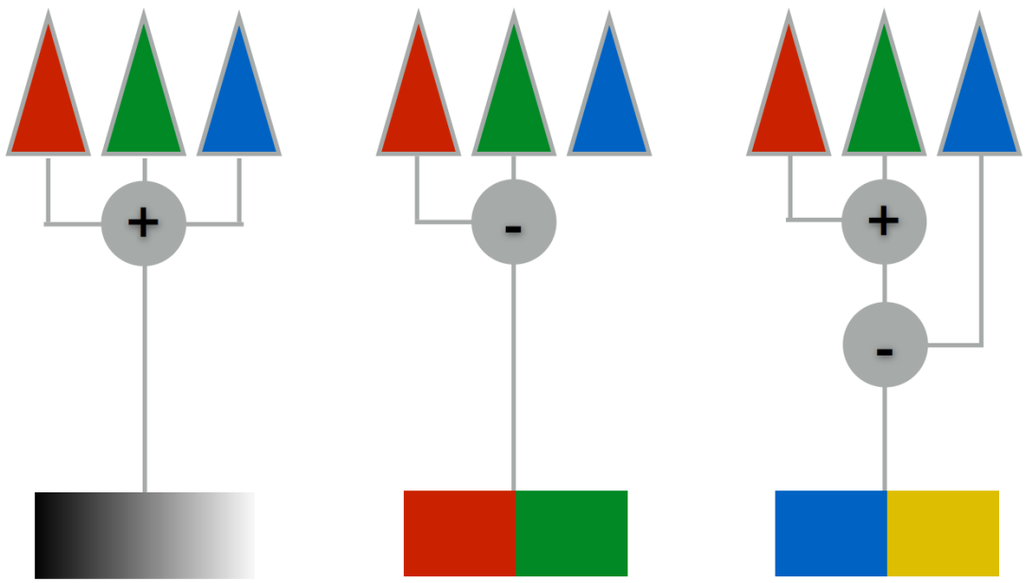




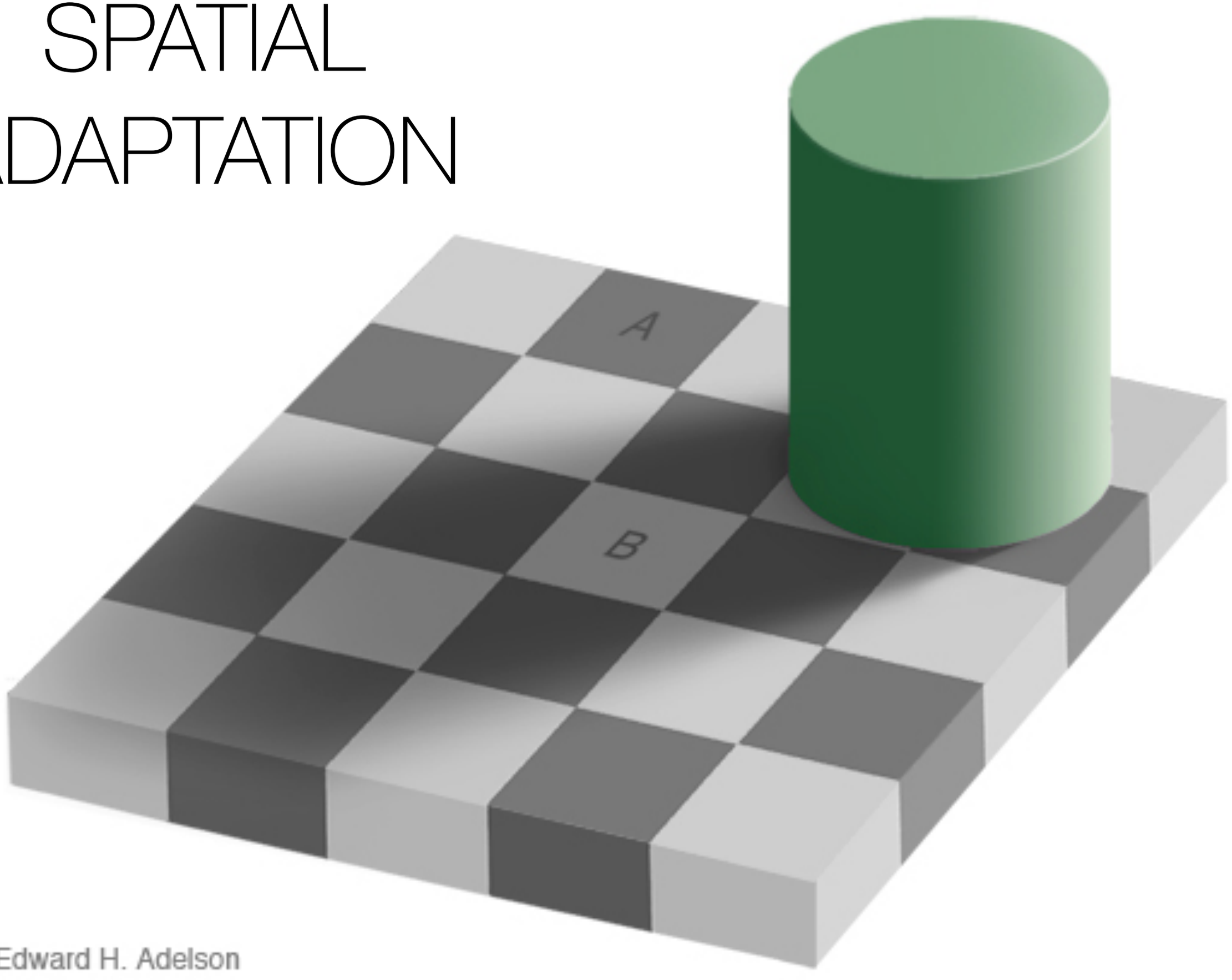
Color Vision Deficiencies

Don’t embarrass me: never use red-green as primary color discriminator!



