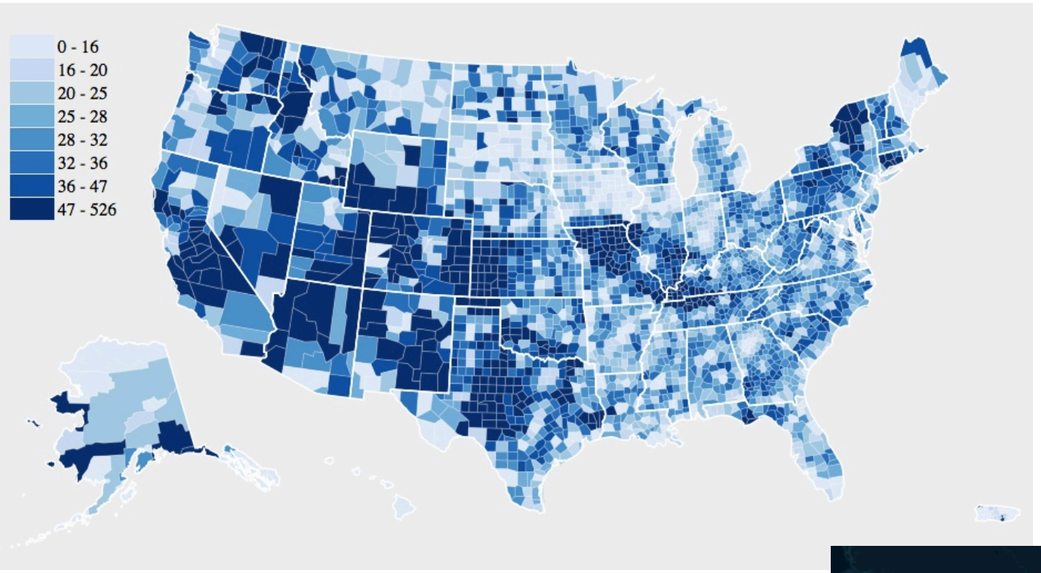
Visualizing Geo Data

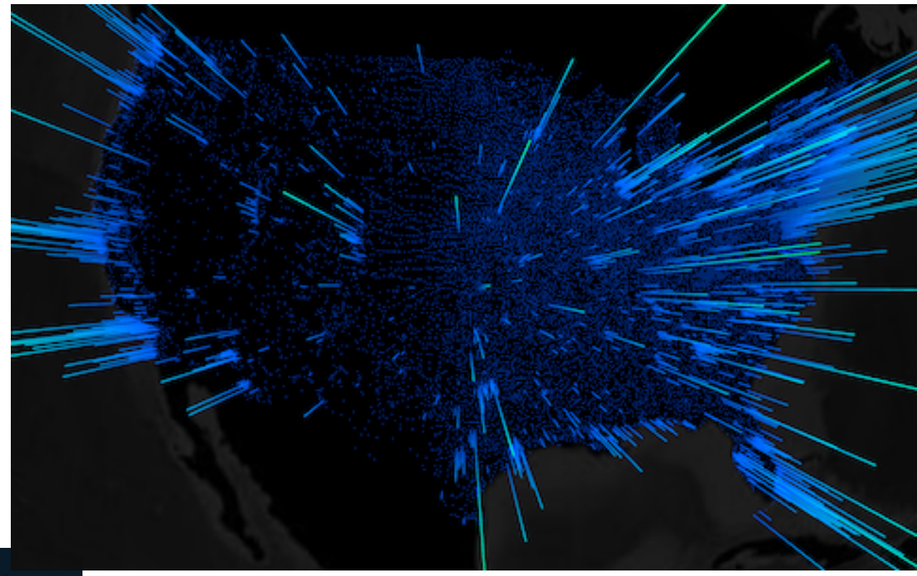
Jason Sundram
Senior Data Scientist
PayPal | Data Science
@jsundram

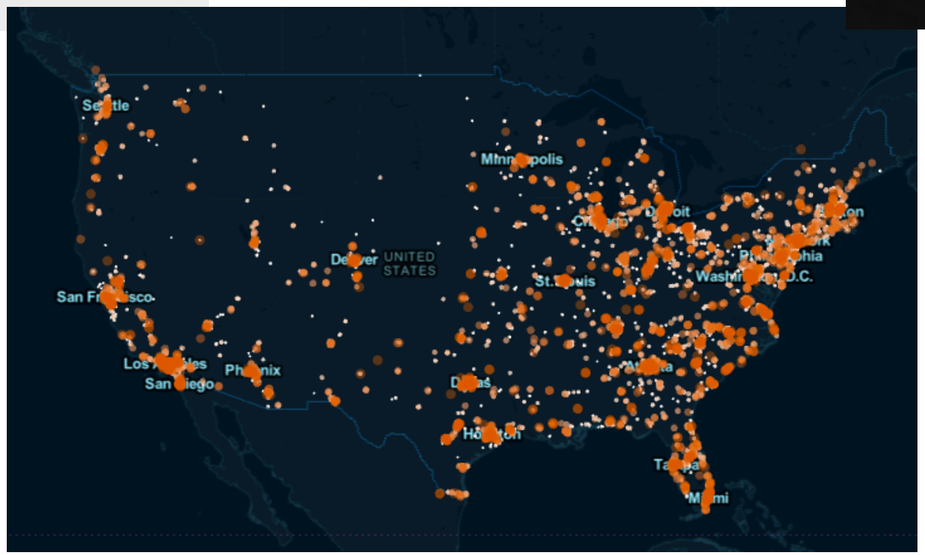
From text:

```
wherewolf:search_logs $  head app01.nimo.search.log
10.220.89.77 -  -  [09/Jul/2011:06:25:13 +0000] "GET /nimo/search?radius=25&mode=lp&sort=distance&mobile=true&uuid=d64f5e4cd405a6f8c6565cc77
33b322b5e259214&lat=32.85509866&lng=-96.67660158&query=Auto%20zone%20&pcount=20&ppage=0&lcount=1&lpage=0 HTTP/1.1" 200 1732 "-" "Where/70644
 CFNetwork/485.12.30 Darwin/10.4.0" 237
82.103.128.63 -  -  [09/Jul/2011:06:25:19 +0000] "GET /nimo/search HTTP/1.1" 200 36 "-" "Pingdom.com_bot_version_1.4_(http://www.pingdom.com
/)" 4
10.220.89.77 -  -  [09/Jul/2011:06:25:19 +0000] "GET /nimo/search?radius=25&mode=lp&sort=distance&mobile=true&uuid=c030c9463fe0519e0edcbc88e
118330aa4176924&lat=43.03044589502692&lng=-87.91138180811002&query=Moest&pcount=20&ppage=0&lcount=1&lpage=0 HTTP/1.1" 200 136 "-" "Where/706
44 CFNetwork/485.13.9 Darwin/11.0.0" 101
10.79.94.203 -  -  [09/Jul/2011:06:25:19 +0000] "GET /nimo/search?radius=25&mode=lp&sort=distance&mobile=true&lat=33.49010272170366&lng=-111
.987054916501&query=Restaurants&pcount=20&ppage=0&lcount=1&lpage=0 HTTP/1.1" 200 2054 "-" "Where/70641 CFNetwork/485.13.9 Darwin/11.0.0" 588
10.79.94.203 -  -  [09/Jul/2011:06:25:22 +0000] "GET /nimo/search?query=india&pcount=15&ppage=0&lcount=2&lpage=0&uuid=androidA1000017E5EA39&
mode=lp&sort=distance&mobile=true&lat=39.9225255&lng=-105.07795615999999&radius=20 HTTP/1.1" 200 1692 "-" "Mozilla/5.0 (iPhone; U; CPU like
Mac OS X; en) AppleWebKit/420+ (KHTML, like Gecko) Version/3.0 Mobile/1A543a Safari/419.3" 213
10.79.94.203 -  -  [09/Jul/2011:06:25:28 +0000] "GET /nimo/search?radius=25&mode=lp&sort=distance&mobile=true&uuid=f567253ba6a0c7f55afa9540e
86851cd6f31f052&lat=-25.77646365437229&lng=28.36891443947248&query=Shopping&pcount=20&ppage=0&lcount=1&lpage=0 HTTP/1.1" 200 136 "-" "Where/
70643 CFNetwork/485.13.9 Darwin/11.0.0" 333
10.79.94.203 -  -  [09/Jul/2011:06:25:31 +0000] "GET /nimo/search?query=starbucks&pcount=15&ppage=0&lcount=2&lpage=0&uuid=androidA1000017E5E
A39&mode=lp&sort=distance&mobile=true&lat=39.9225255&lng=-105.07795615999999&radius=20 HTTP/1.1" 200 1331 "-" "Mozilla/5.0 (iPhone; U; CPU l
ike Mac OS X; en) AppleWebKit/420+ (KHTML, like Gecko) Version/3.0 Mobile/1A543a Safari/419.3" 176
10.79.94.203 -  -  [09/Jul/2011:06:25:32 +0000] "GET /nimo/search?query=White+Castle&pcount=15&ppage=0&lcount=2&lpage=0&uuid=androidA1000017
A7CACC&mode=lp&sort=distance&mobile=true&lat=39.76734280586243&lng=-86.18511021137238&radius=20 HTTP/1.1" 200 1442 "-" "Mozilla/5.0 (iPhone;
 U; CPU like Mac OS X; en) AppleWebKit/420+ (KHTML, like Gecko) Version/3.0 Mobile/1A543a Safari/419.3" 166
10.79.94.203 -  -  [09/Jul/2011:06:25:34 +0000] "GET /nimo/search?radius=25&mode=lp&sort=distance&mobile=true&uuid=58ee2a7b430d7b86d4573edff
d828dcefd13c3d2&lat=38.90169298333333&lng=-121.06642865&query=Louie%27s&pcount=20&ppage=0&lcount=1&lpage=0 HTTP/1.1" 200 345 "-" "Where/7064
4 CFNetwork/485.13.9 Darwin/11.0.0" 105
10.220.89.77 -  -  [09/Jul/2011:06:25:36 +0000] "GET /nimo/search?radius=25&mode=lp&sort=distance&mobile=true&uuid=d64f5e4cd405a6f8c6565cc77
33b322b5e259214&lat=32.85437629561206&lng=-96.67642224589054&query=Auto%20zone%20&pcount=20&ppage=0&lcount=1&lpage=0 HTTP/1.1" 200 1730 "-"
"Where/70644 CFNetwork/485.12.30 Darwin/10.4.0" 151
```

To maps:







