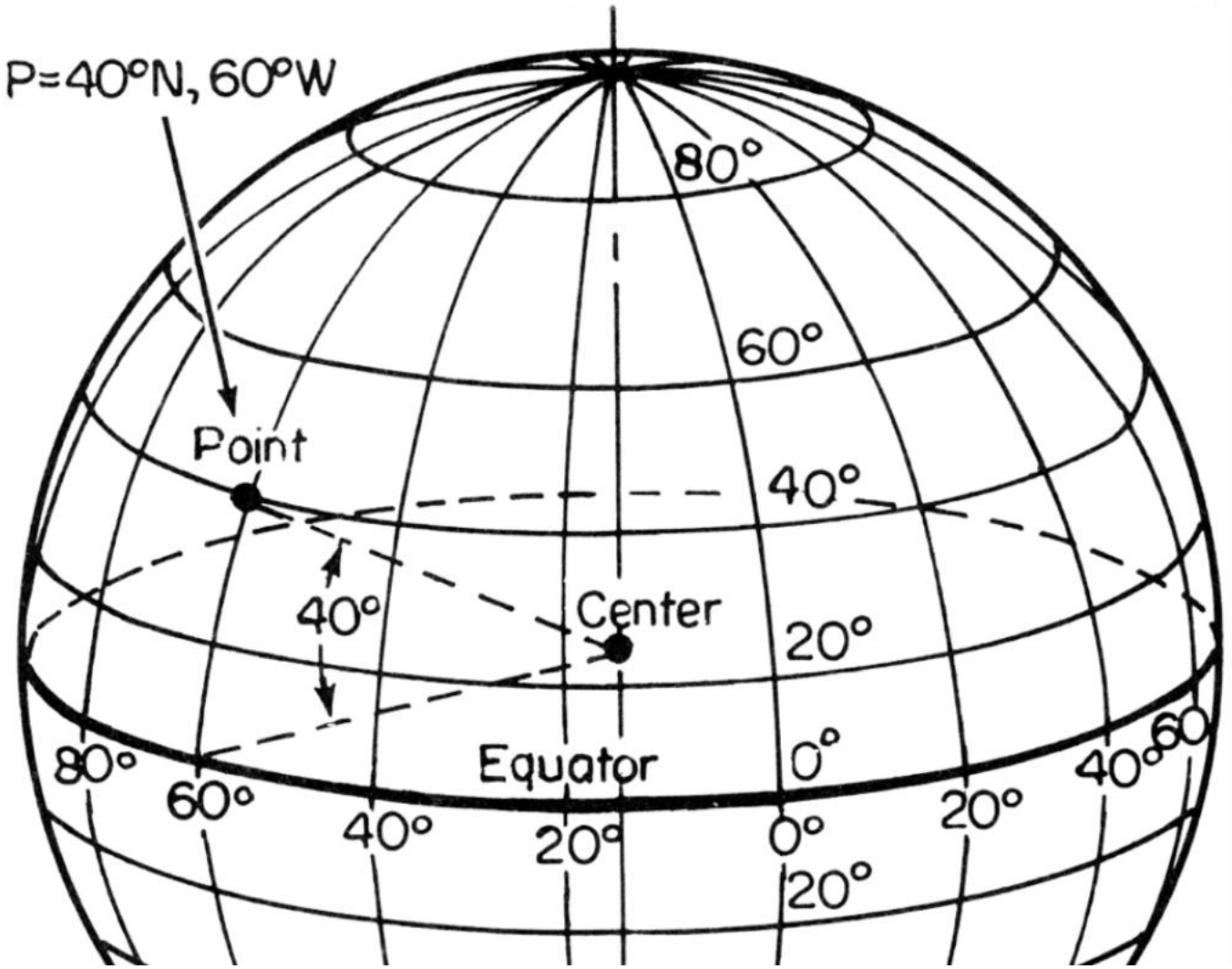
Geospatial visualization


Arctic Ocean
YT
NU
Canada
Iceland
Poland
Labrador Sea
SK
WA
QC
Italy
ID
MN
United States
PA
KY
Libya
North Atlantic Ocean
NM
TX
SC
LA
Mexico
Mali
Cuba
Puerto Rico
Nigeria
Nicaragua
Gabon
Colombia
Suriname
Ecuador
PE
AC
Brazil
BA
GO
South Pacific Ocean
Bolivia
South Atlantic Ocean
SP
RJ
Chile
Argentina