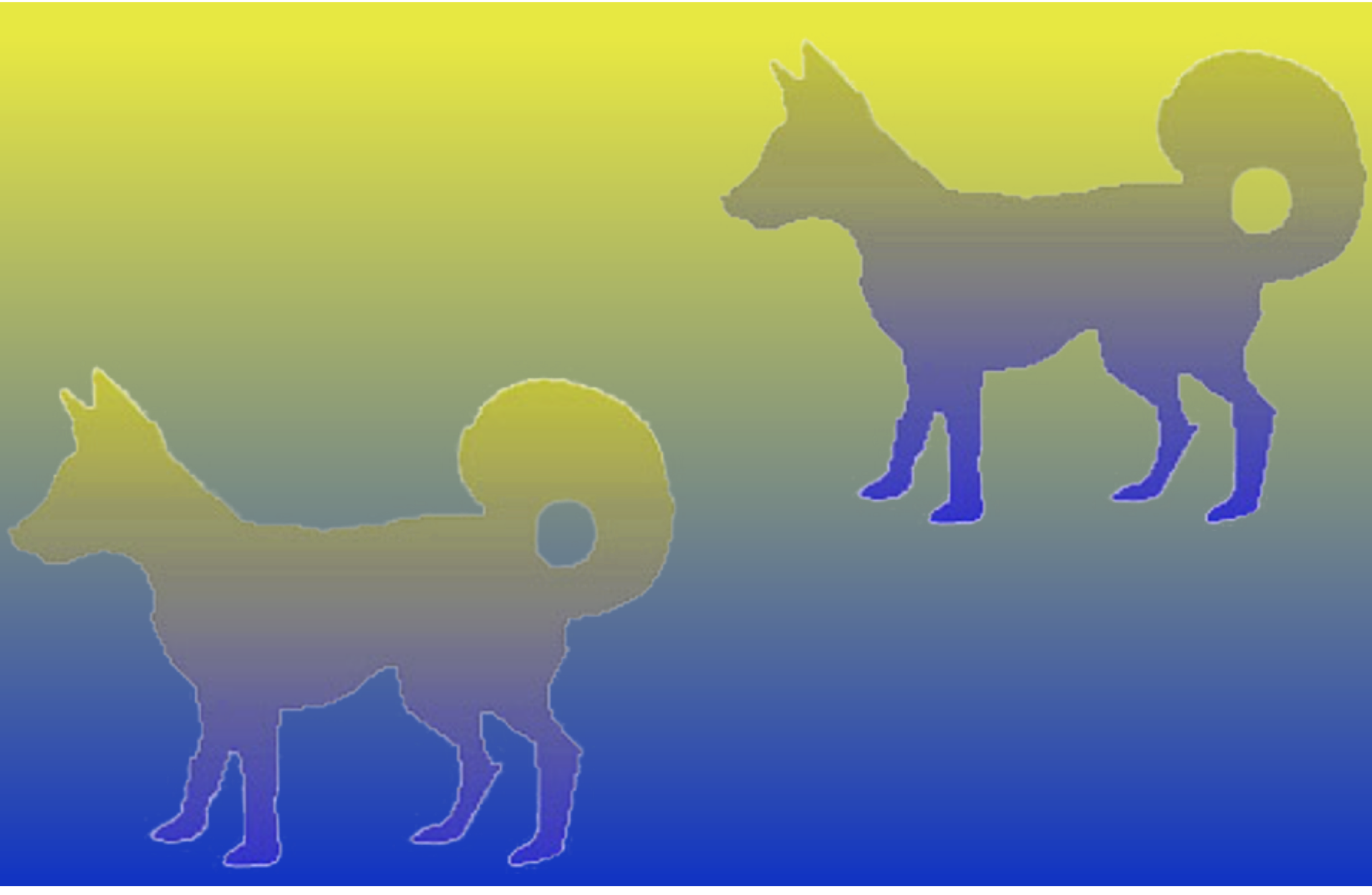


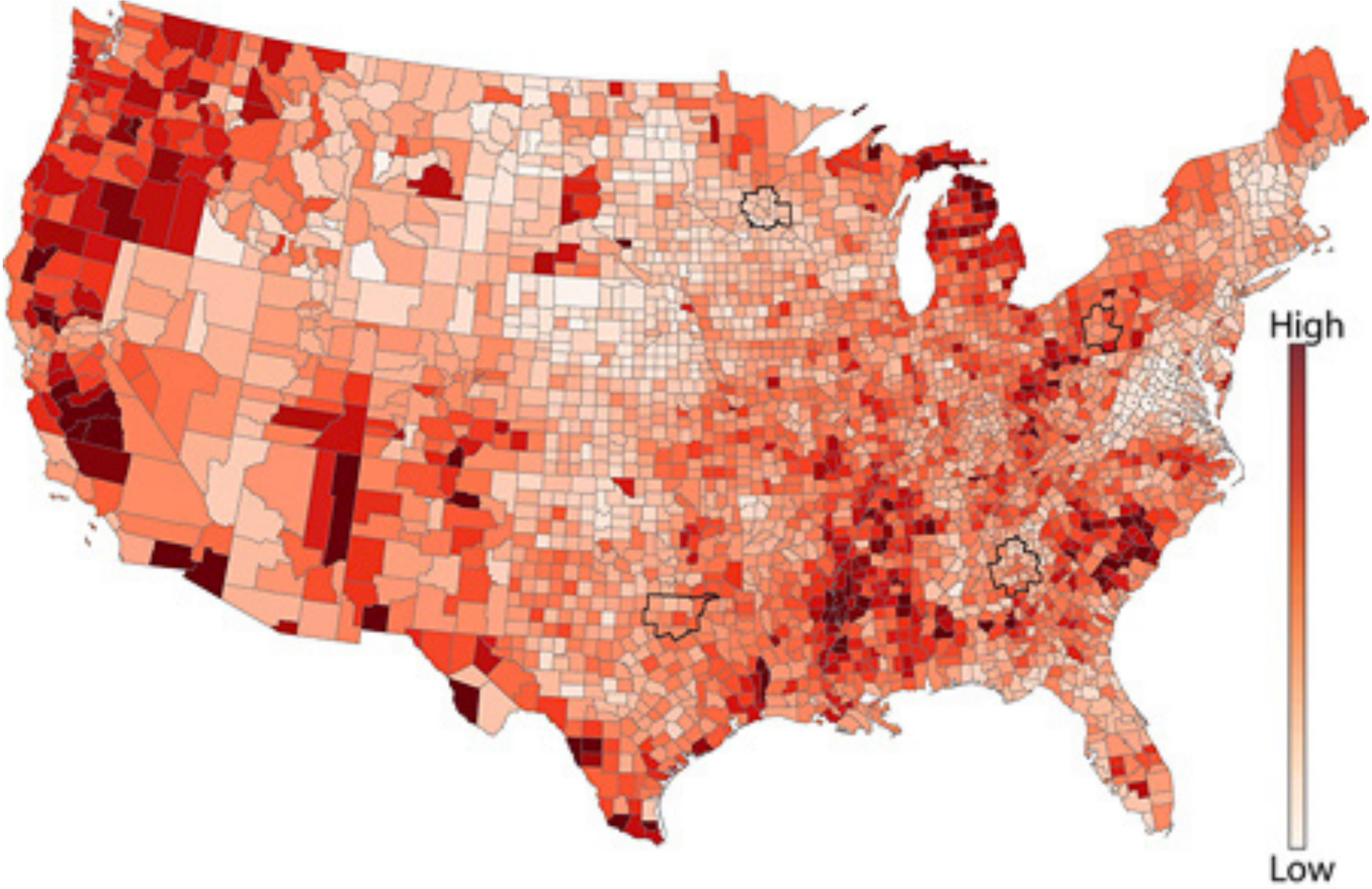
SPATIAL ADAPTATION



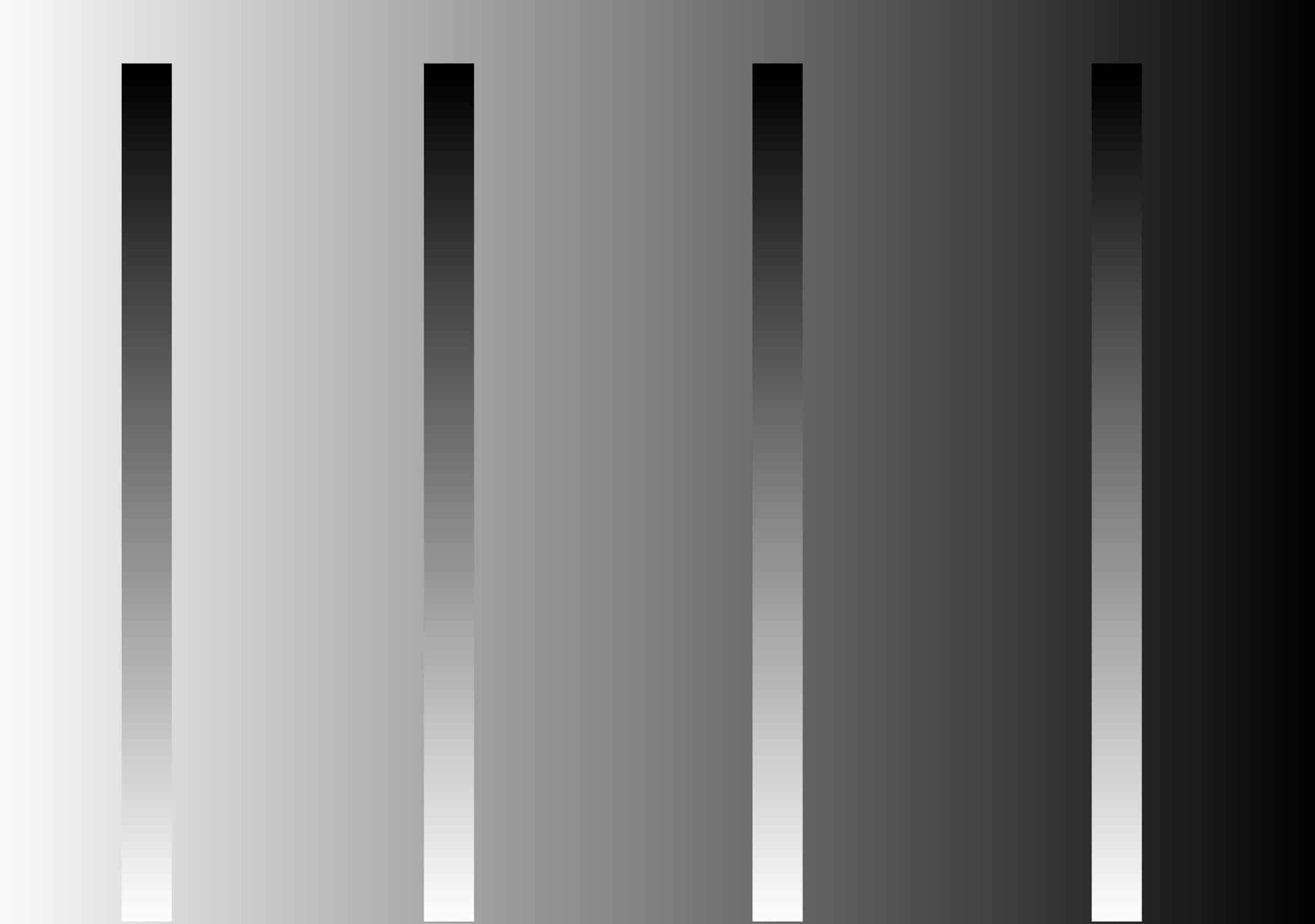
Edward H. Adelson

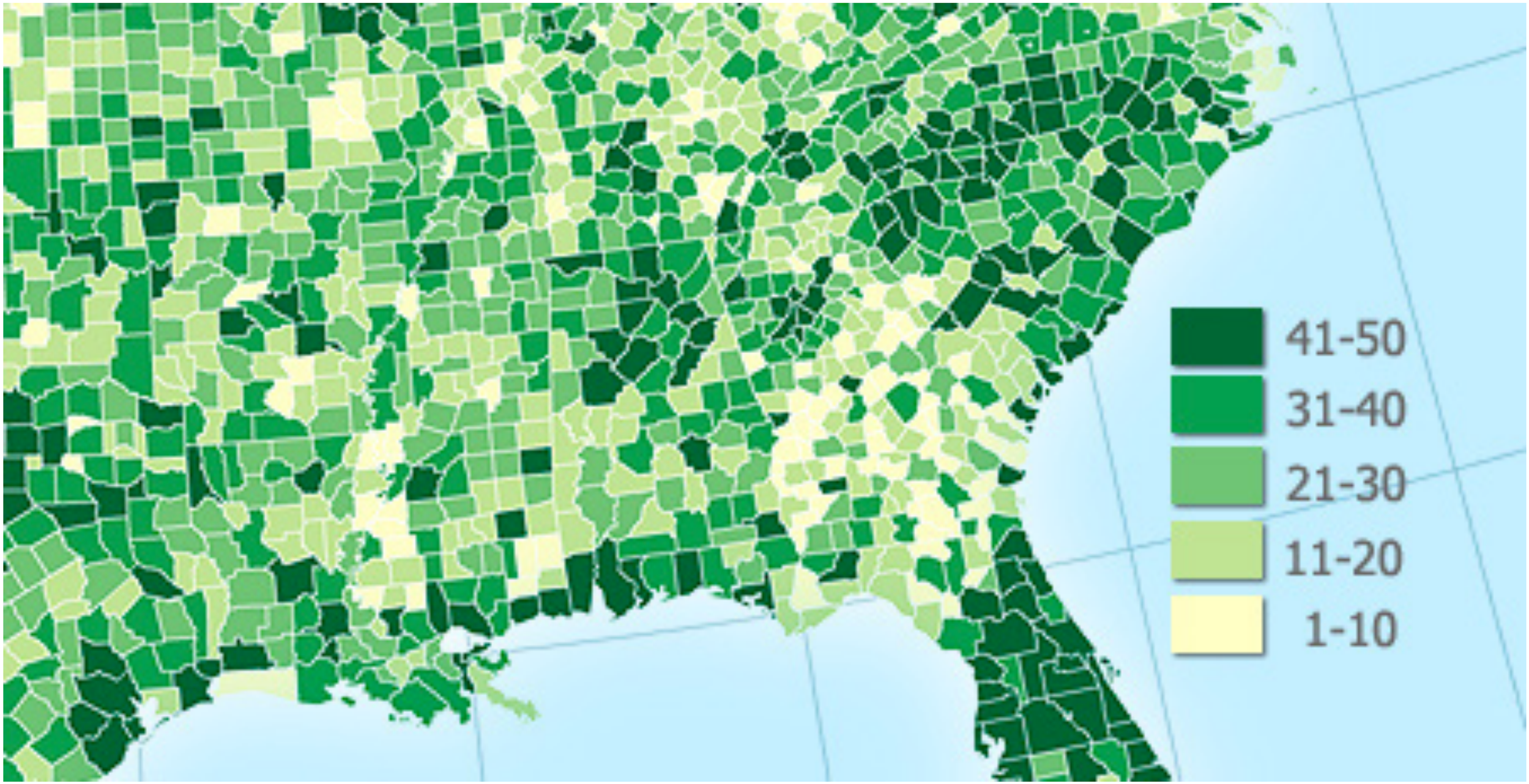
SPATIAL ADAPTATION





http://axismaps.github.io/thematic-cartography/





http://axismaps.github.io/thematic-cartography/

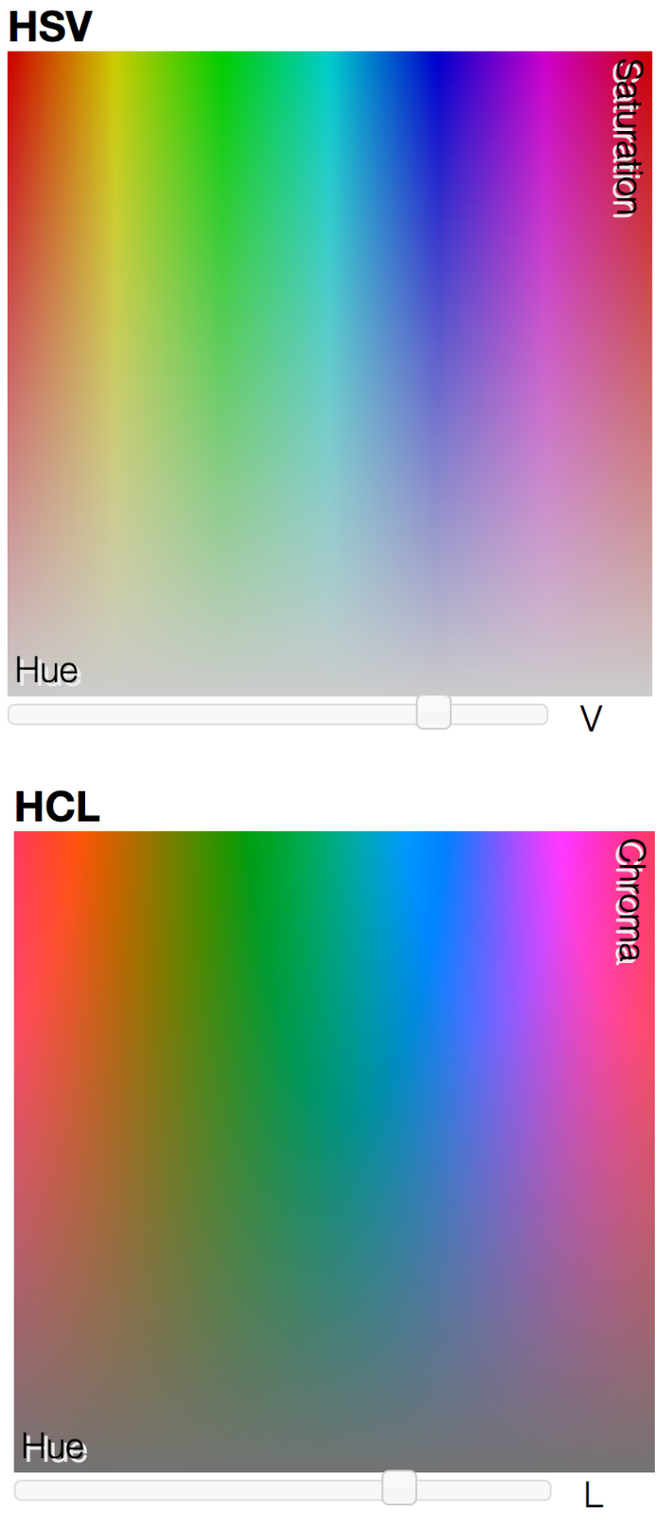
TEMPORAL ADAPTATION

http://www.moillusions.com/black-and-white-in-colour-again.html/13191556xteeocm7

Color Spaces

- RGB, CMYK, HSL: Device dependent. **Good for computers, bad for humans**

- Lab, Polar Lab (“HCL”): Perceptually-driven, better
  - **distances in coordinates are meaningful**
  - **coordinates are perceptually meaningful**