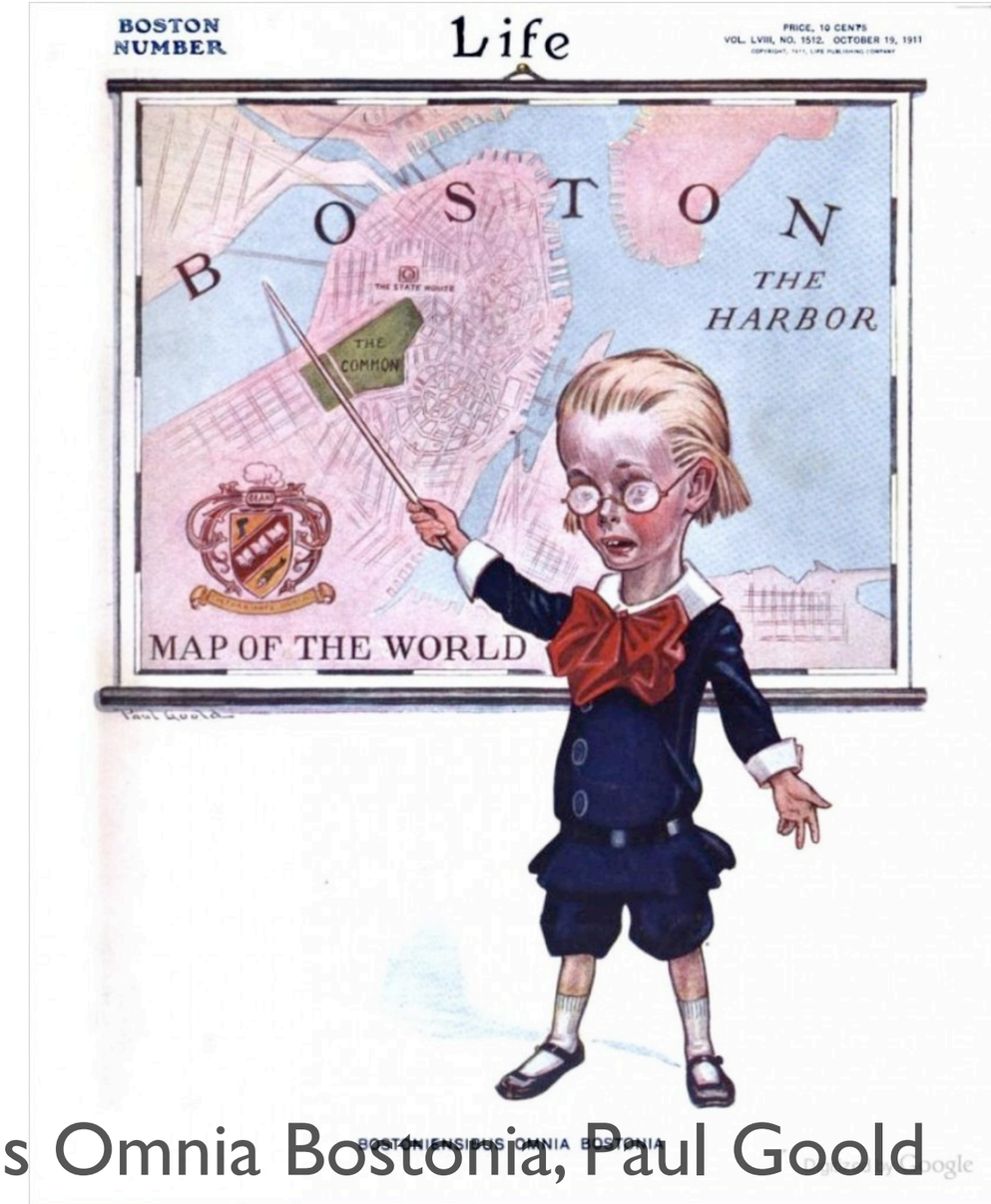
Who Am I?

- Math
- Music
- Data!
- Visualization

@jsundram
runningwithdata.com

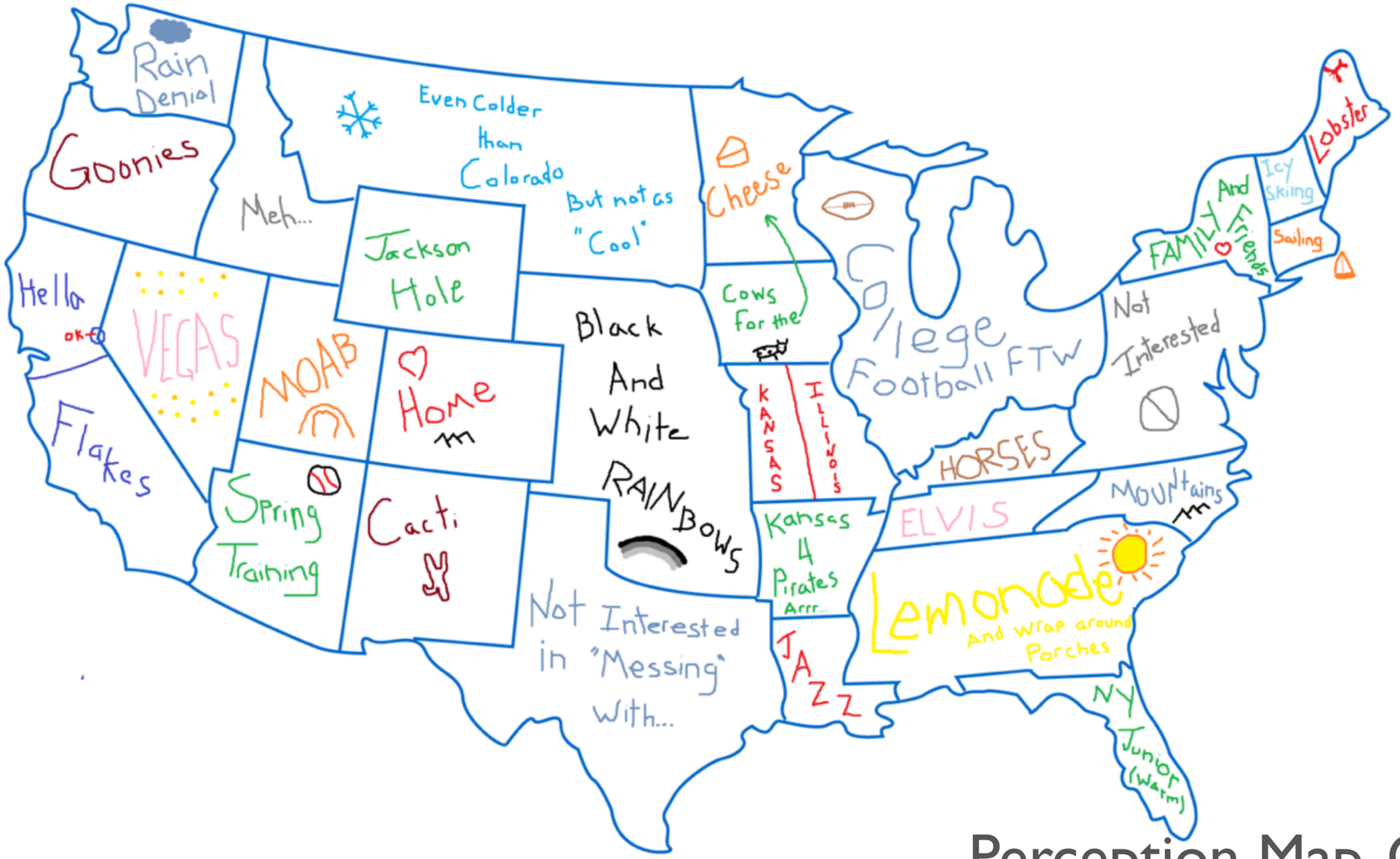
PayPal Data Science

- Acquired Where.com,
  - local recommendations
- Projects:
  - Demand Gen
  - Churn
  - MGM
  - Visualization



Bostoniensibus Omnia Bostonia, Paul Goold

What I won't cover



Perception Map, Ginatown.net

Three ways of looking

- Just plot it
- Spatial aggregation
- Heat map

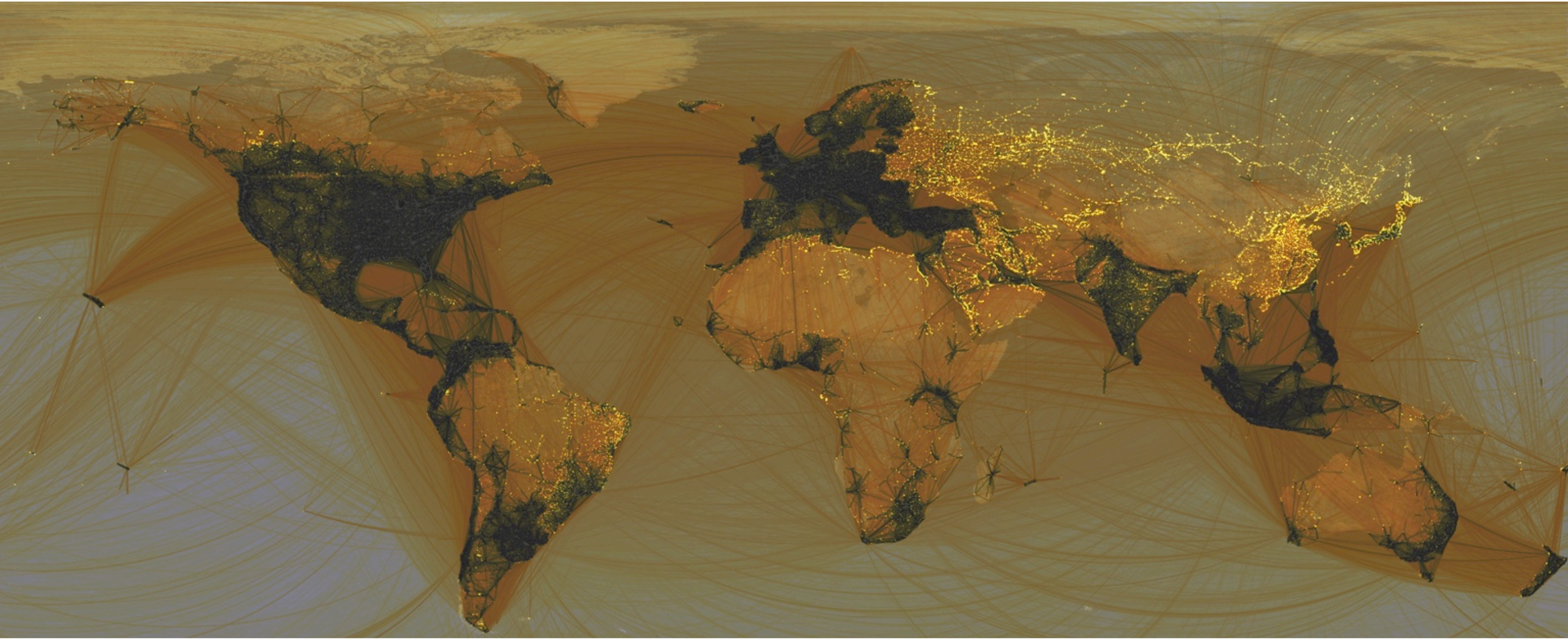
Just Plot It

- Benefits:
  - Easy
  - Pretty
- Drawbacks:
  - Reference points?