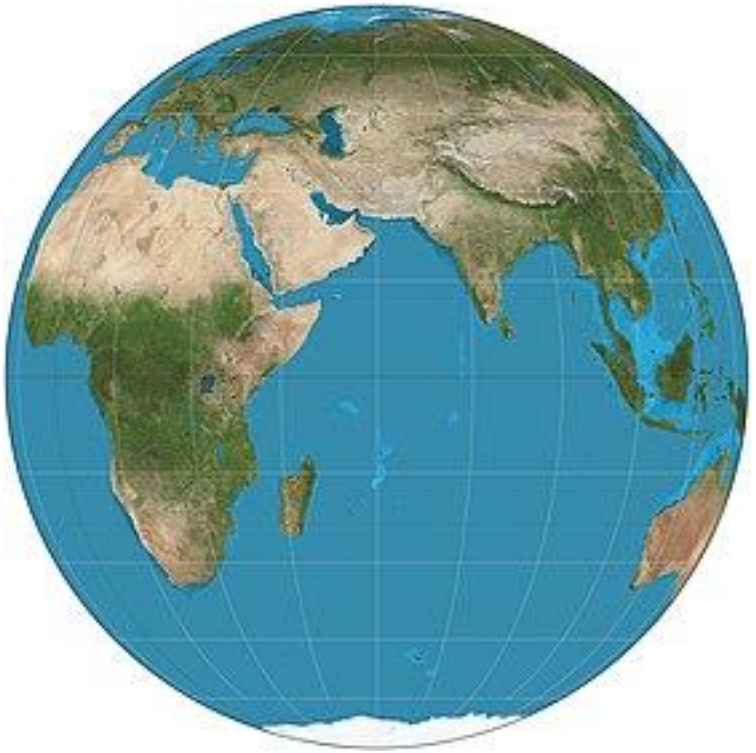
Considerations for geospatial visualization