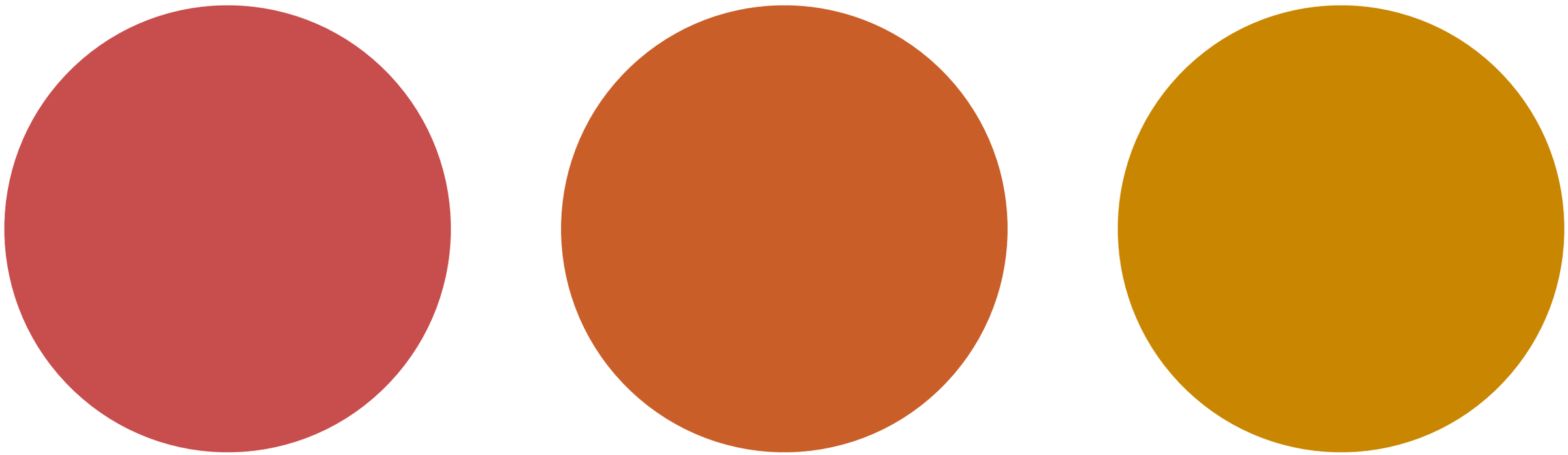


Do not rely only on hue boundaries to depict shape

Do not rely only on hue boundaries to depict shape

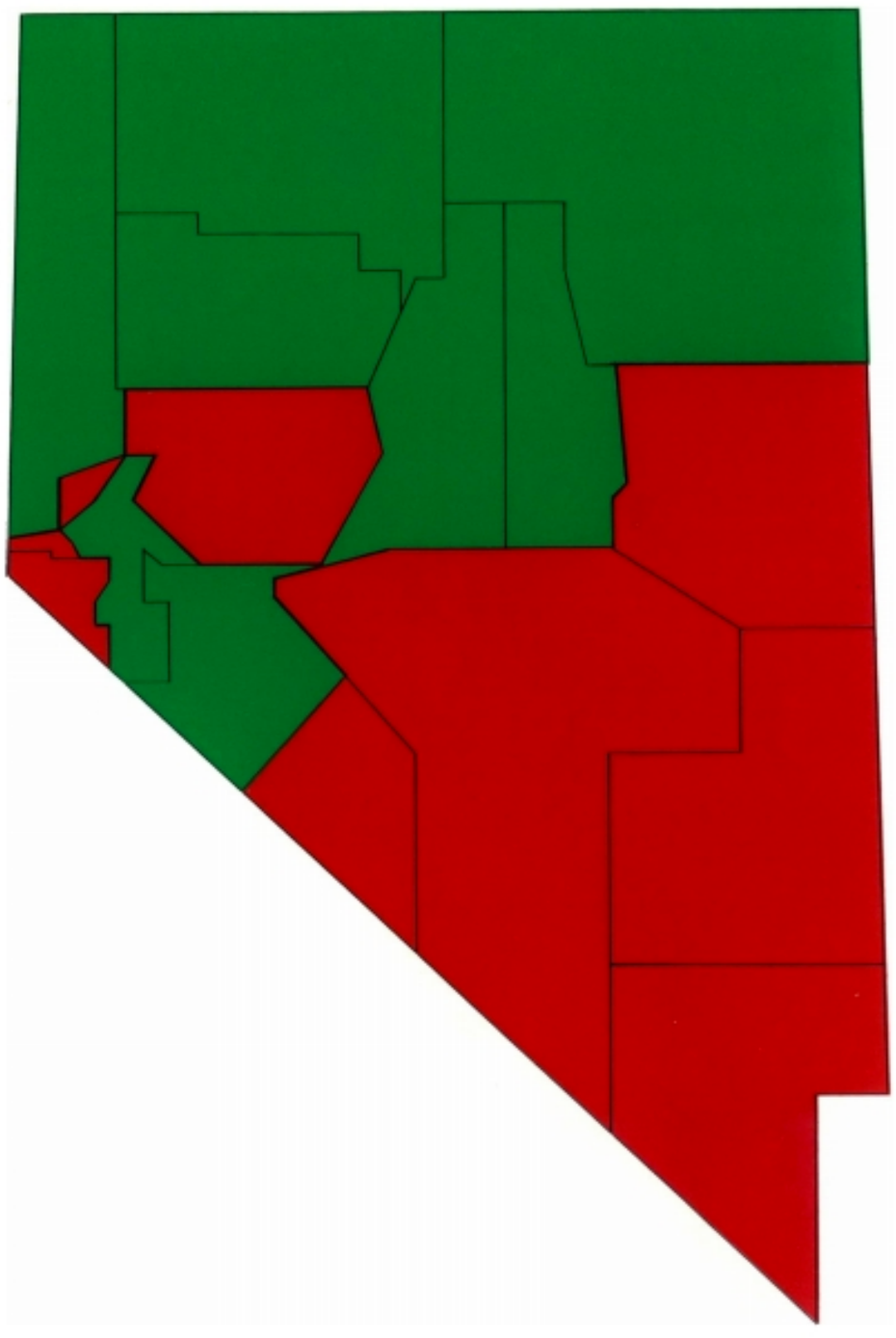
Area affects saturation perception

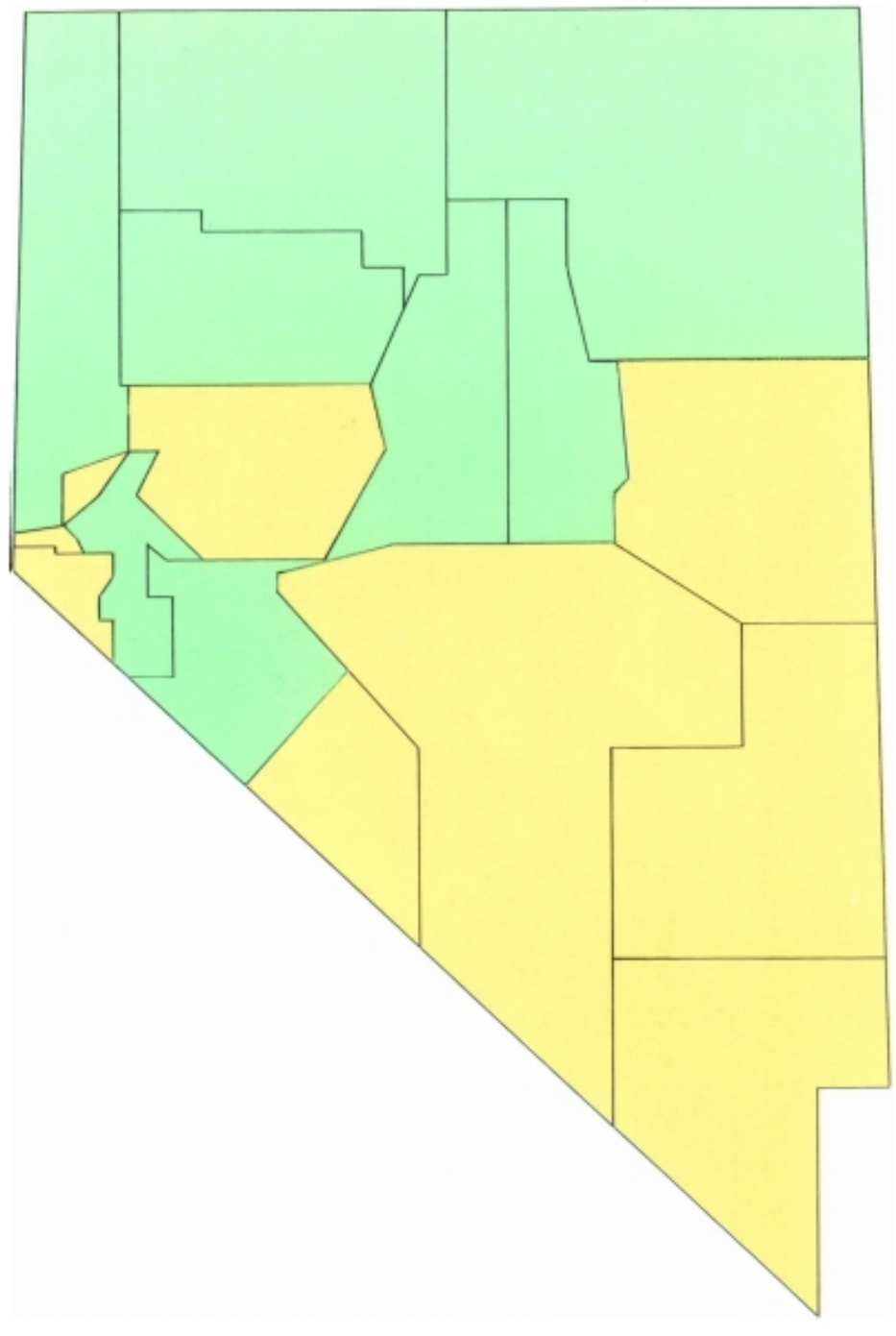
Area affects saturation perception



Saturation affects area perception

102 © The American Statistician, May 1983, Vol. 37, No. 2





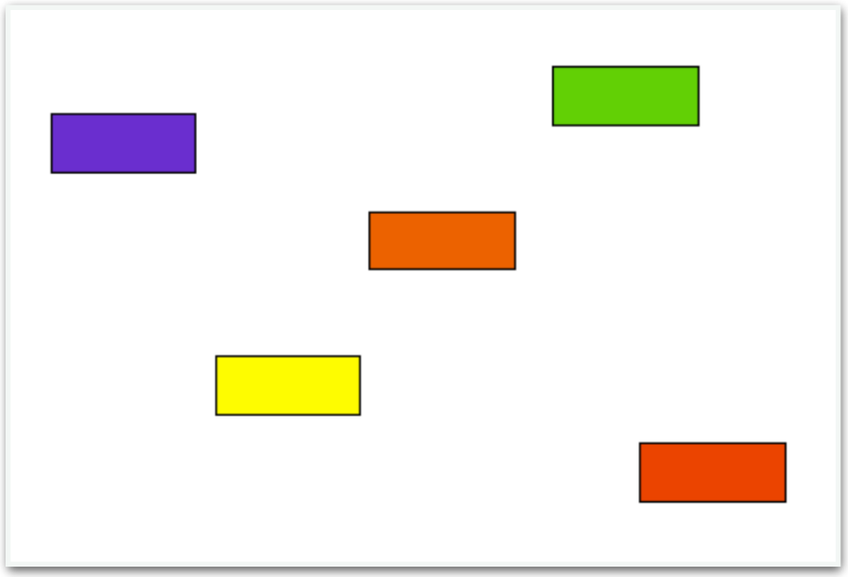
Area affects saturation perception

Saturation affects area perception

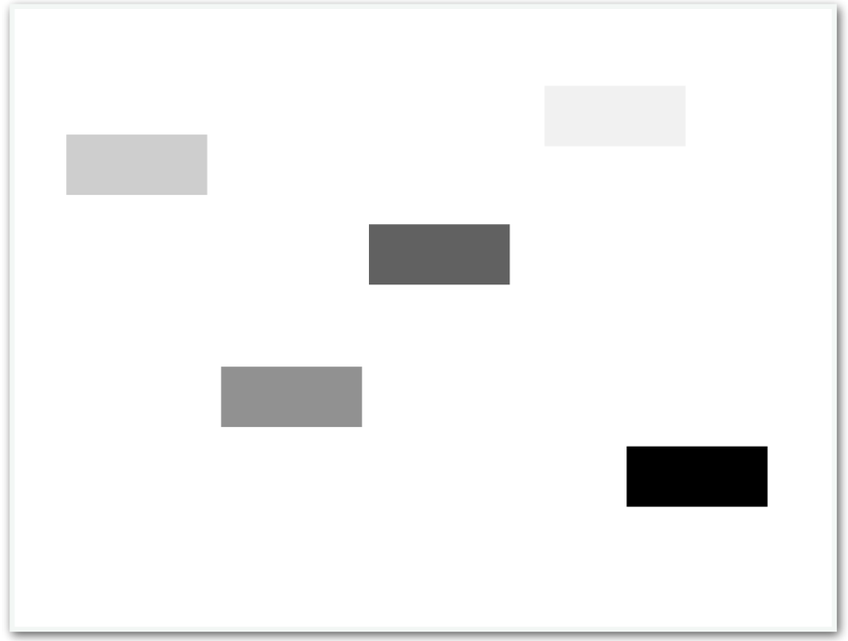
Do not change saturation if task involves area judgement

Do not change area if task involves saturation judgement

Consider implied ordering in color channels



Hue



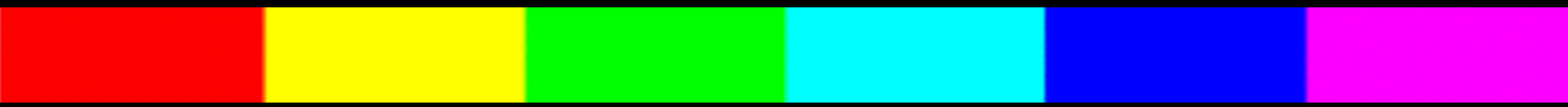
Luminance

Saturation

If you're going to use the rainbow colormap, use an **isoluminant** version, **quantize** it, or **both**

Bad

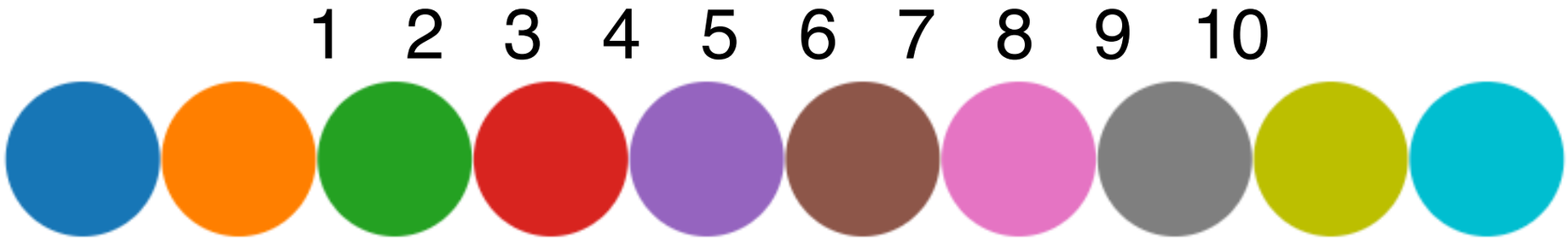