Just Plot It

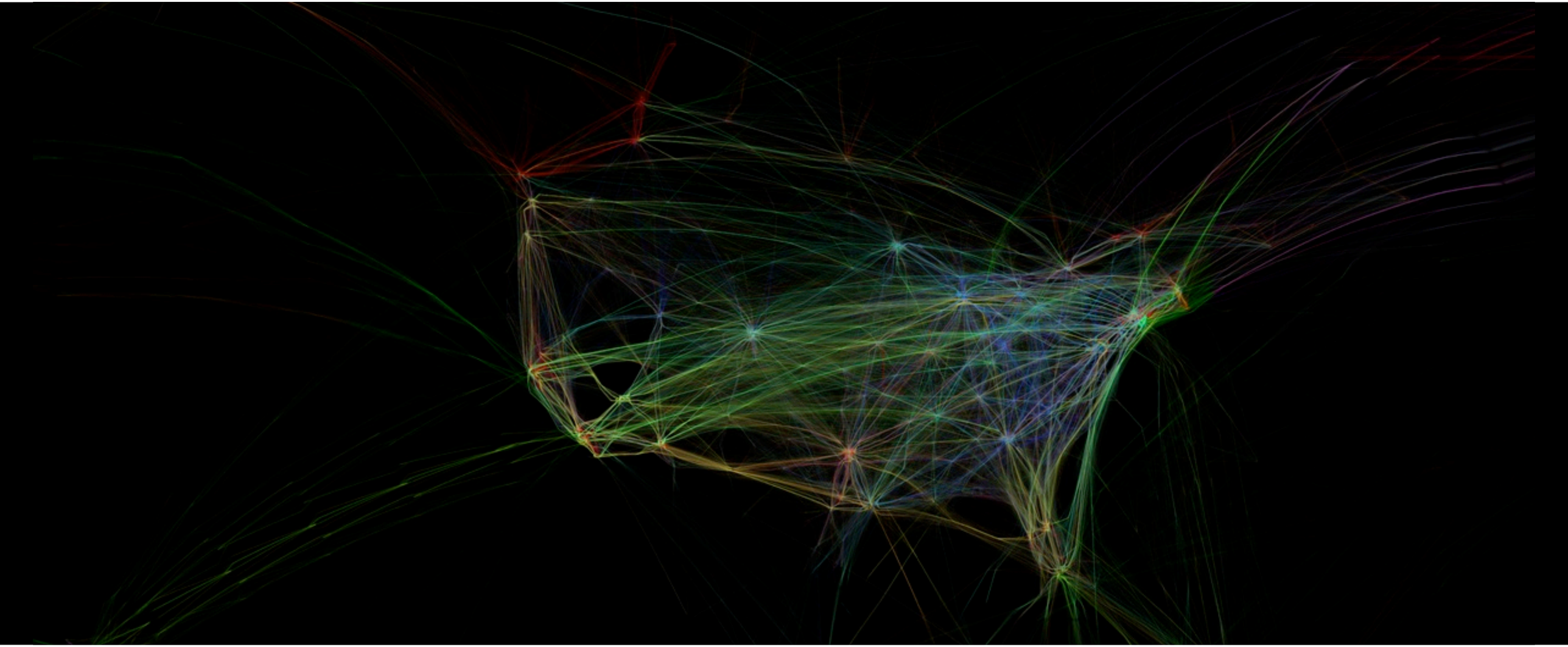
Visualizing Friendships, Paul Butler

Just Plot It



The UnFacebook World, Ian Wojtowicz

Just Plot It



Flight Patterns, Aaron Koblin

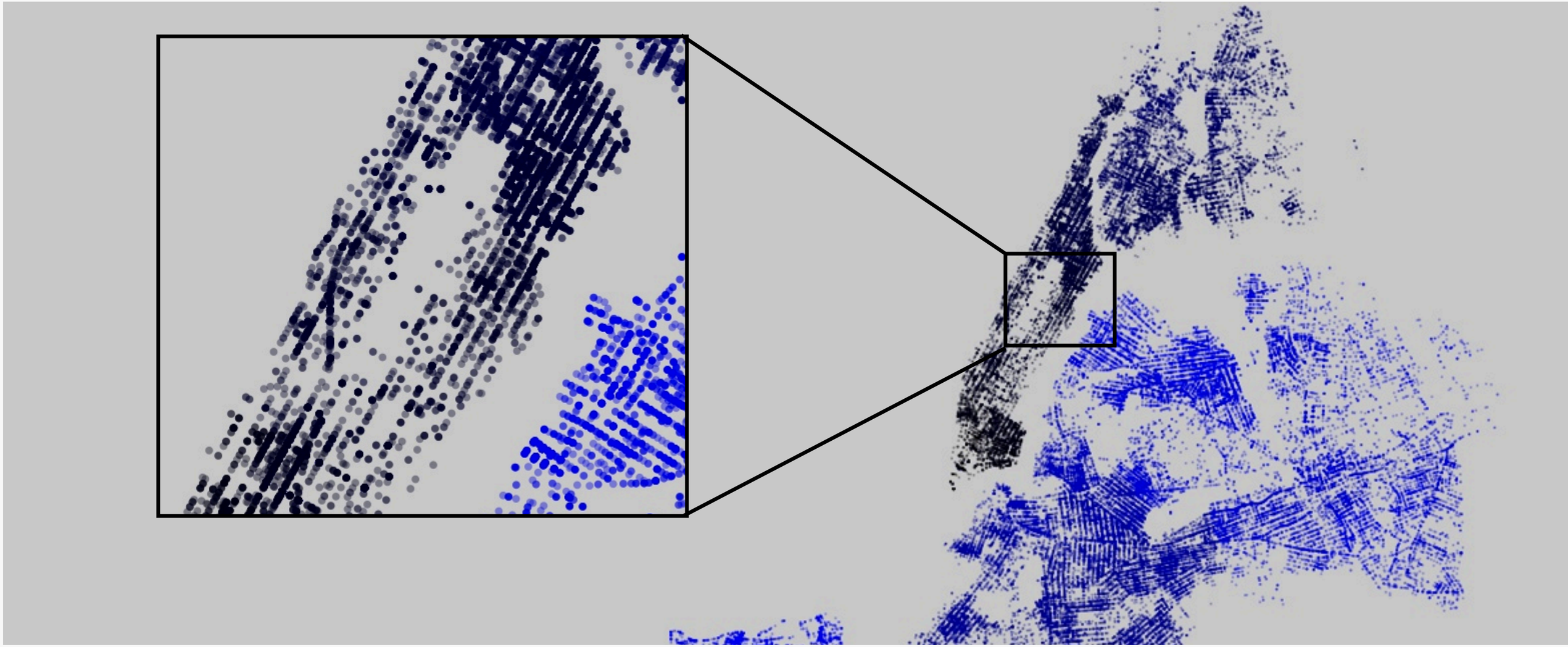
Just Plot It

Wispy Routes, Eric Fischer

Just Plot It

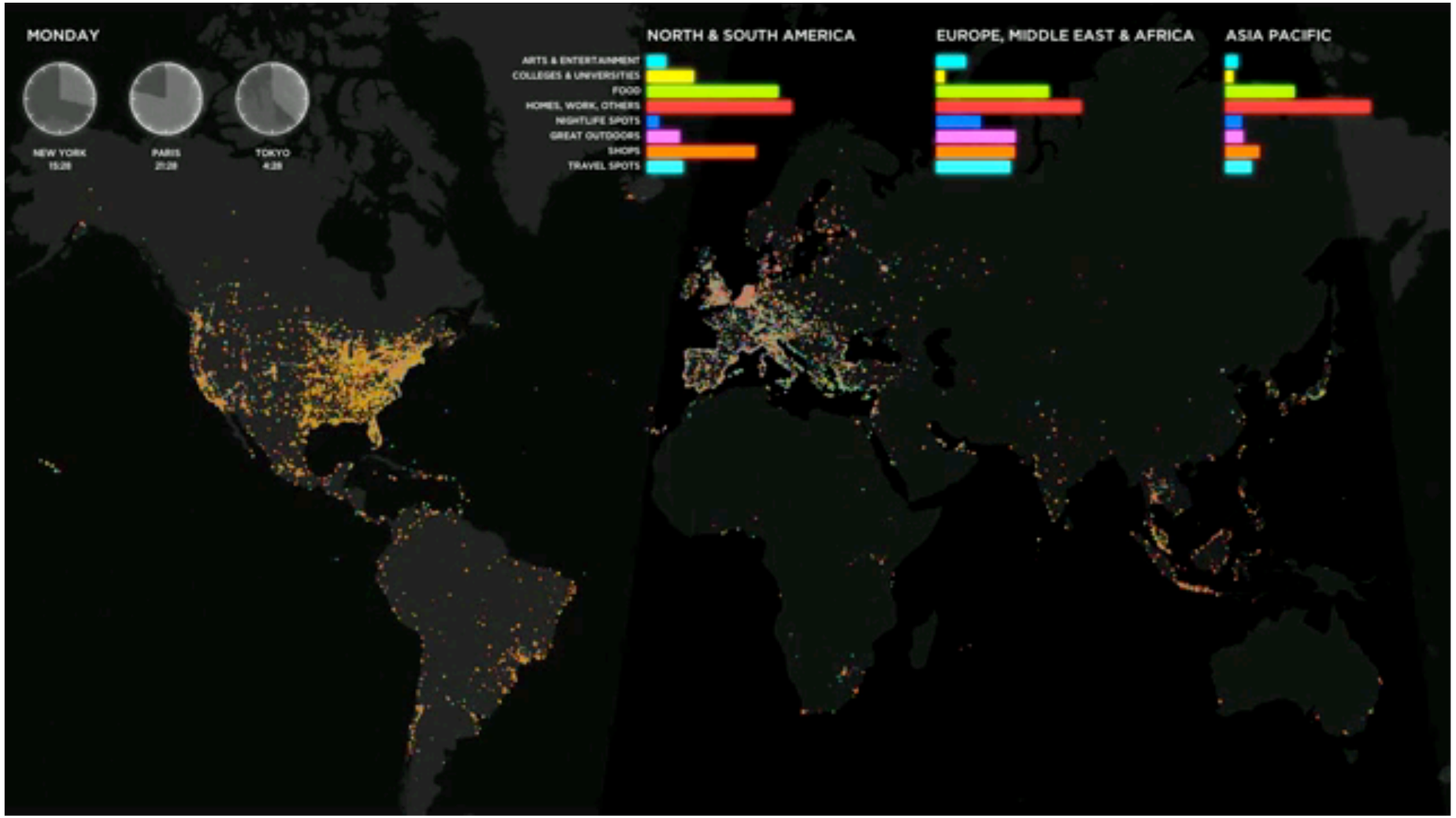
- Data Without Borders - NYC Data Dive
- NY Civil Liberties Union
  - 1,193,763 stops from 2010 alone

Just Plot It



Stop, Question, Frisk

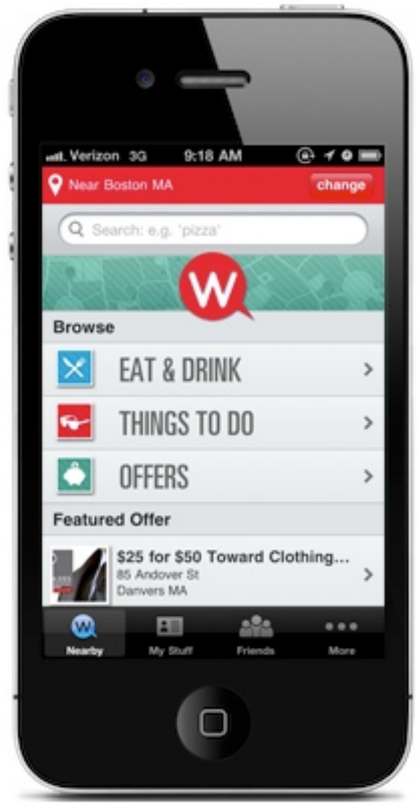
Just Plot It - Time



A Week of Check-Ins, Matthew Healy

Just Plot It

- Data:
  - Search logs - 6 million searches
  - Facebook profiles
- Motivation:
  - Showcase demographic & behavioral targeting