- Projection
- Context
- Distortion
- Interpolation

Latitude and longitude

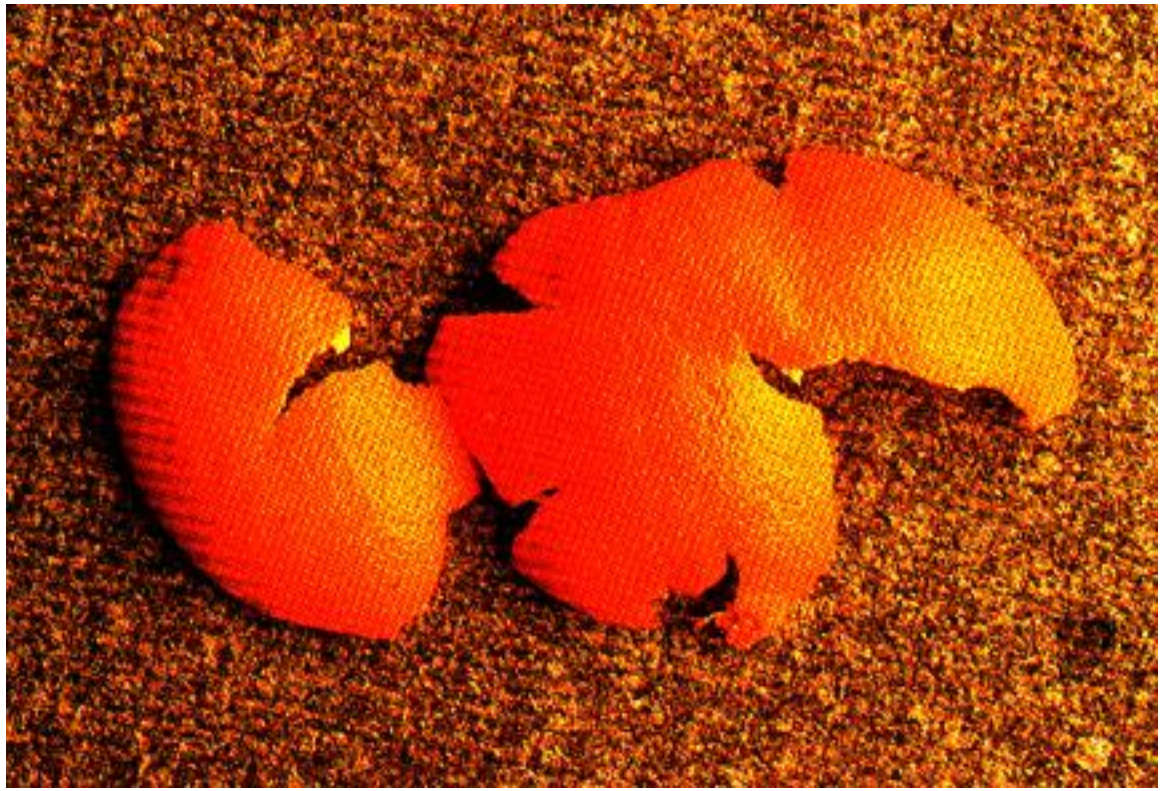


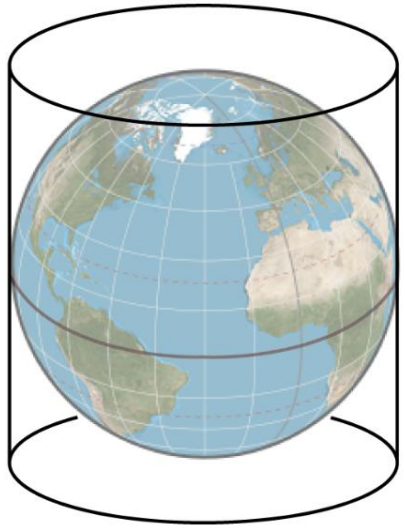
Orthographic projection

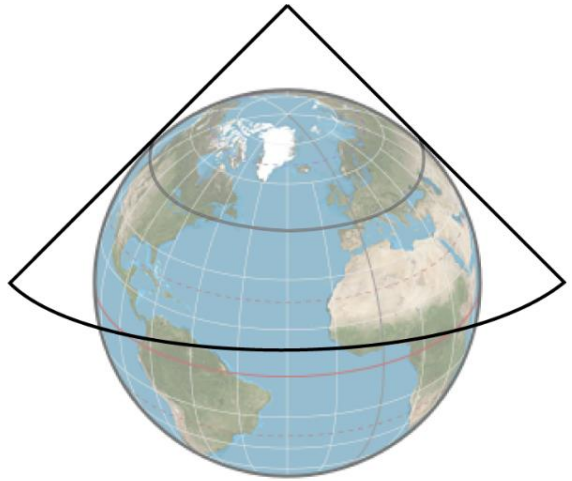


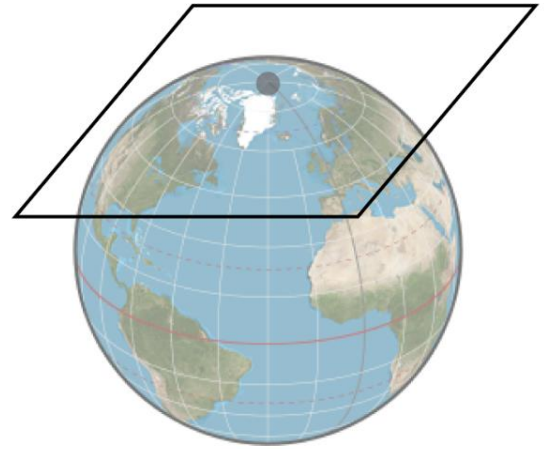
$$x = R\,\cos\varphi\,\sin(\lambda - \lambda_0)$$
$$y = R\big(\cos\varphi_0\,\sin\varphi - \sin\varphi_0\,\cos\varphi\,\cos(\lambda - \lambda_0)\big)$$