Better



Be aware of implied and perceptually forced color relationships

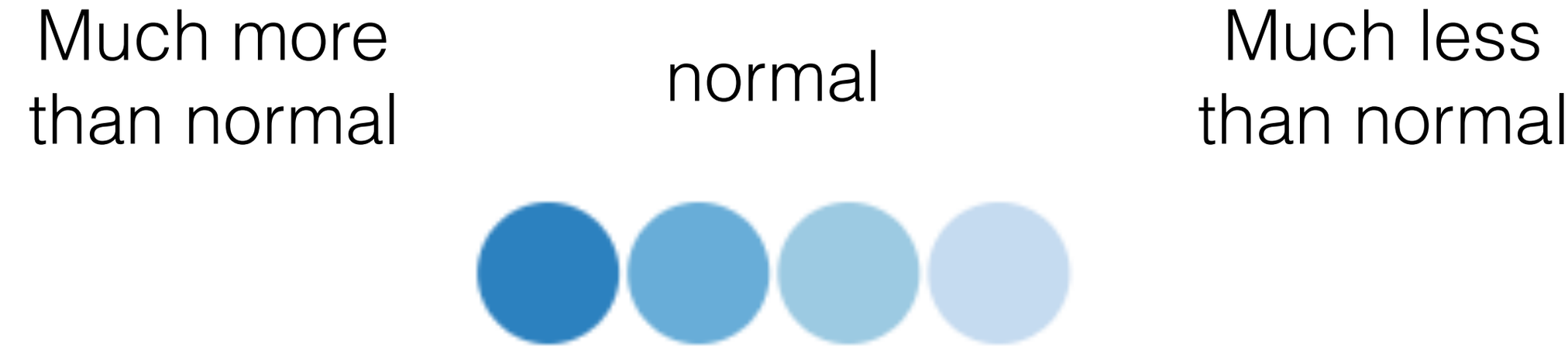
For categorical data, use color only when you have few categories (less than 10)

Q: You're given this color scale for a **map of temperatures**. What's wrong?



http://bl.ocks.org/aaizemberg/78bd3dade9593896a59d

Q: You're given this color scale for a map of rainfall variation **(from much less than normal, to normal, to much more than normal)**. What's wrong?



http://bl.ocks.org/aaizemberg/78bd3dade9593896a59d

Q: You’re given this color scale for a map of **locally popular religious views across a country**. What’s wrong?

http://bl.ocks.org/aaizemberg/78bd3dade9593896a59d

THE STANDARD VISUAL CHANNELS

Position

- Horizontal
- Vertical
- Both

Color

Shape

Tilt

Size

- Length
- Area
- Volume

Cleveland/McGill perception papers

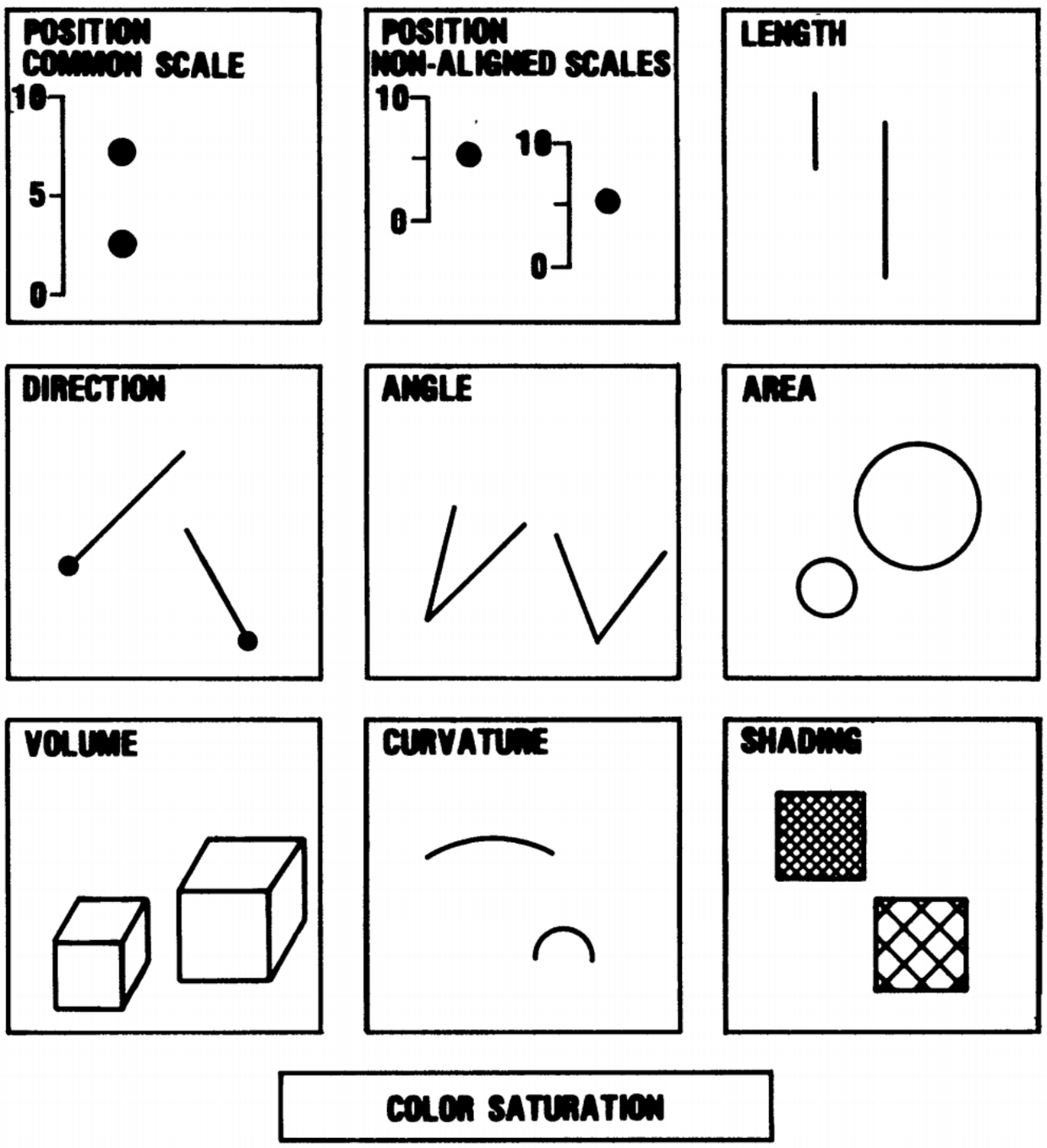
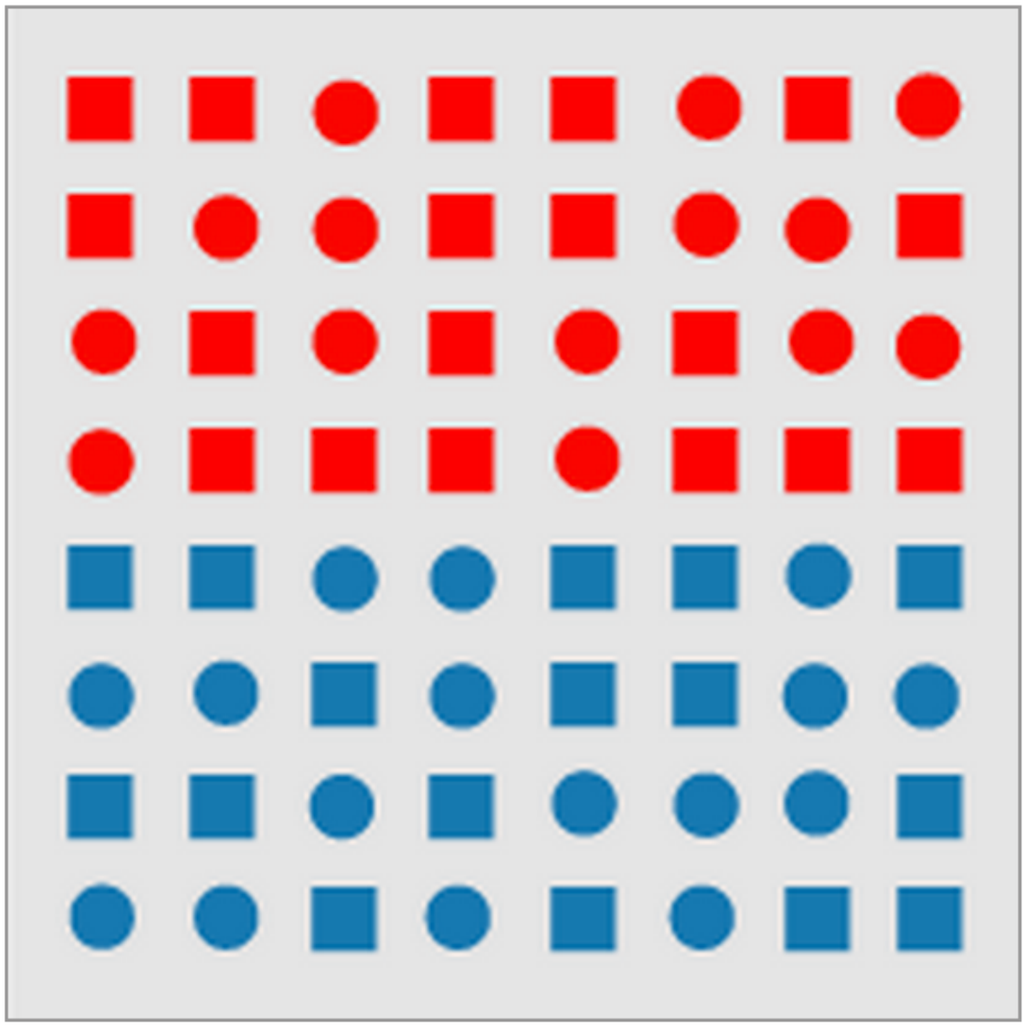