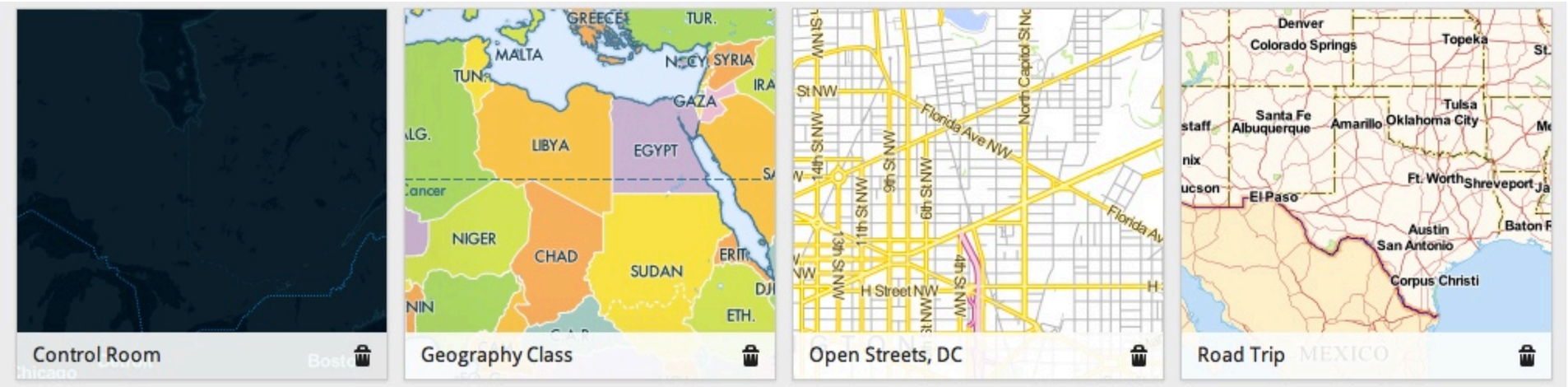
Where.com

Processing

- All-purpose drawing
- Java
- Easy to get started
- Addictive

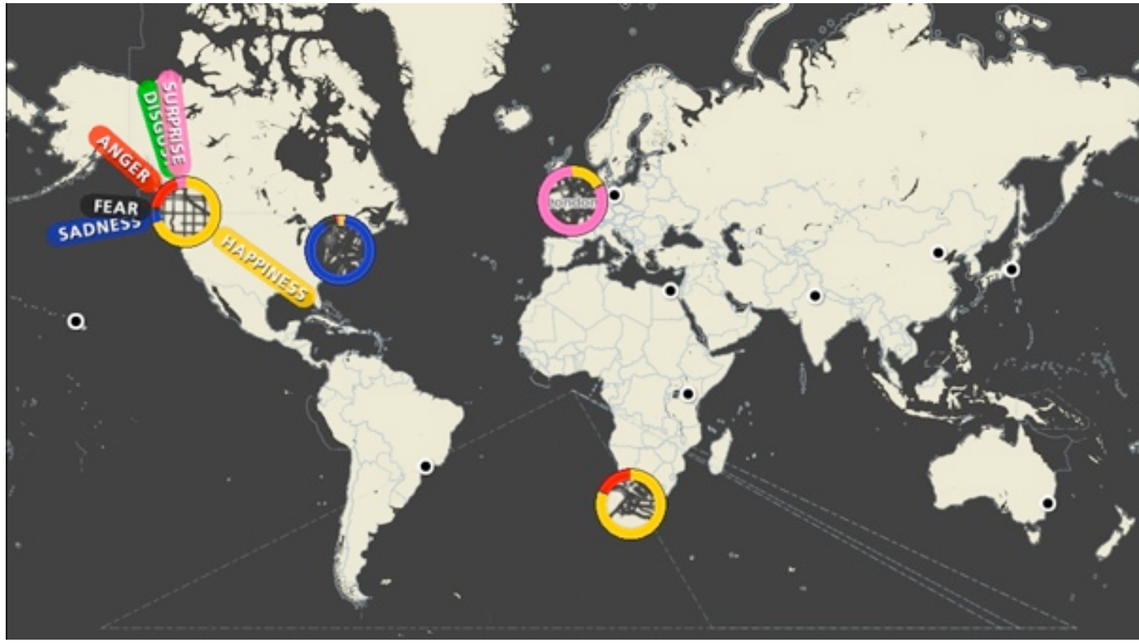
Tile Mill

- Design and edit maps
- Slippy
- Easy



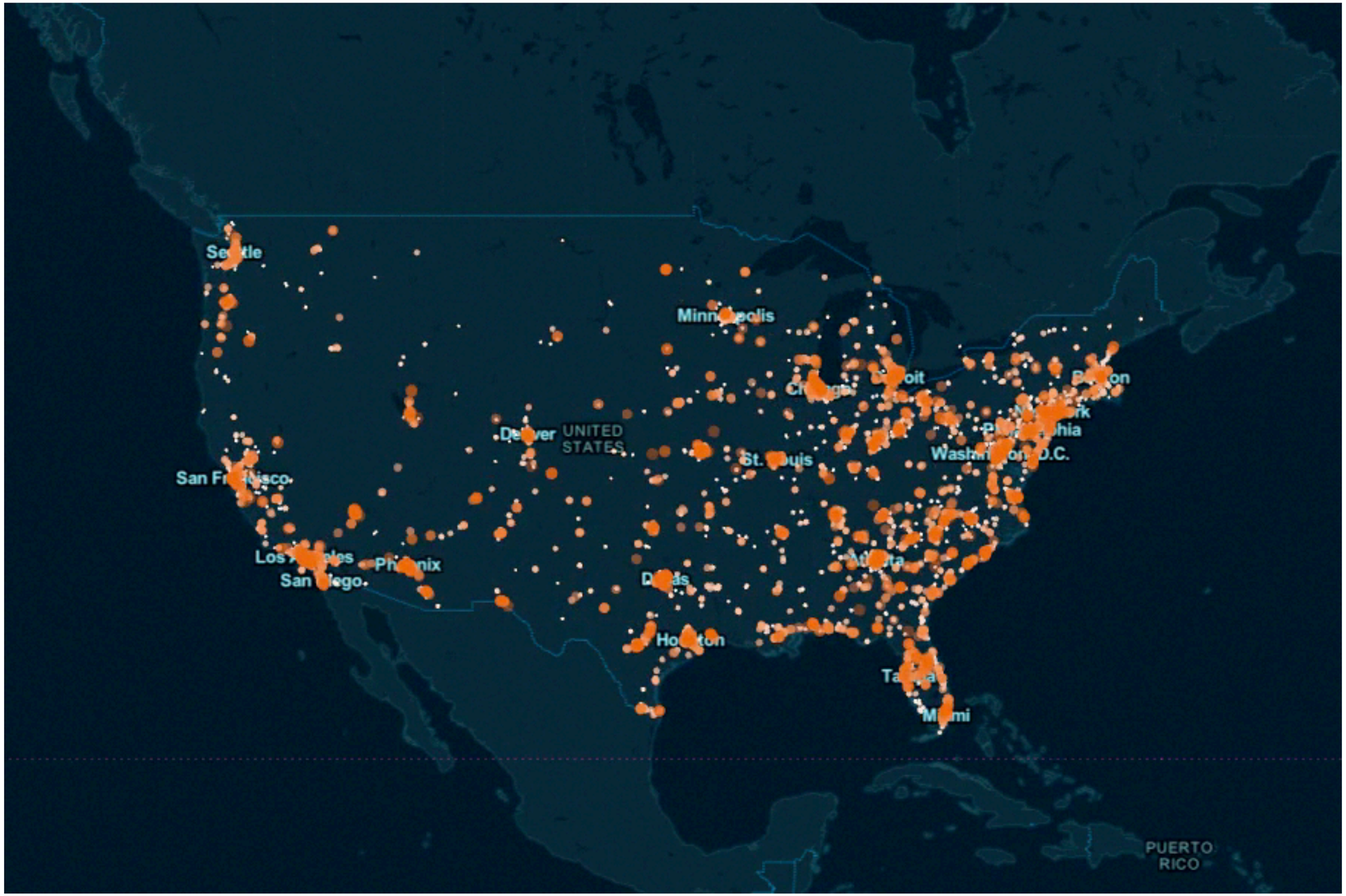
Unfolding

- Easily manipulate map tiles in Processing
- CloudMade, OpenStreetMaps, TileMill



Emography, Daniel Palmer

Putting it all together



Three ways of looking

1. Just plot it
2. Spatial aggregation
3. Heat map

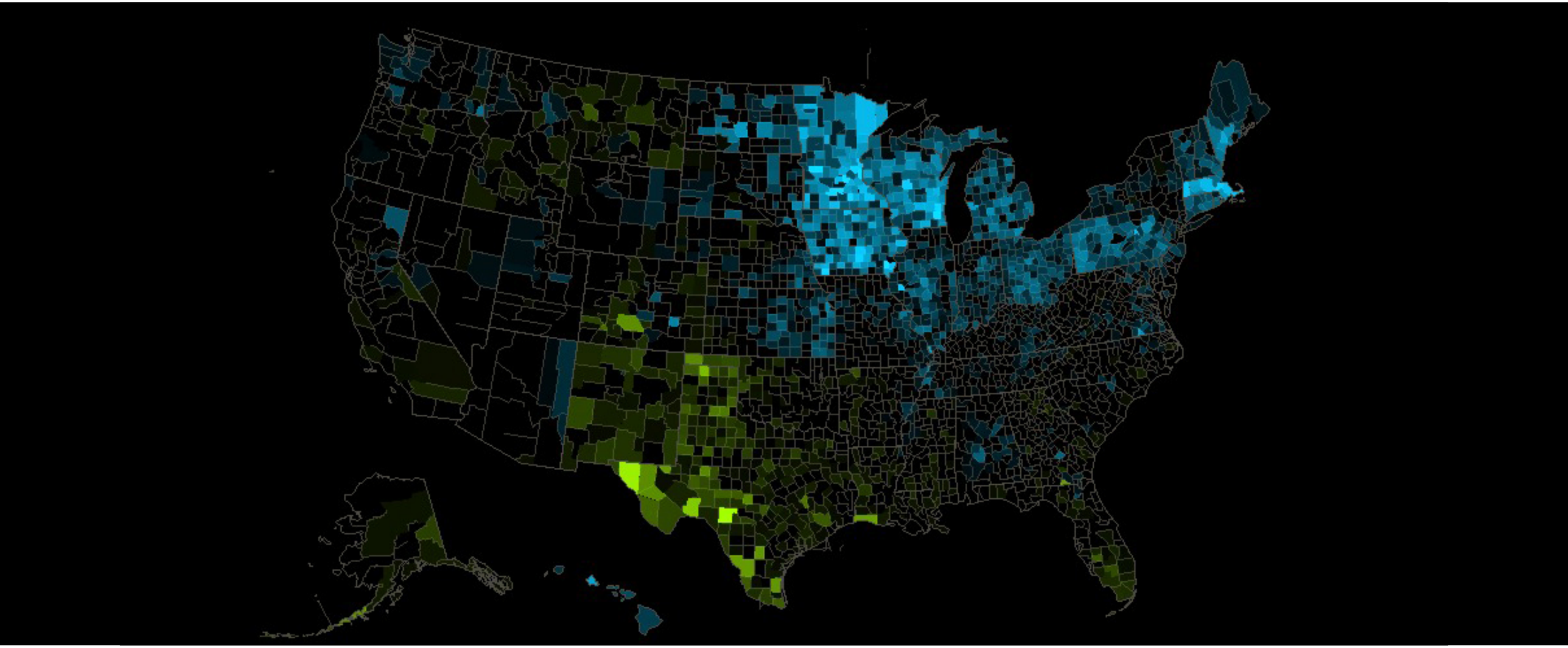
Spatial Aggregation

- Choropleth = Choro (Area/Region) + Pleth (quantity)
- Cartogram (Contiguous, Non-Contiguous, Dorling)