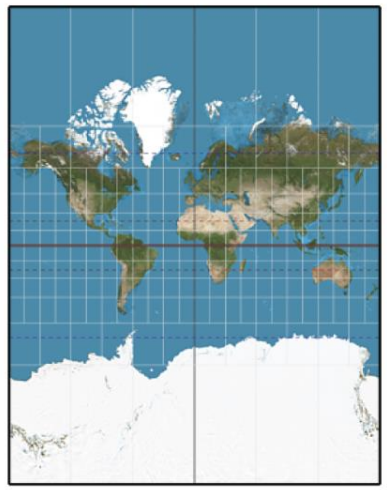
How do we display in 2-d?

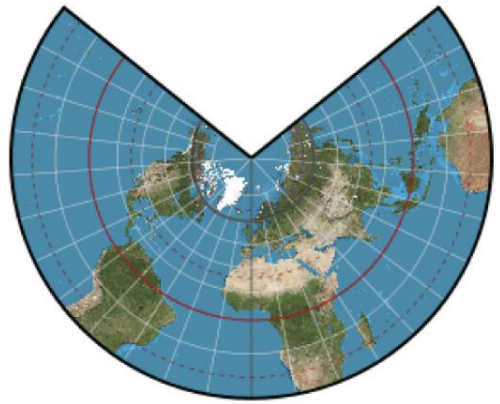


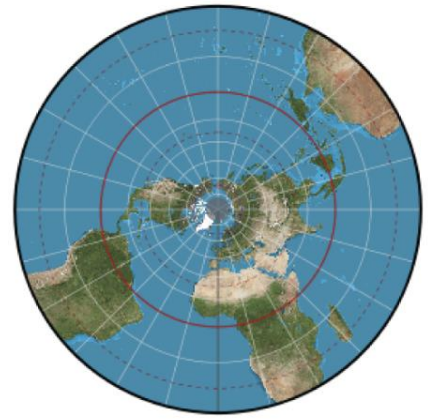










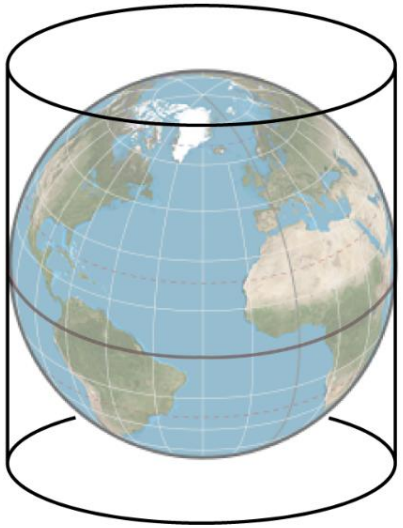


Cylindrical

Conical

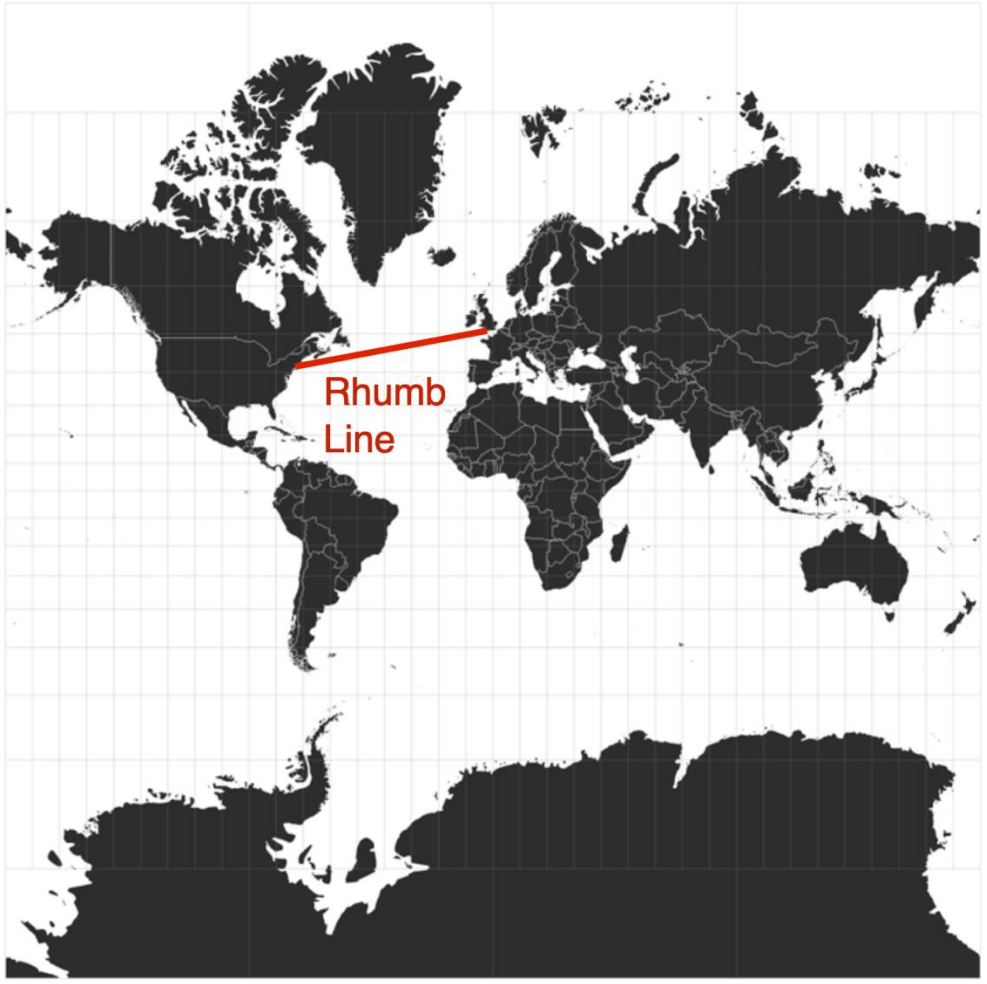
Azimuthal

Mercator projection