Figure 1. Elementary perceptual tasks.

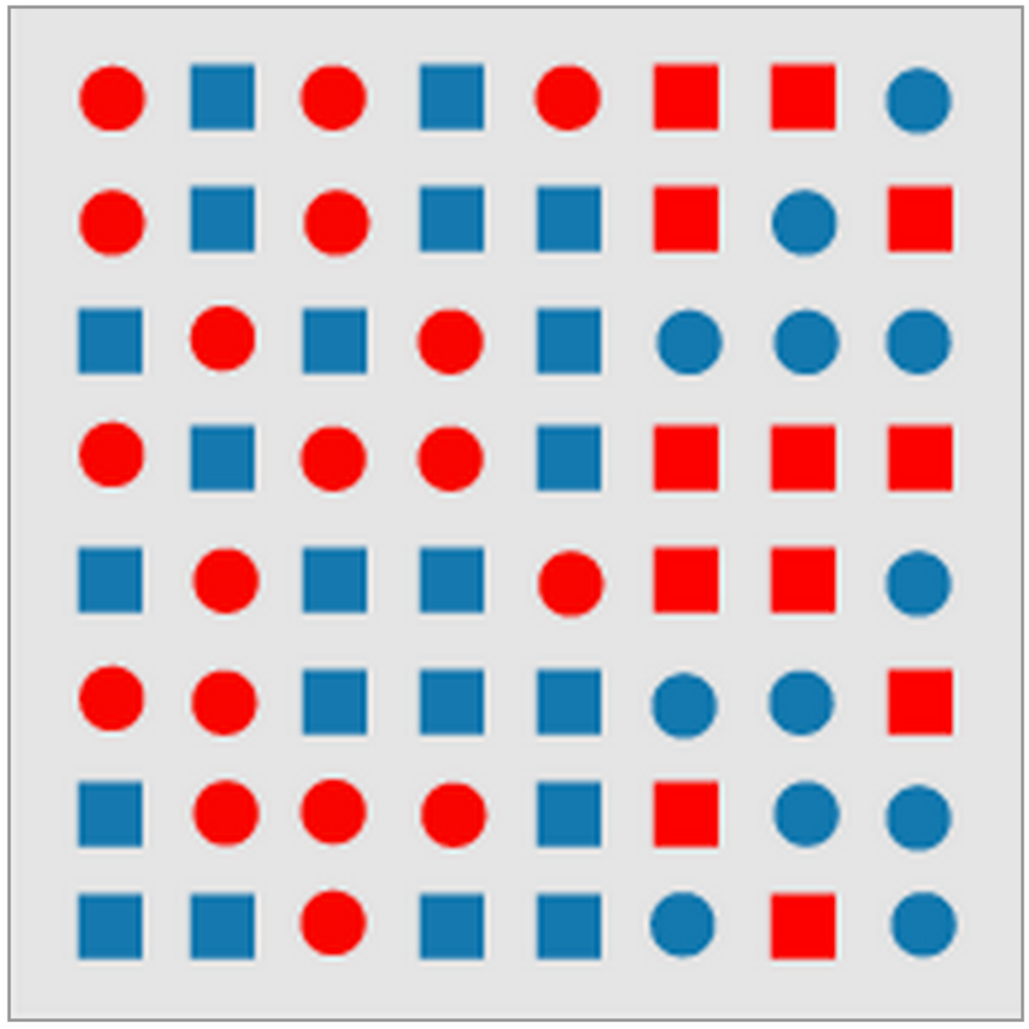
1. Position along a common scale
2. Positions along nonaligned scales
3. Length, direction, angle
4. Area
5. Volume, curvature
6. Shading, color saturation

PREATTENTIVENESS,

OR “VISUAL POP-OUT”



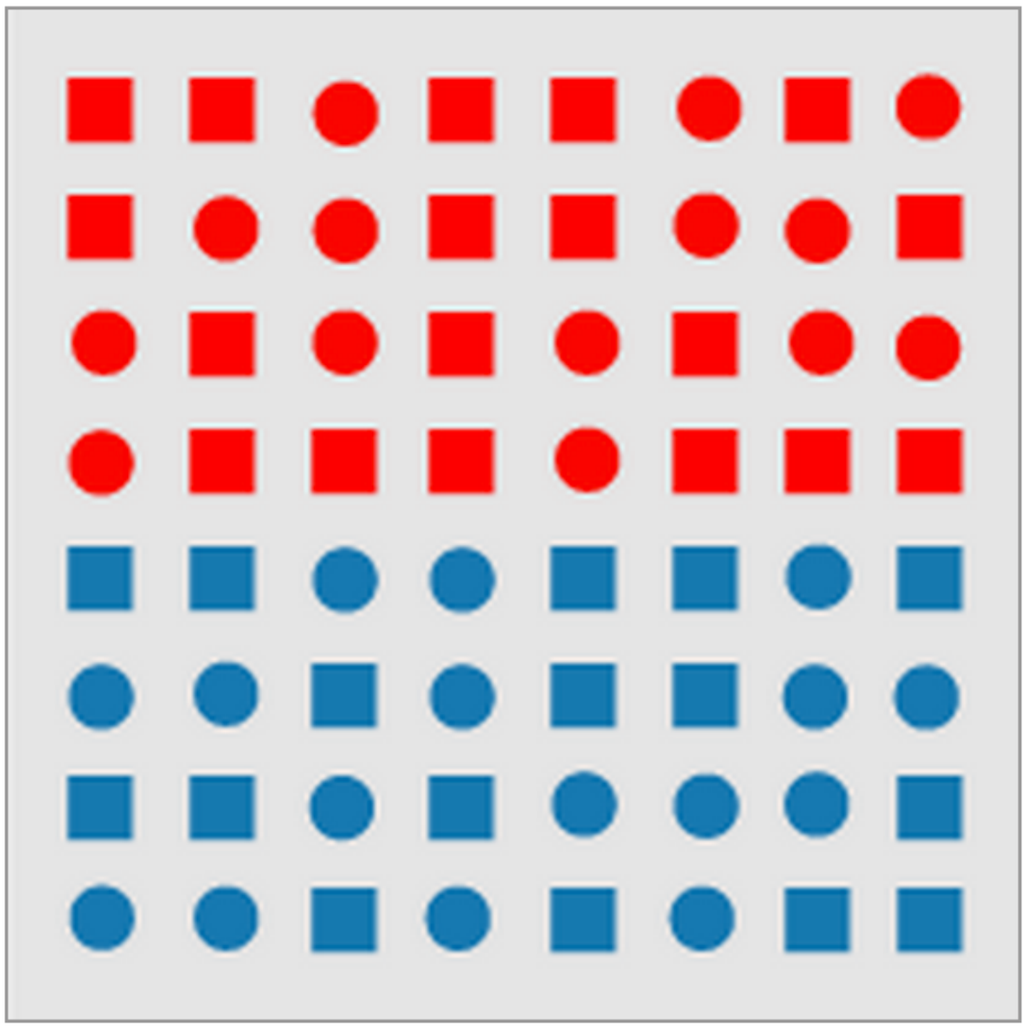
(a)



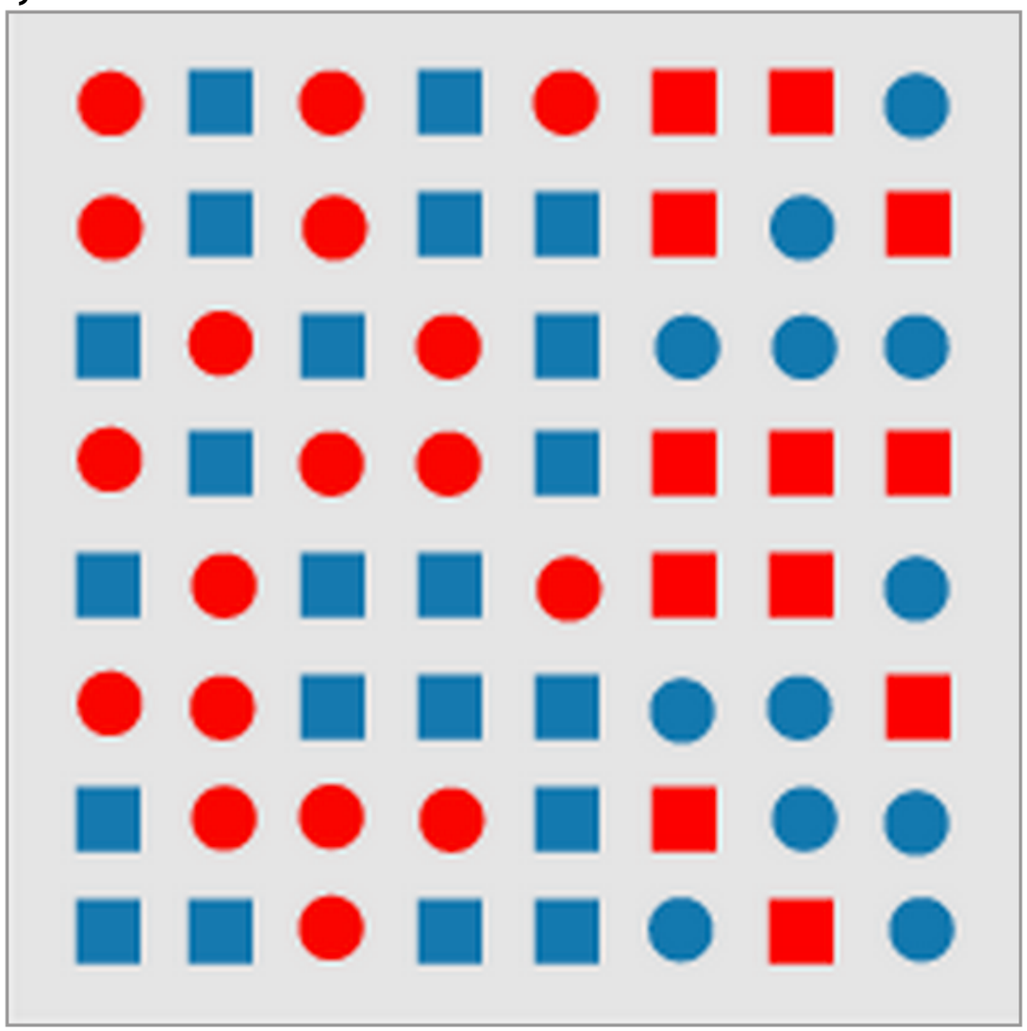
(b)

```
function(d) {
  if (d.row > 4)
    return “blue”;
  else
    return “red”;
}
```

```
function(d) {
  if (d.column > 4) {
    if (d.shape === “square”)
      return “red”;
    else
      return “blue”;
  } else {
    if (d.shape === “square”)
      return “blue”;
    else
      return “square”;
  }
}
```



(a)



(b)

Preattentiveness

(mostly) works one-channel-at-a-time.