Spatial Aggregation

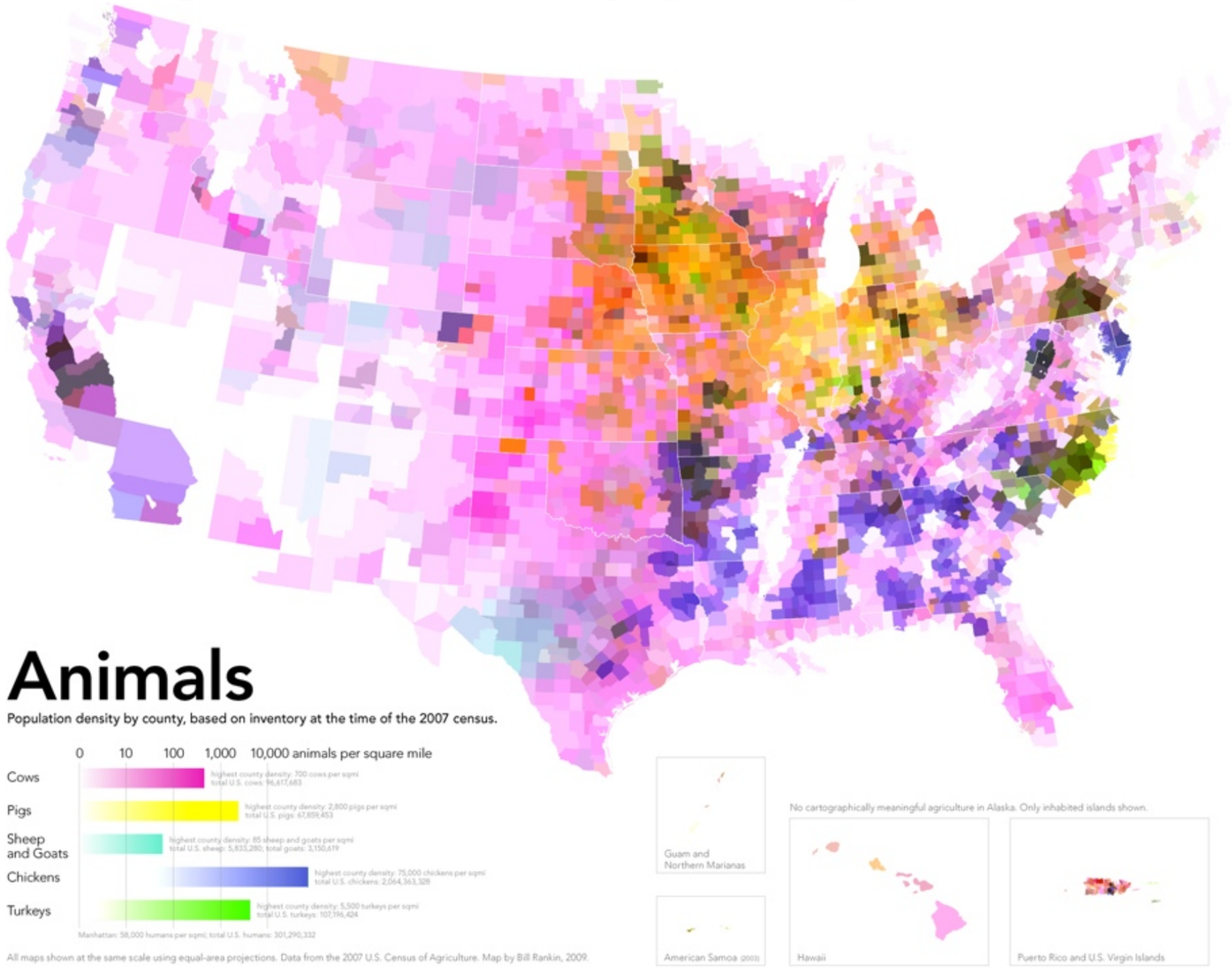
- Benefits:
  - Easy to see simple patterns
- Drawbacks:
  - Perceptual issues
  - Sensitive to color/class choices

Spatial Aggregation



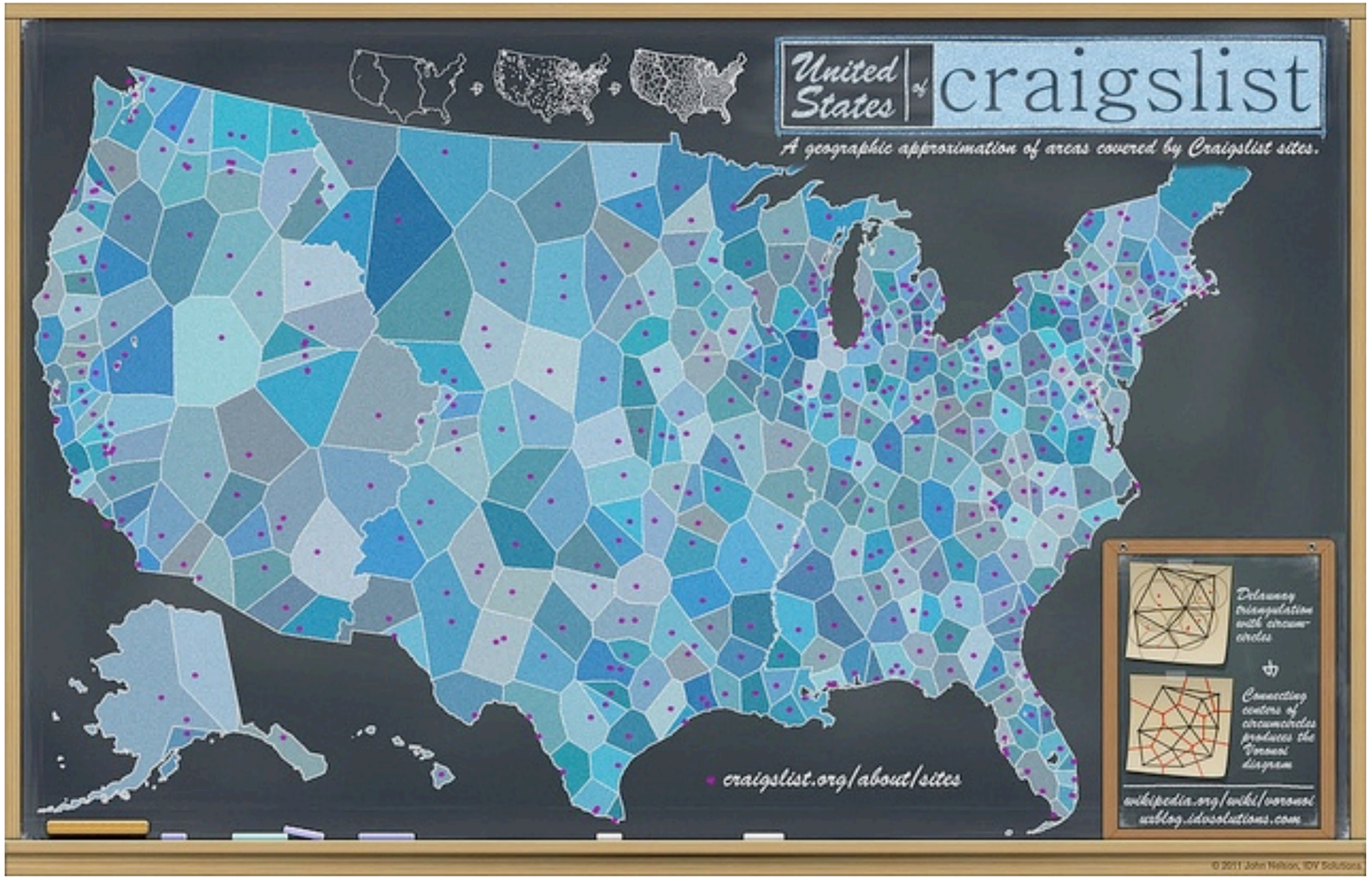
Uninsured (under 65) from Stats of the Union, Fathom

Spatial Aggregation



U.S. Agriculture, Bill Rankin

Spatial Aggregation



United States of Craigslist, John Nelson

Spatial Aggregation

- Data:
  - One day of geo-tagged barcode scans
- Motivation
  - Exploratory

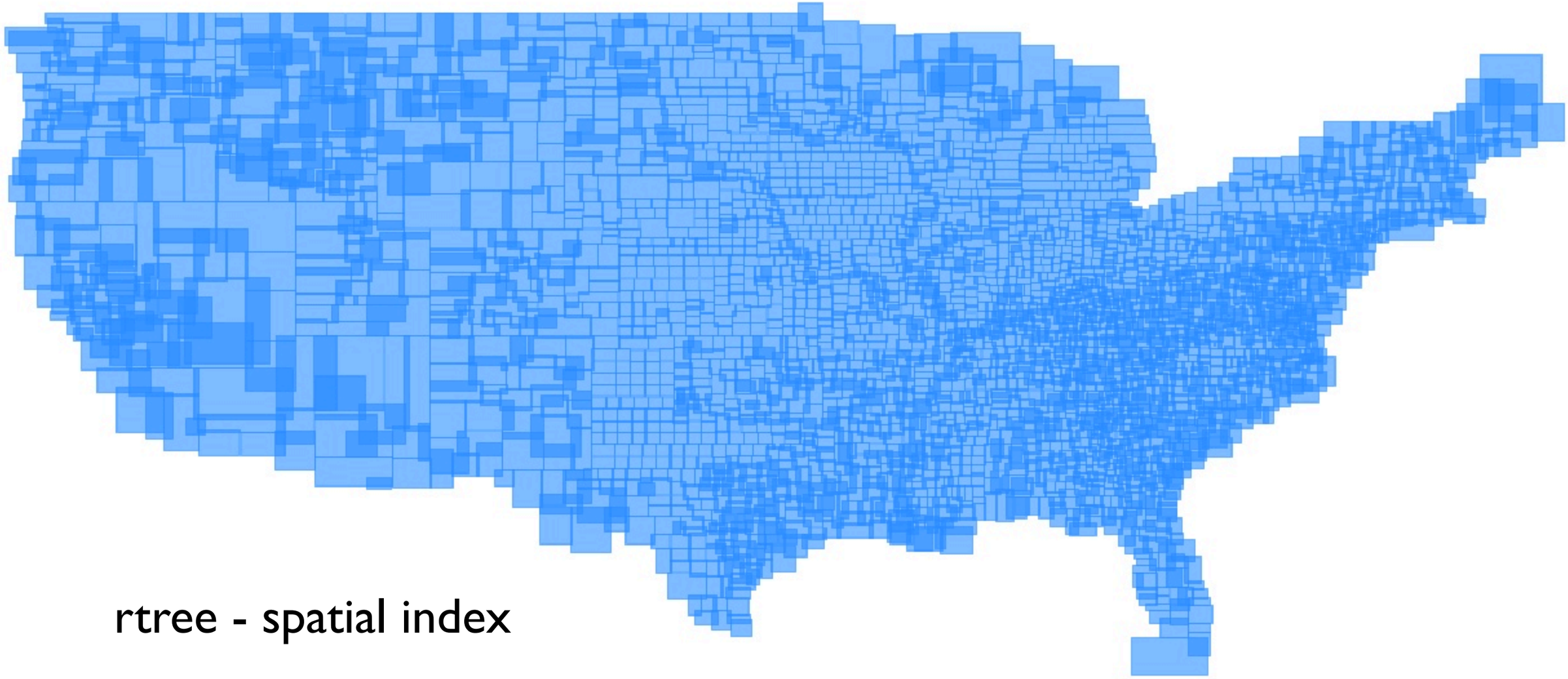
RedLaser.com

Spatial Aggregation

- Shapefile - points, polylines, polygons
  - http://www.census.gov/geo/www/tiger/
- pyshp - python shapefile library