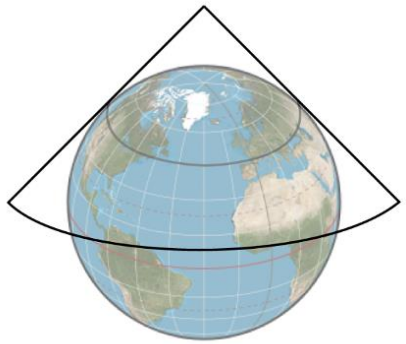


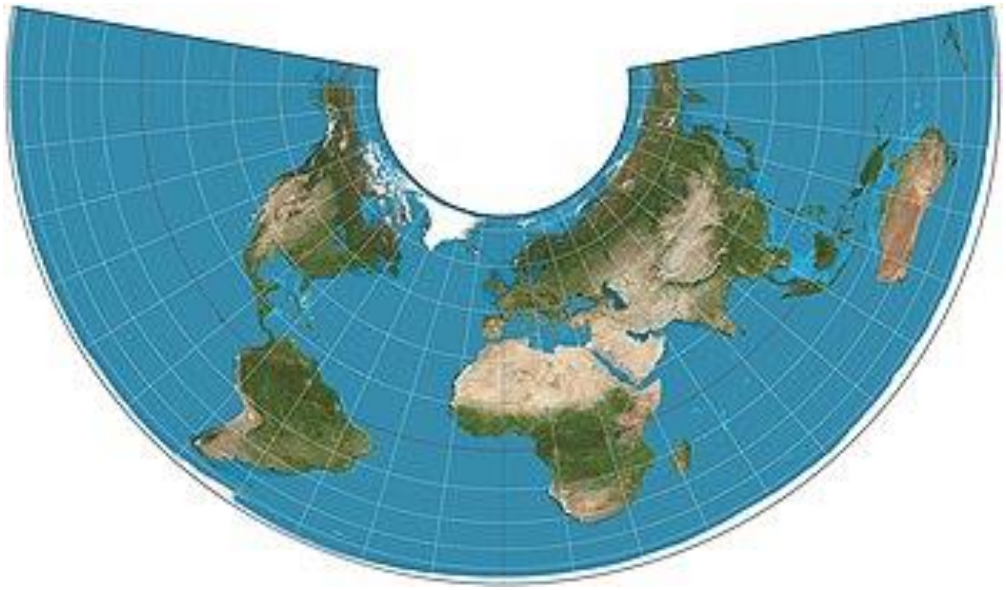
$$x = R(\lambda - \lambda_0), \qquad y = R \ln\left[\tan\left(\frac{\pi}{4} + \frac{\varphi}{2}\right)\right]$$

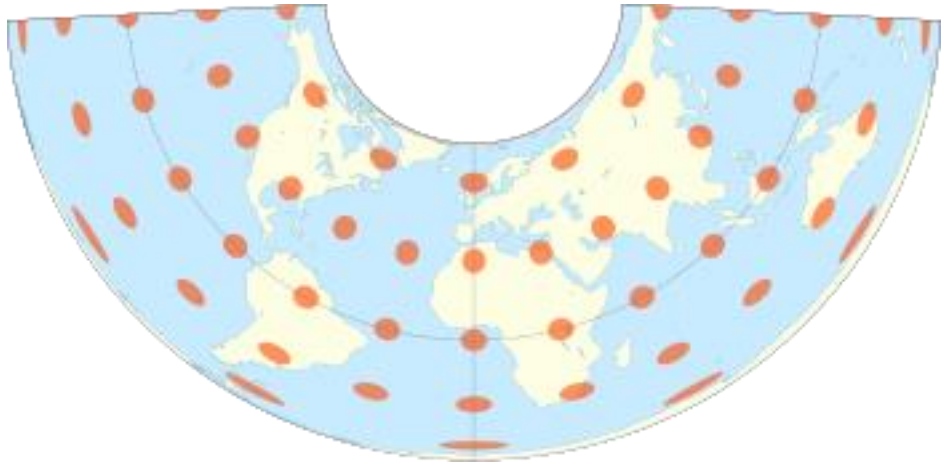
Mercator projection – Preserves direction