Integral vs. Separable Channels

- Do humans perceive values “as a whole”, or “as things that can be split”?

- **Use separable channels for multi-variate encodings**

Integral vs. Separable Channels

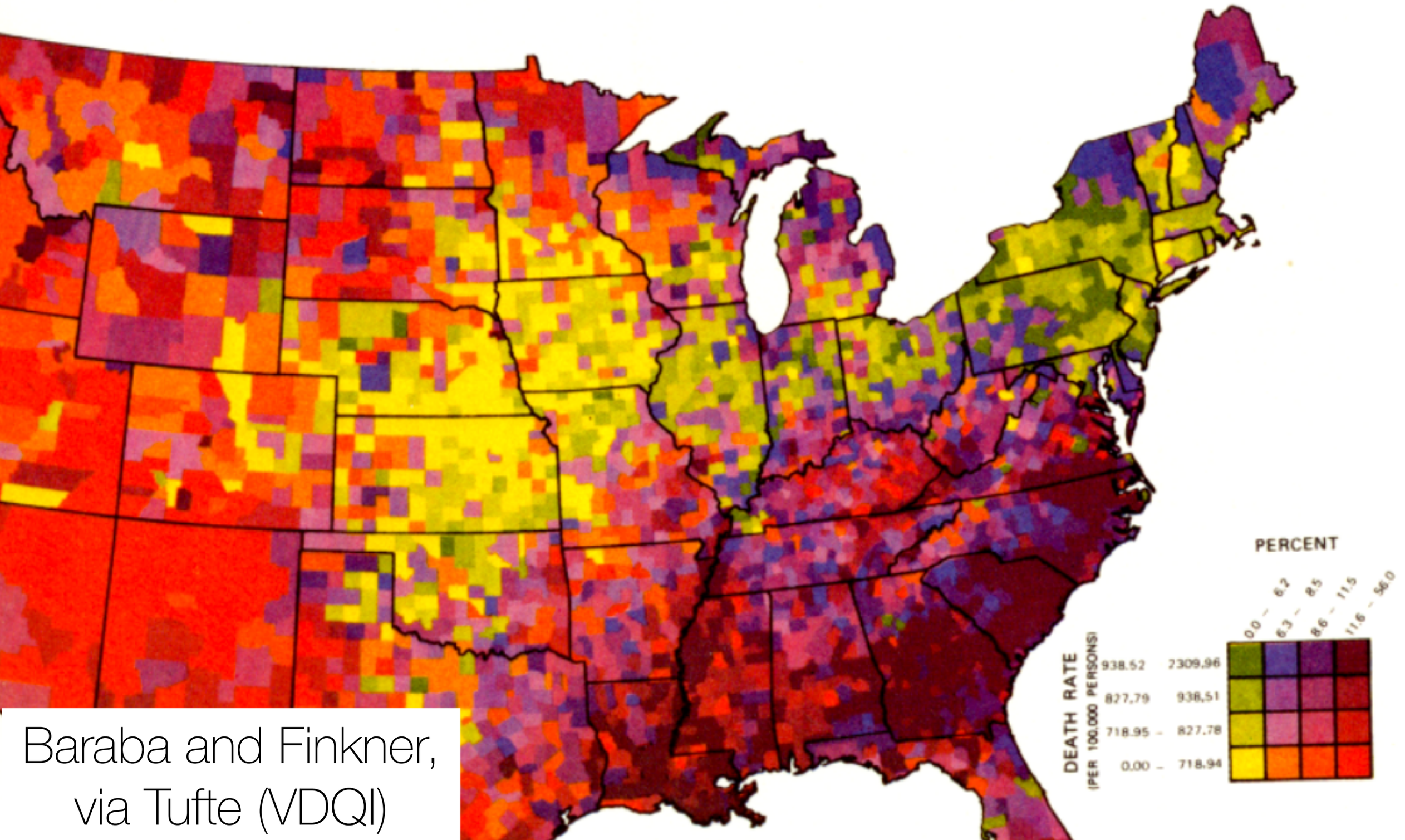
Separable ⟷ Integral

color x location **color x shape** **x-size x y-size**

color x motion **size x orientation** **r-g x y-b**

Colin Ware, 2004, p180

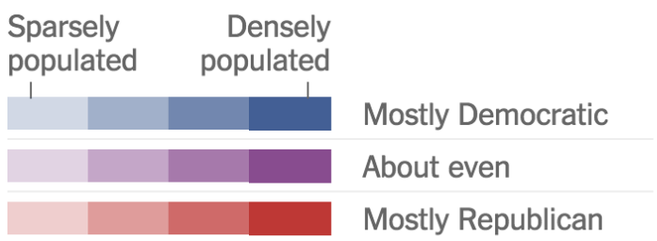
Bivariate Color Maps (This one is bad)

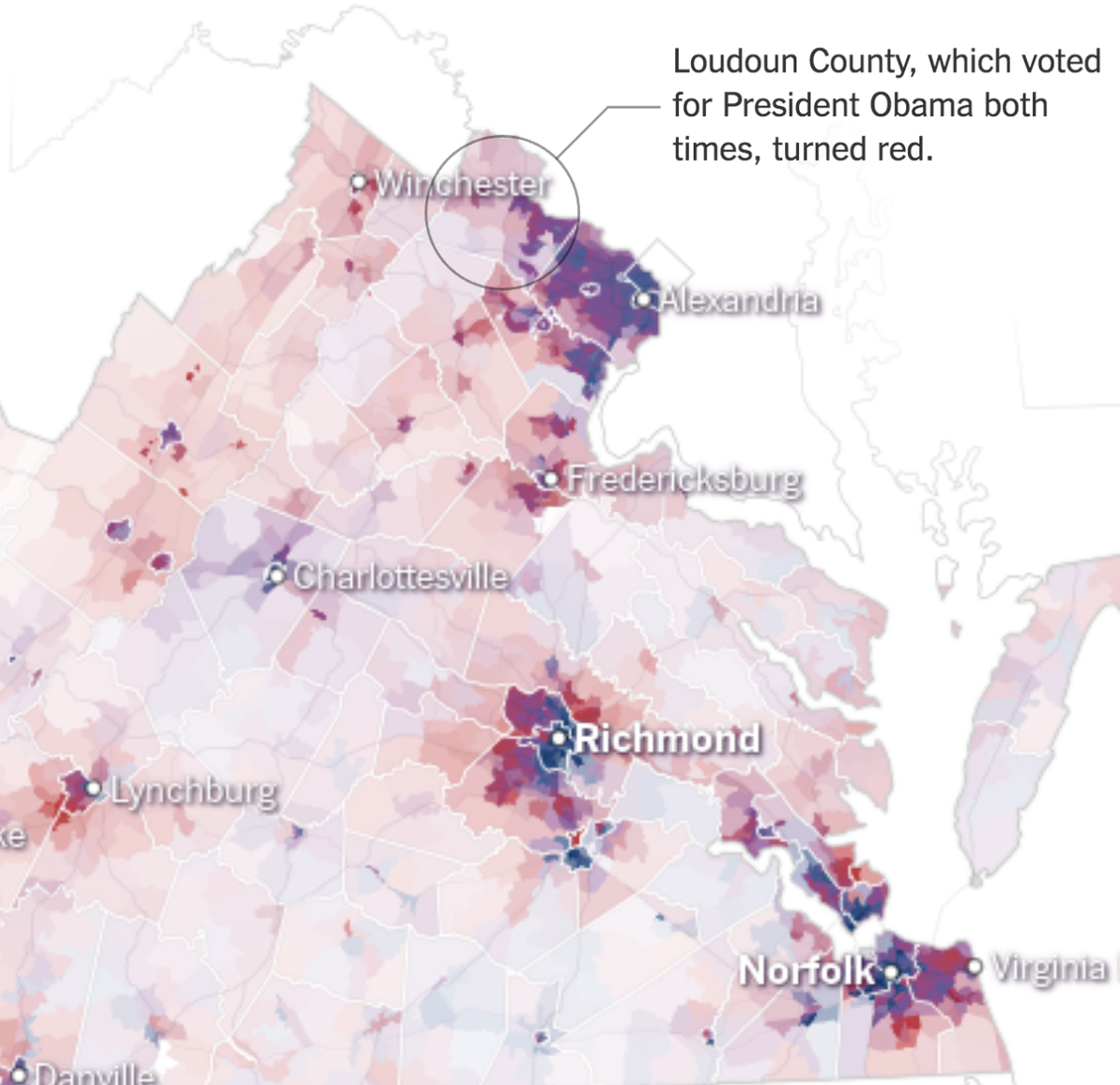


Baraba and Finkner, via Tufte (VDQI)

Bivariate Color Maps (This one is pretty good)

http://www.nytimes.com/interactive/2014/11/04/upshot/senate-maps.html





Q: Why?

To get (some) separability in colors, use Luminance, Saturation, and Hue

INTERACTION, FILTERING, AGGREGATION

Q: Your data has five different attributes. How to show all relationships?

- “use five different channels in a single plot”
  - **wrong answer**: we lose preattentiveness, and there aren’t that many good channels

What if there’s too much data?

- Sometimes you can’t present all the data in a single plot

- Show multiple good plots and **linked views**

  - **Interaction**

What if there’s too much data?