California County Boundaries, geocommons.com

Spatial Aggregation



rtree - spatial index

Visualizing County Bounding Boxes

Spatial Aggregation

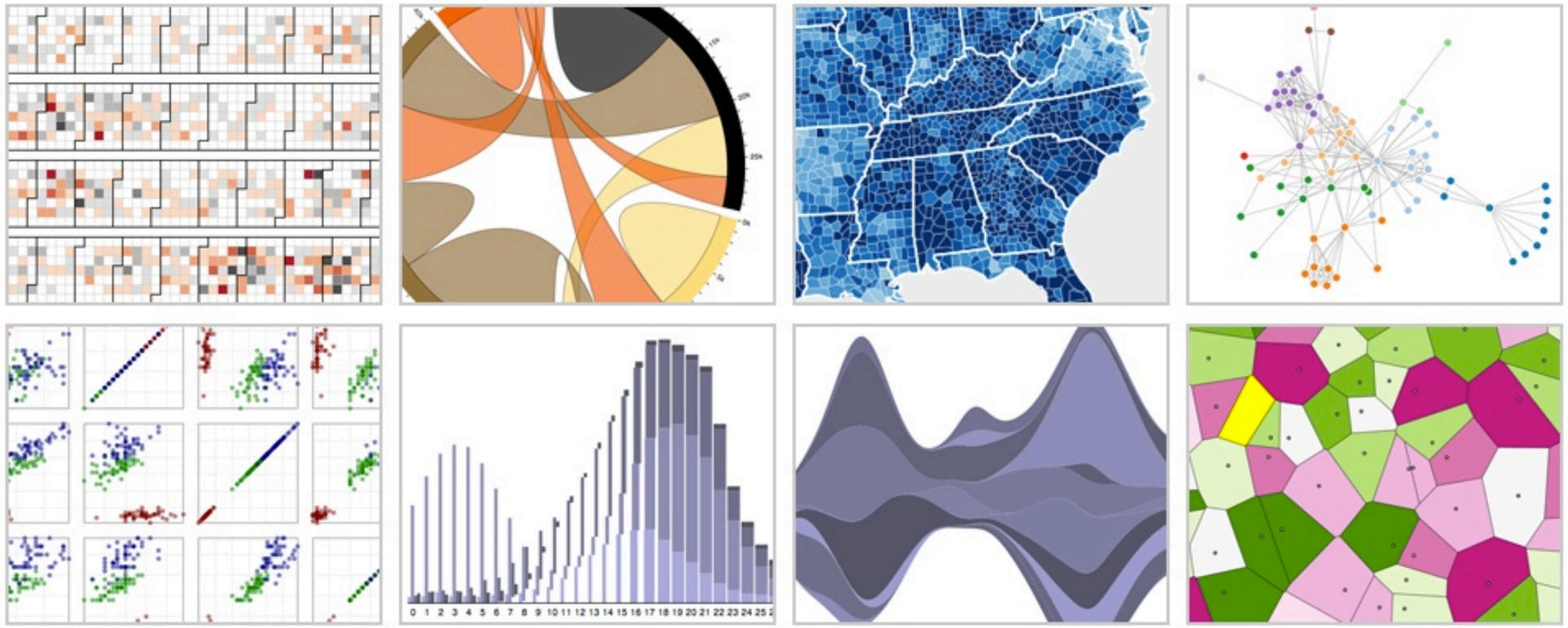
```python
import json
from collections import defaultdict
from itertools import izip
from rtree import index
import shapefile

def read_shapefile(shape_dir):
    sf = shapefile.Reader(shape_dir)
    shapes = {}
    for (s, sr) in izip(sf.shapes(), sf.shapeRecords()):
        shapes[int(sr.record[3])] = s
    return shapes

def make_index(county_shapes):
    idx = index.Index()
    for (countyId, s) in county_shapes.iteritems():
        idx.insert(countyId, s.bbox)
    return idx

def query_index(idx, shapes, lat, lng, e=.0000001):
    for hit in idx.intersection((lng - e, lat - e, lng + e, lat + e)):
        if hit_test(lng, lat, shapes[hit].points):
            return hit

def hit_test(x, y, polygon):
    inside = False
    x1, y1 = polygon[0]
    for i in range(len(polygon)) + [0]:
        x2, y2 = polygon[i]
        if min(y1, y2) < y <= max(y1, y2) and x <= max(x1, x2):
            xints = (y-y1) * (x2-x1) / (y2-y1) + x1
            if x1 == x2 or x <= xints:
                inside = not inside
        x1, y1 = x2, y2
    return inside

def process(your_data):
    shapes = read_shapefile('./us_county') # http://www2.census.gov/geo/tiger/TIGER2009/tl_2009_us_county.zip
    idx = make_index(shapes)

    summary = defaultdict(int)
    for (lat, lng) in your_data:
        countyId = query_index(idx, shapes, lat, lng)
        summary[countyId] += 1
    json.dump(summary, open('data.json', 'w') # scale this later
```

data.json


```
{"01001": 0.00203, "01003": 0.00558, "01005": 0.00244,
"01007": 0.00074, "01009": 0.00468, "01013": 0.00119,
"01015": 0.00329, "01017": 0.00117, "01019": 0.00123,
"01021": 0.00380, "01023": 0.00173, "01025": 0.00058,
"01027": 0.00057, "01029": 0.00020, "01031": 0.00202,
"01033": 0.00158, "01037": 0.00069, "01039": 0.00143,
"01041": 0.00093, "01043": 0.00020, "01045": 0.00187,
"01047": 0.00027, "01049": 0.00082, "01051": 0.00145,
"01053": 0.00162, "01055": 0.00182, "01057": 0.00220,
"01059": 0.00268, "01061": 0.00022, "01063": 0.00232,
"01065": 0.00121, "01067": 0.00329, "01069": 0.00291,
"01071": 0.00224, "01073": 0.00093, "01075": 0.00158,
"01077": 0.00756, "01079": 0.00207, "01081": 0.00250,
"01083": 0.00488, "01085": 0.00053, "01087": 0.00769,
"01089": 0.00533, "01091": 0.00190, "01093": 0.00900,
"01095": 0.00190, "01097": 0.00351, "01099": 0.00473,
"01101": 0.00359, "01103": 0.00110, "01105": 0.00330,
"01107": 0.00081, "01109": 0.00182, "01111": 0.00253,
"01113": 0.00060, "01115": 0.00407, "01117": 0.01513,
"01119": 0.00029, "01121": 0.00044, "01123": 0.00214,
"01125": 0.00496, "01127": 0.00283, "01129": 0.00228,
"01131": 0.00017, "01133": 0.00188, "02016": 0.00102,
"02020": 0.00002, "02050": 0.00006, "02068": 0.00164,
"02070": 0.00103, "02090": 0.00309, "02100": 0.04306,
"02122": 0.03153, "02150": 0.00250, "02170": 0.00238,
"02180": 0.00021, "02185": 0.00318, "02188": 0.00160,
"02230": 0.00000, "02261": 0.00073, "02290": 0.00251,
"04001": 0.00127, "04003": 0.00226, "04005": 0.00042,
"04007": 0.00035, "04009": 0.00124, "04012": 0.00976,
"04013": 0.00204, "04015": 0.00252, "04017": 0.00090,
"04019": 0.00288, "04021": 0.01781, "04023": 0.00084,
"04025": 0.00363, "04027": 0.00210, "05001": 0.00373,
"05003": 0.00215, "05005": 0.00224, "05007": 0.00546,
```