Albers Projection







$$x = \rho \sin\theta$$
$$y = \rho_0 - \rho \cos\theta$$

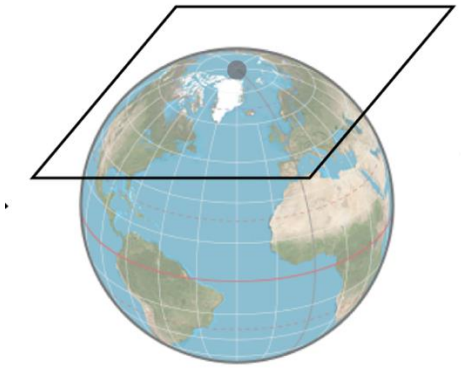
where

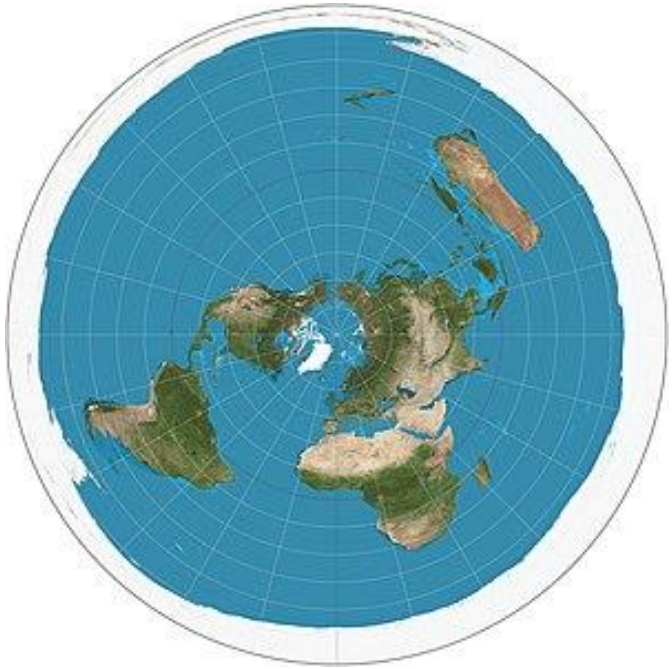
$$n = \tfrac{1}{2}\left(\sin\varphi_1 + \sin\varphi_2\right)$$
$$\theta = n\left(\lambda - \lambda_0\right)$$
$$C = \cos^2\varphi_1 + 2n\sin\varphi_1$$
$$\rho = \tfrac{R}{n}\sqrt{C - 2n\sin\varphi}$$
$$\rho_0 = \tfrac{R}{n}\sqrt{C - 2n\sin\varphi_0}$$