- Sometimes you can’t present all the data in a single plot

- **Interaction**: let the user drive what aspect of the data is being displayed

- **Filtering**: Selectively hide some of the data points

- **Aggregation**: Show visual representations of subsets of the data

Shneiderman's "Visual information seeking mantra"

Overview first,
zoom and filter,
then details-on-demand

Overview first:

Before all else, show a “high-level” view, possibly through appropriate aggregation

Zoom and Filter:

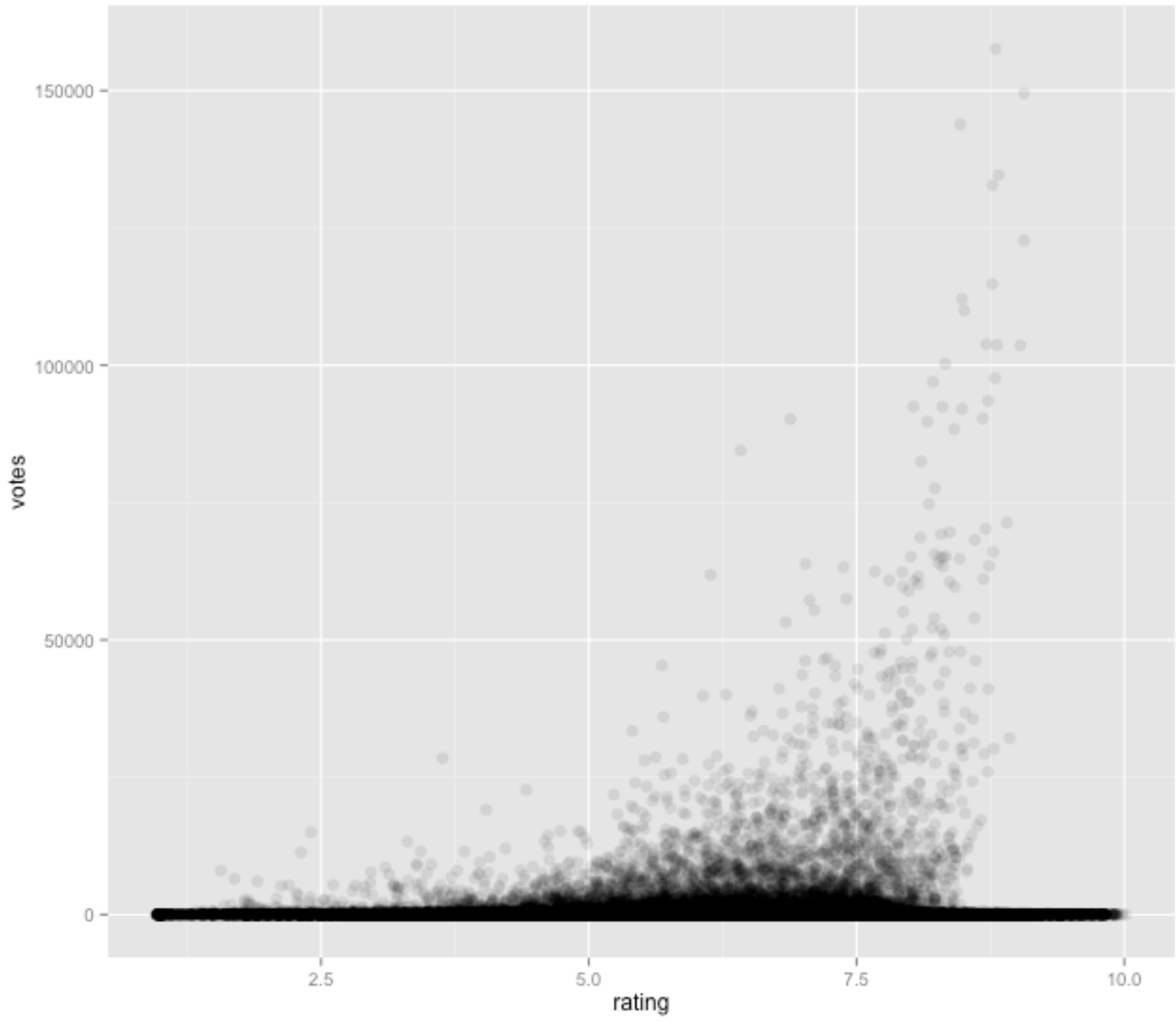
Use interaction to create user-specified views

Details on Demand:

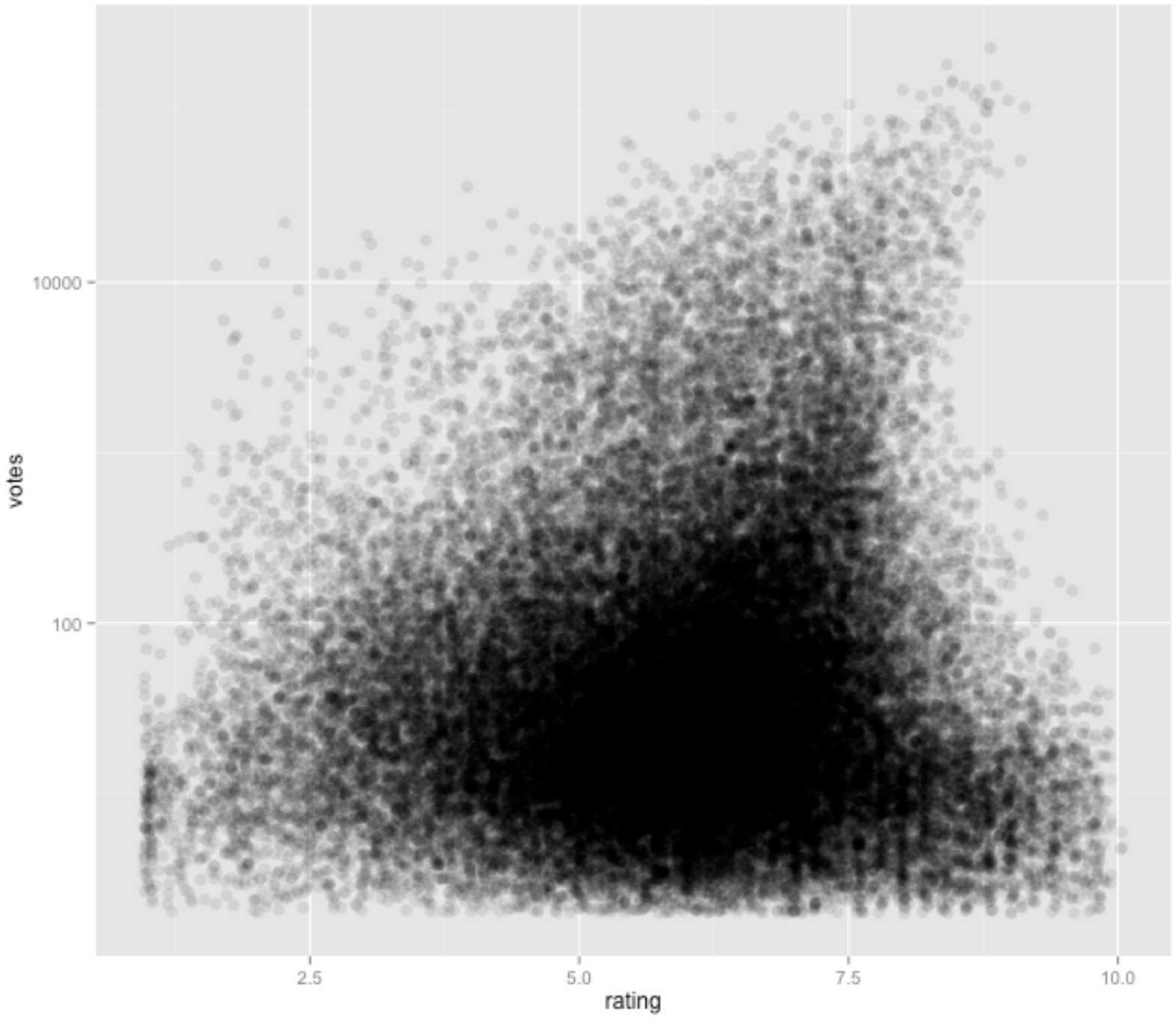
Individual points or attributes should be available, but only as requested