Spatial Aggregation

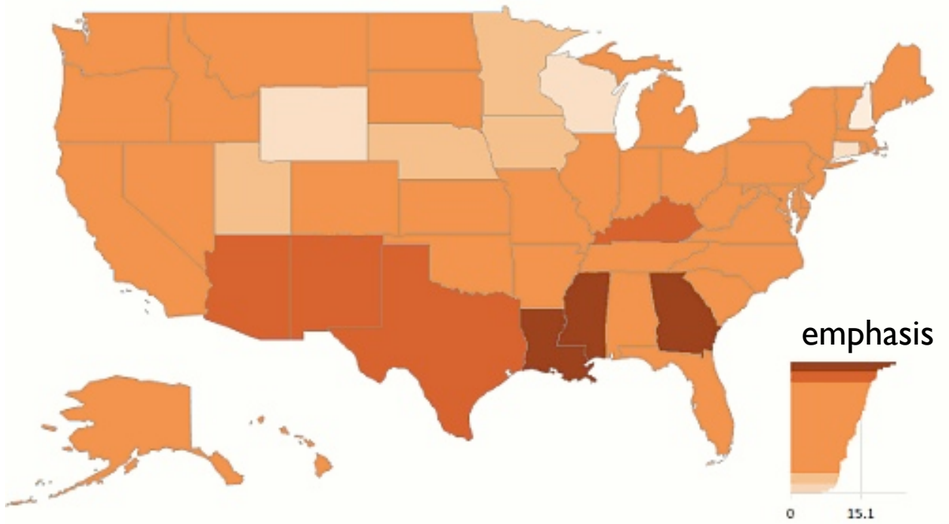
- d3.js - Data Driven Documents



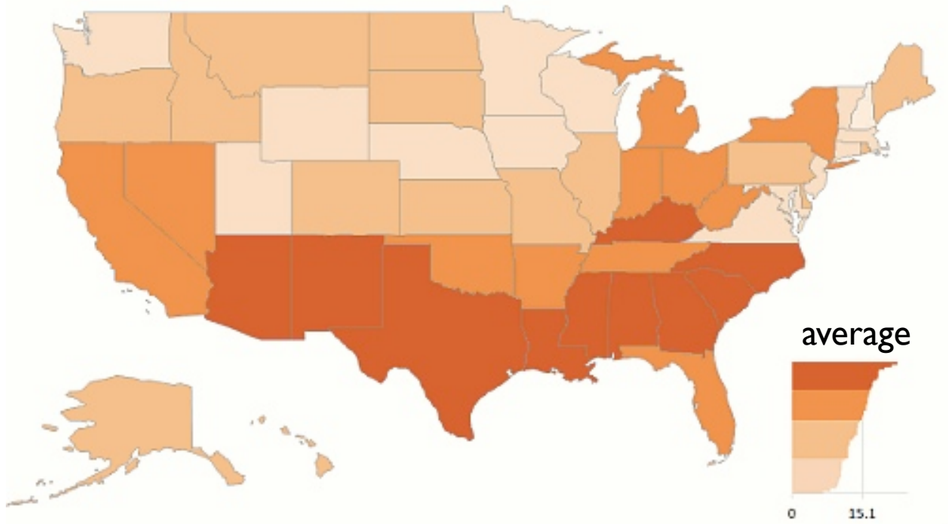
d3.js, Mike Bostock

Spatial Aggregation

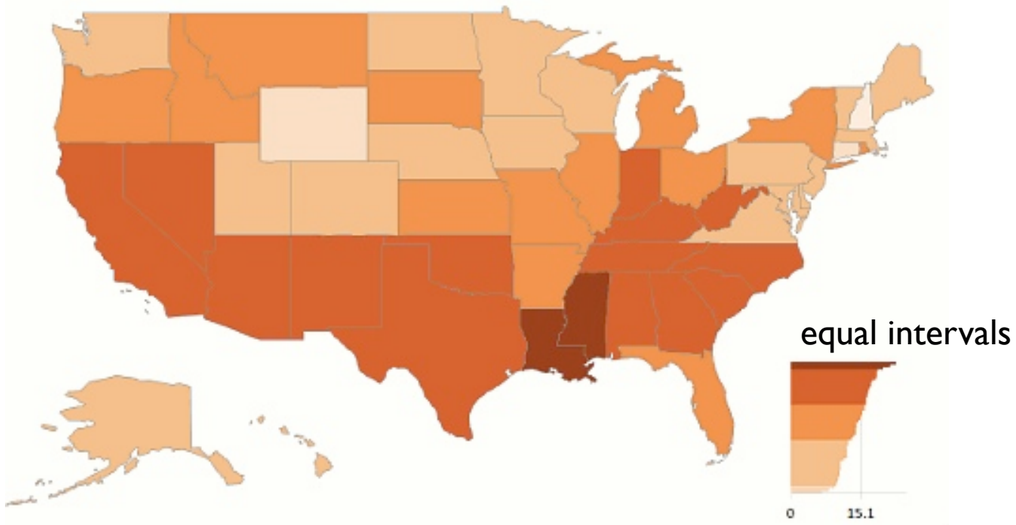
1



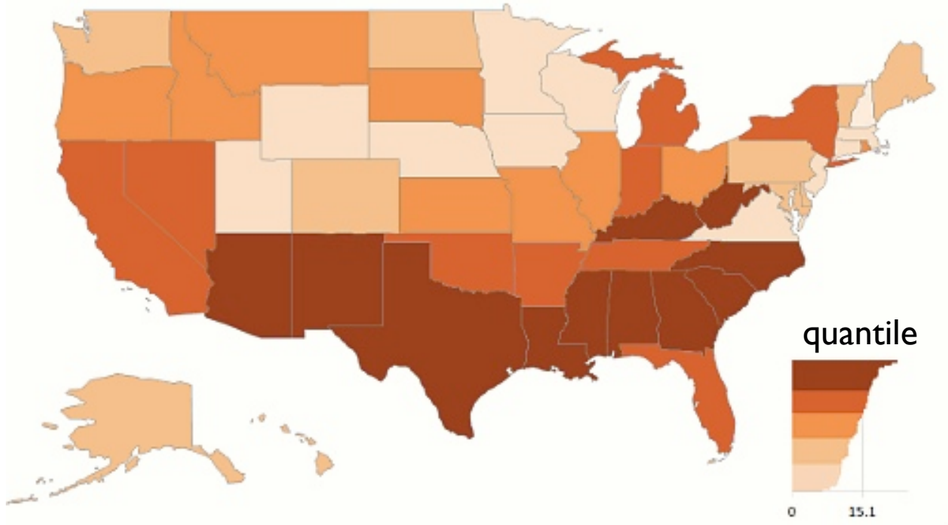
2



3

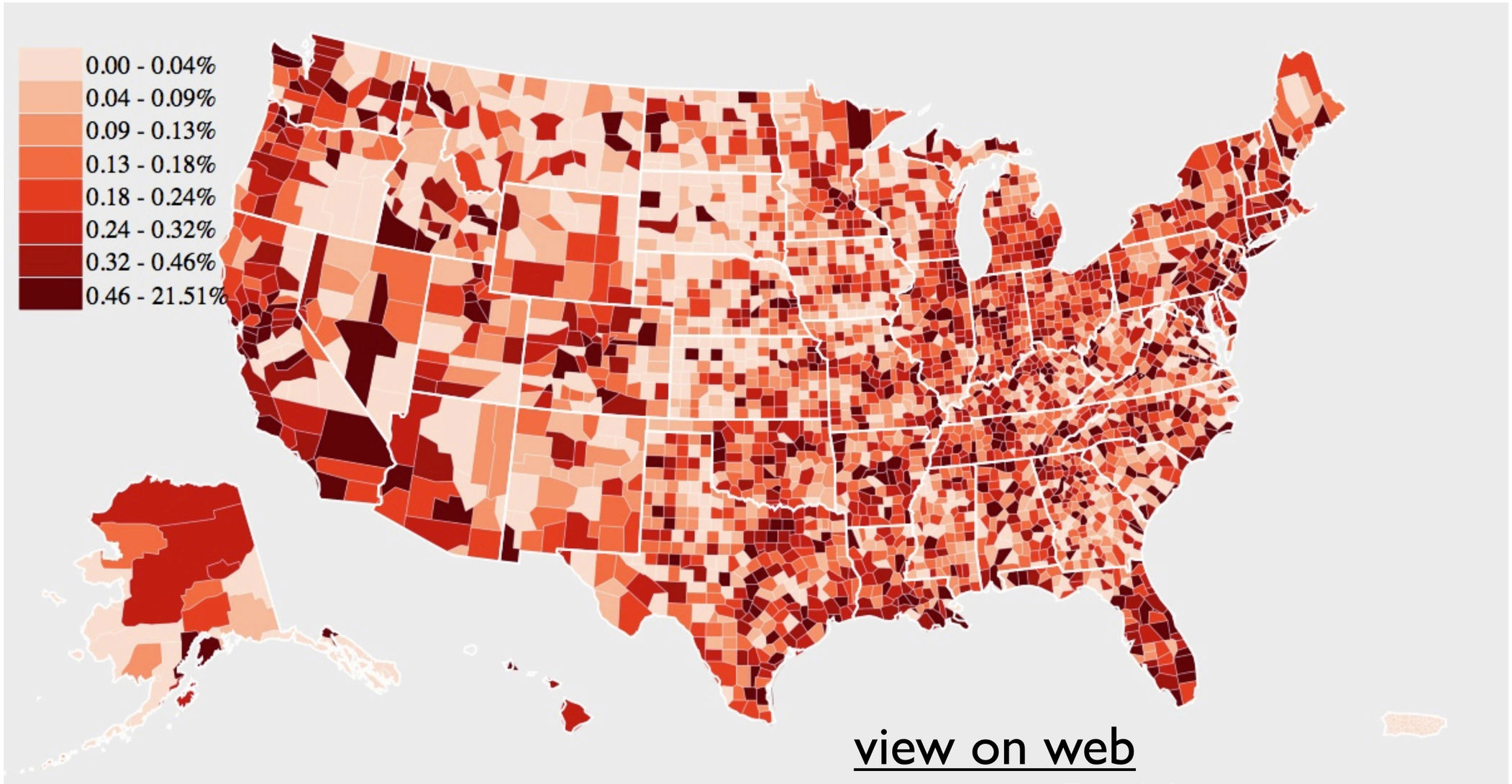


4



Poverty: The same data, the same map, different stories, ExcelCharts

Spatial Aggregation



view on web

Red Laser Scans per Capita

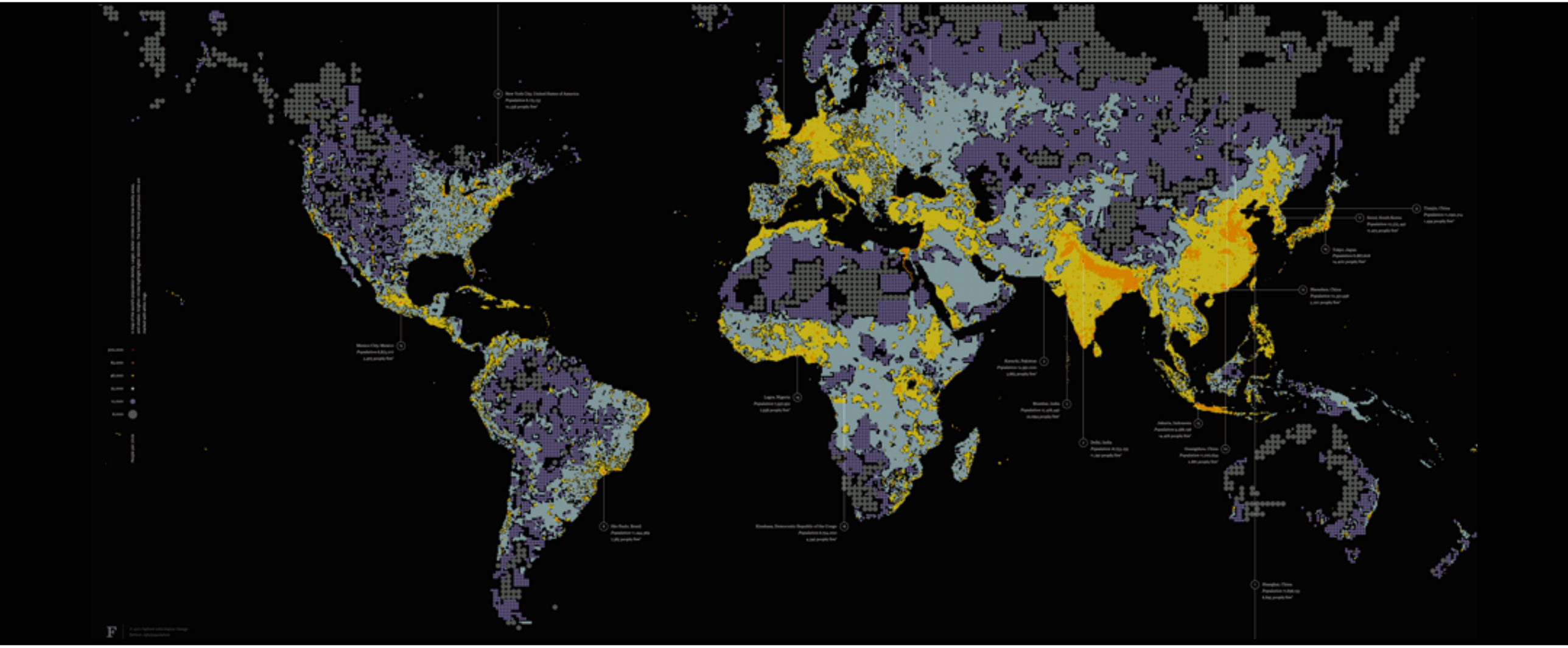
Three ways of looking

1. Just plot it
2. Spatial aggregation
3. Heat map

Heat Map

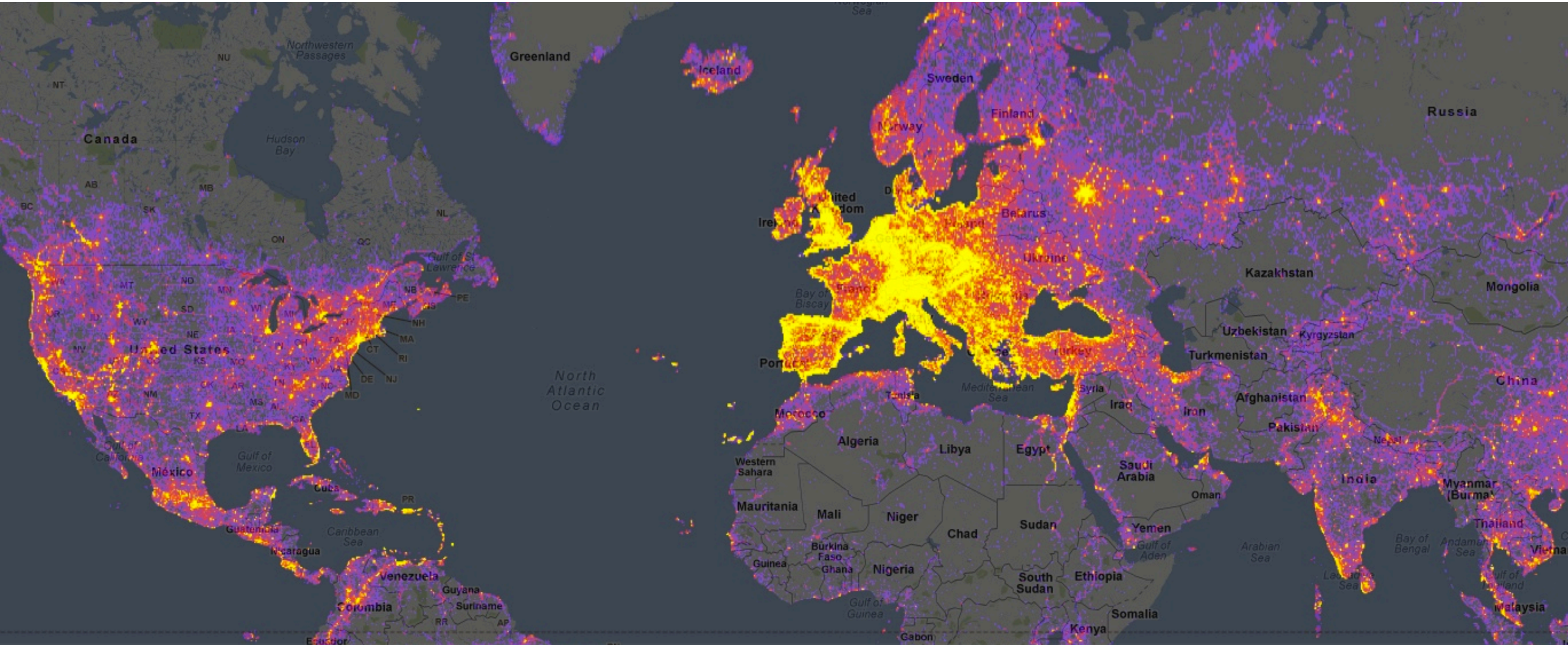
- Pros:
  - Data decides where you look
  - Better than “Just Plot It”?
- Cons:
  - Not a lot of insight