TECHNIQUES:
SPATIAL ARRANGEMENTS

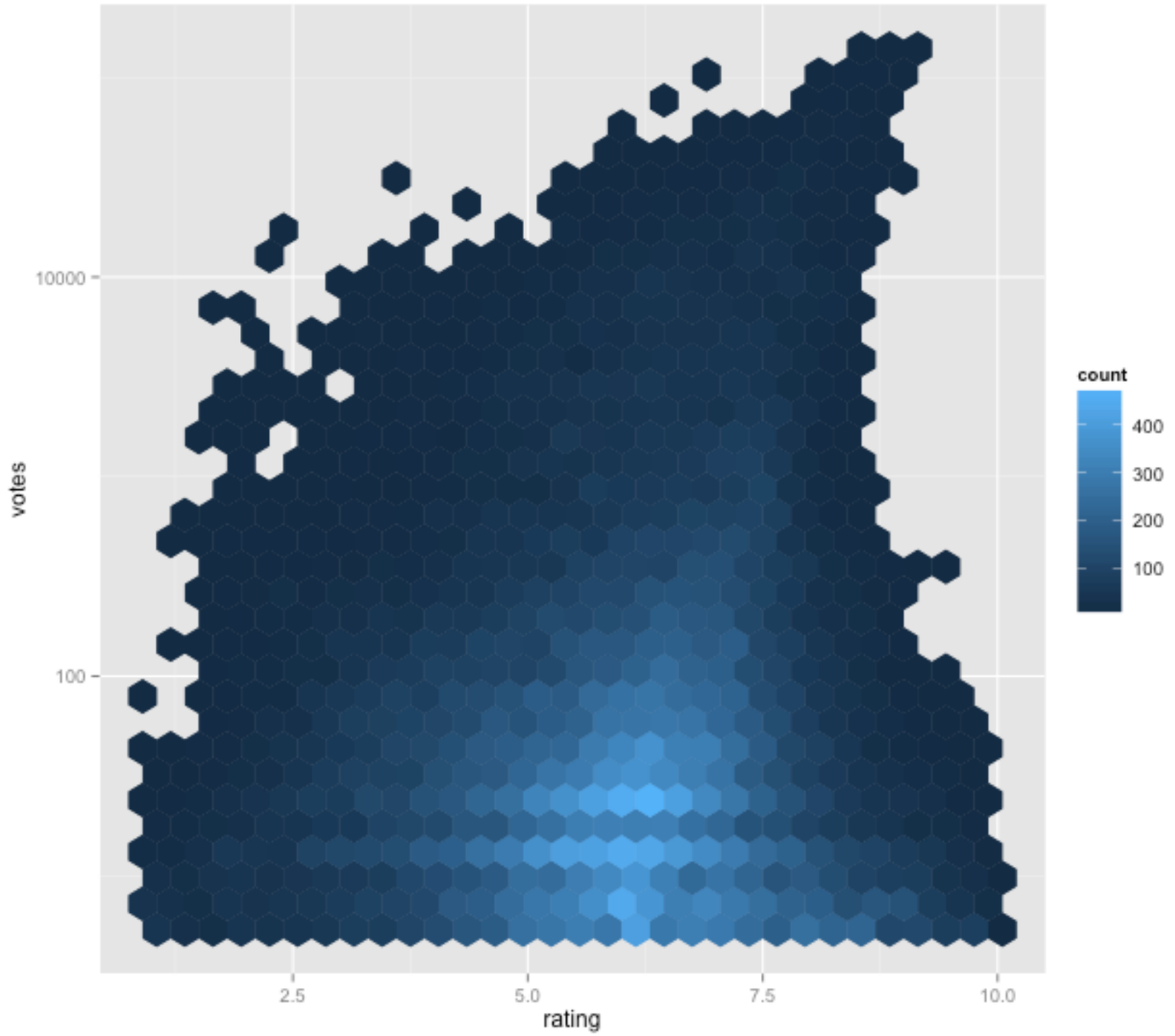
Transformations



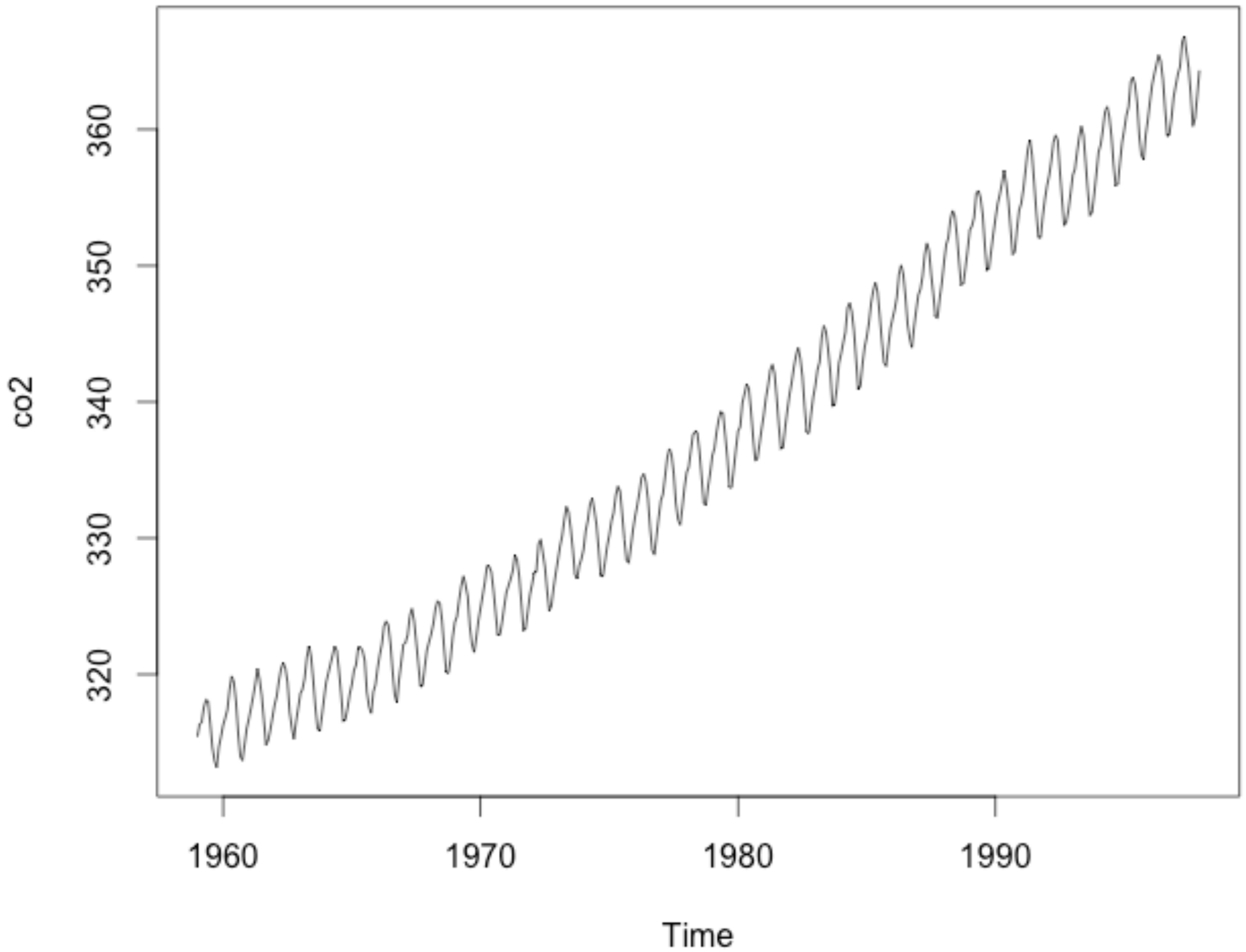
Transformations



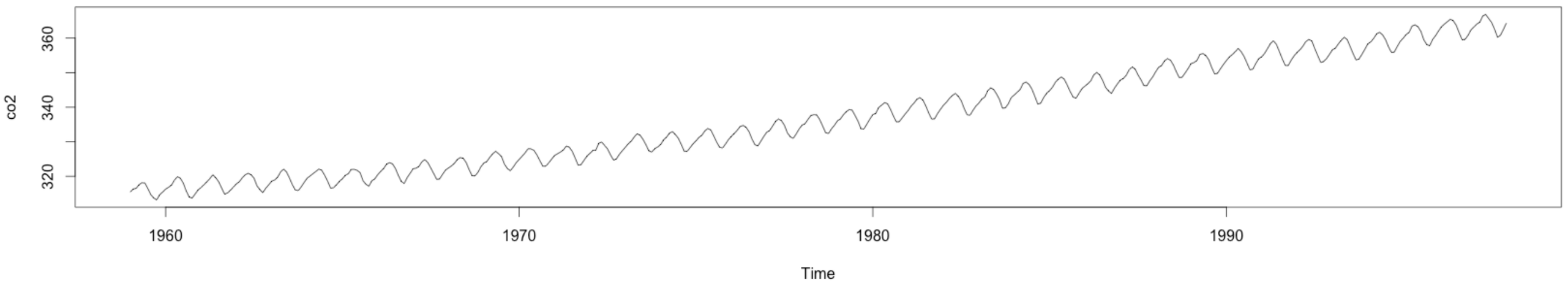
Transformations



Line Charts

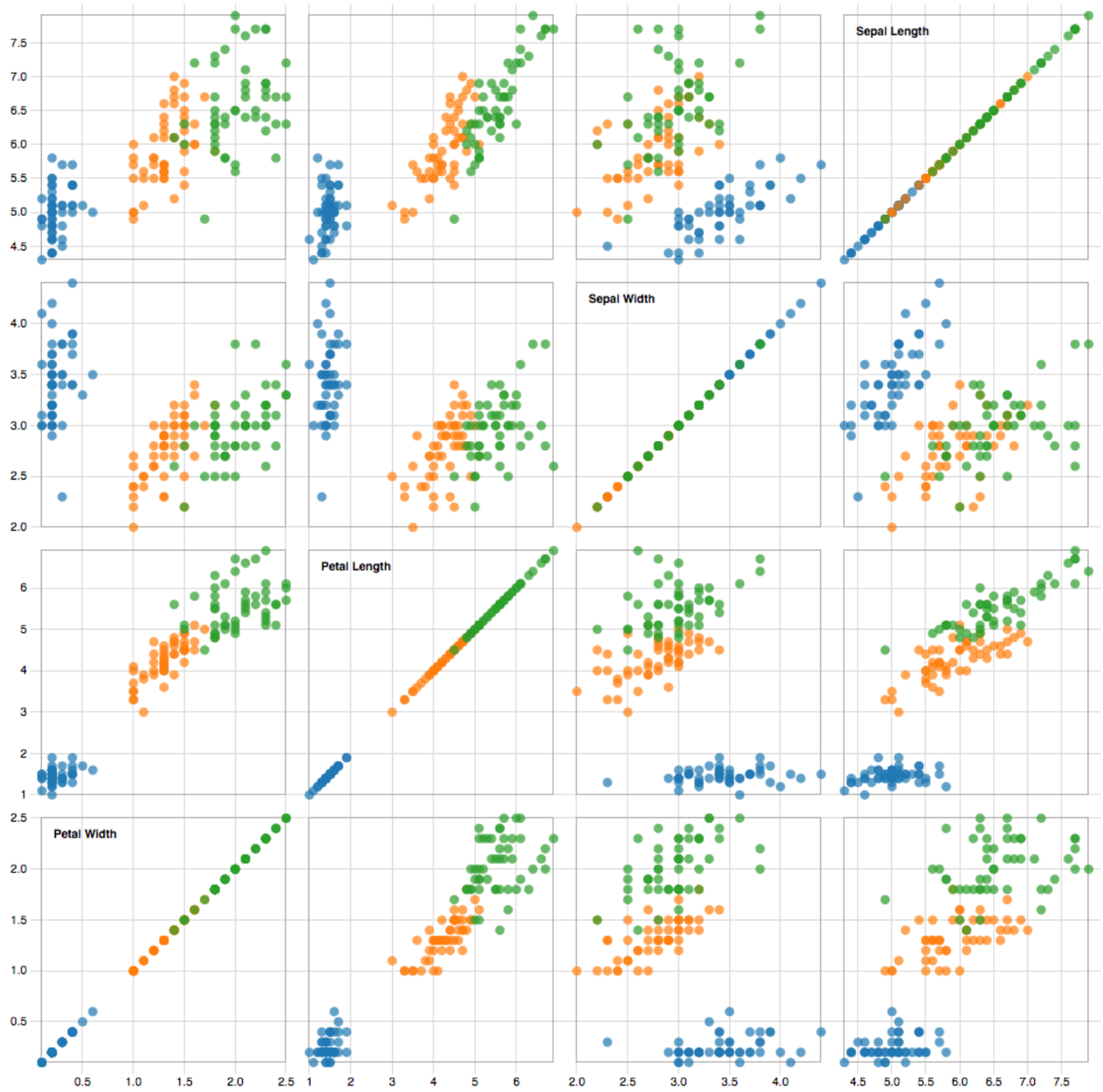


Bank to 45 degrees



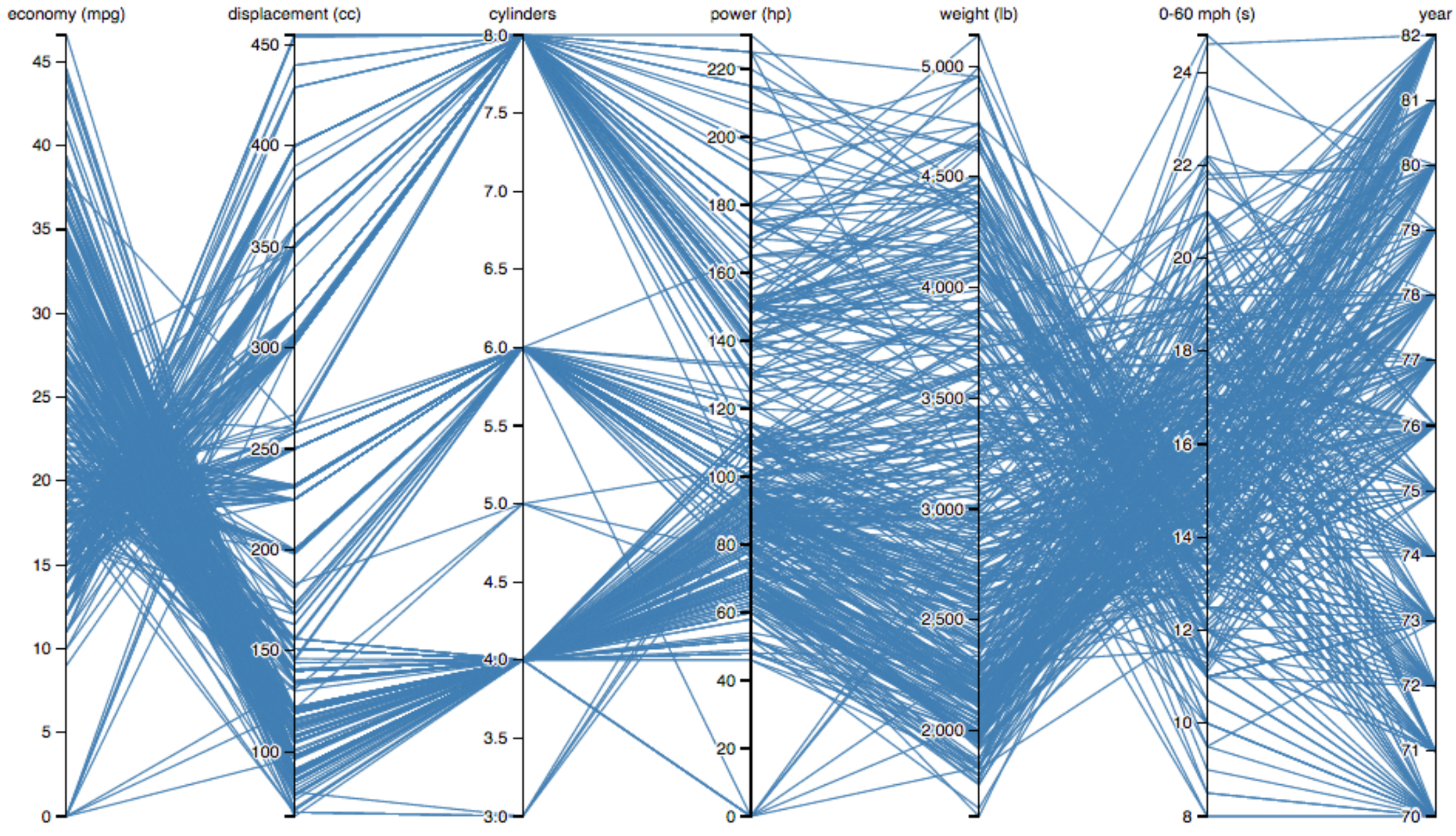
Many dimensions

Small Multiples



http://bl.ocks.org/jasondavies/1341281

Parallel Coordinates



http://bl.ocks.org/jasondavies/1341281