Heat Map



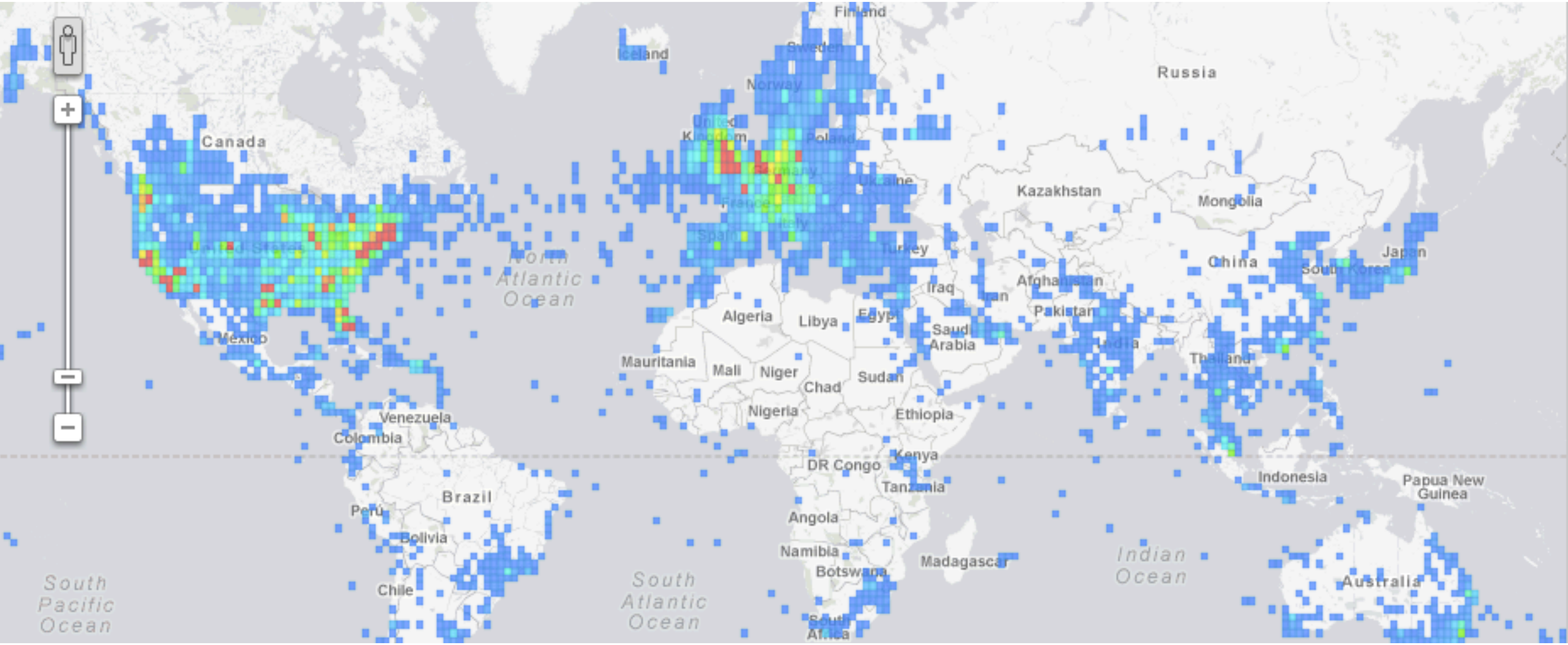
Dencity, Fathom

Heat Map



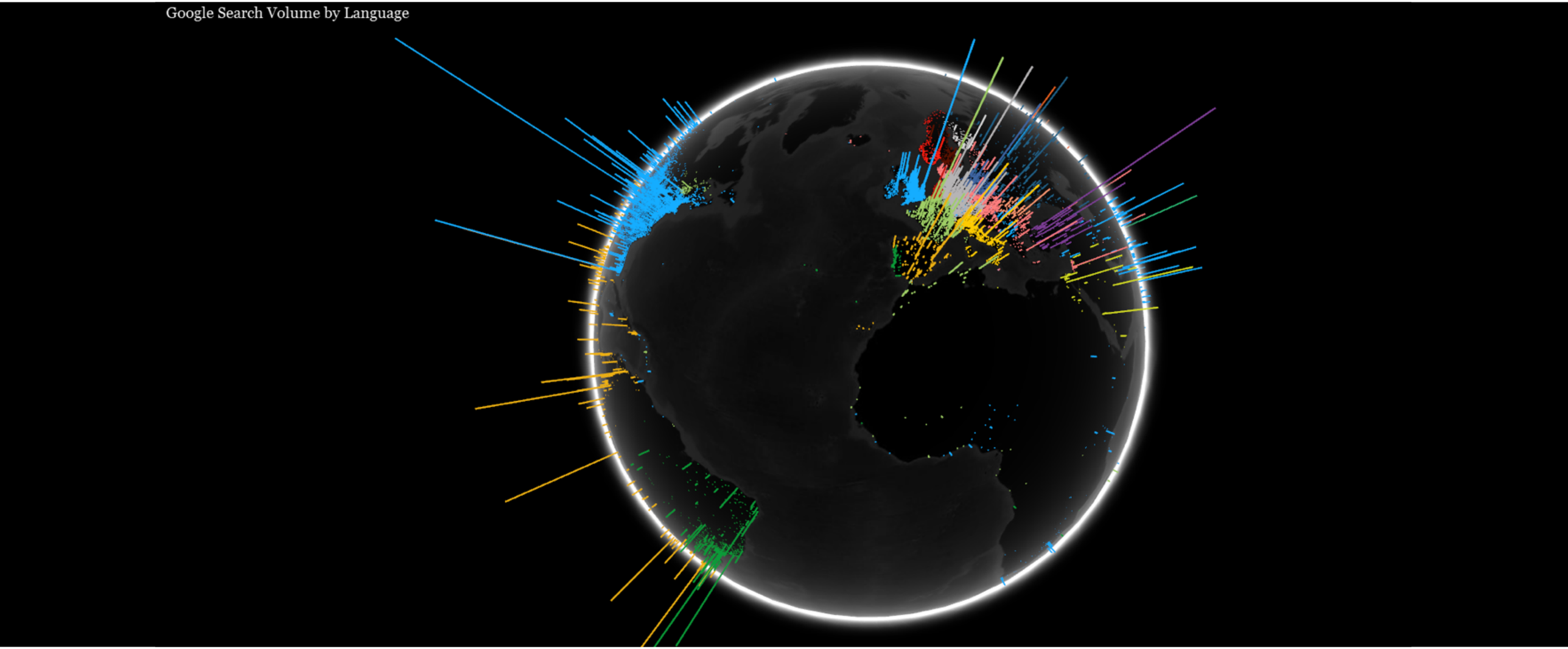
Sightsmap, Tanel Tammet

Heat Map



Participants by Region, OpenPaths.cc

Heat Map

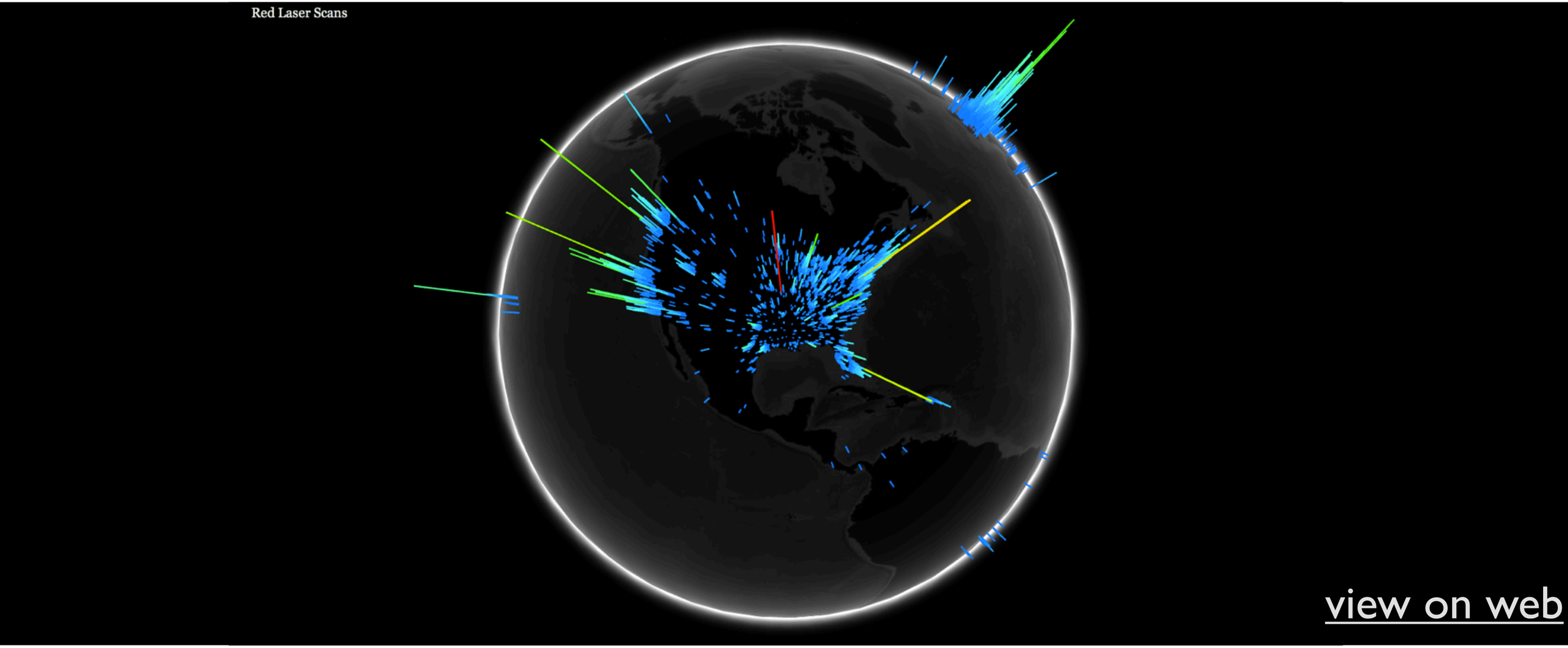


Search Volume by Language, Google

Heat Map

- WebGL Globe
  - Need less than 250,000 points
  - Cheat by binning (e.g. 72.12345→72.1)
  - Scale values

Heat Map



view on web

Red Laser Scans

Three ways of looking

1. Just plot it
2. Spatial aggregation
3. Heat map

Thank You

- Resources:
  - http://viz.runningwithdata.com/dataconf2012/
  - @jsundram

Questions?