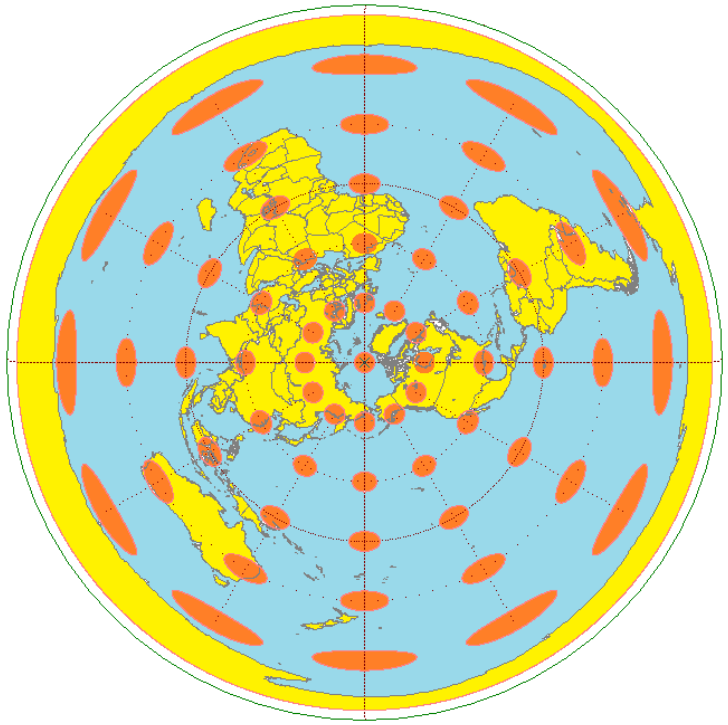
Albers Projection – preserves area

Albers USA

Azimuthal equidistant





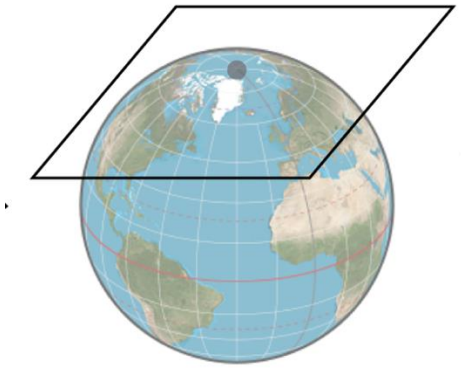


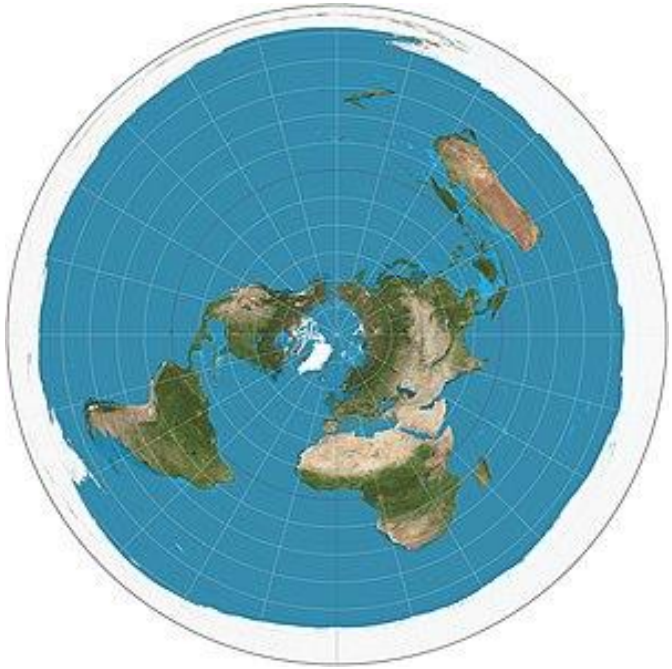
$$x = \rho \sin\theta, \qquad y = -\rho \cos\theta$$

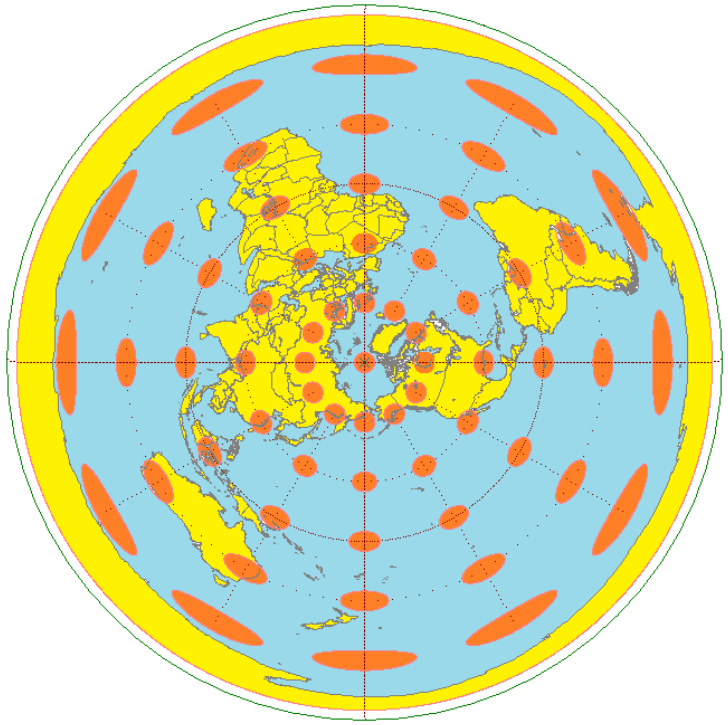
$$\cos\frac{\rho}{R} = \sin\varphi_0 \sin\varphi + \cos\varphi_0 \cos\varphi \cos(\lambda - \lambda_0)$$

$$\tan\theta = \frac{\cos\varphi \sin(\lambda - \lambda_0)}{\cos\varphi_0 \sin\varphi - \sin\varphi_0 \cos\varphi \cos(\lambda - \lambda_0)}$$

Azimuthal equidistant







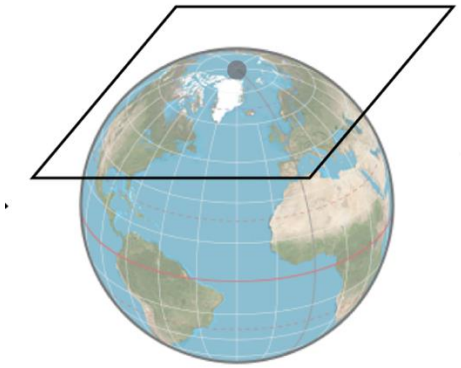
$$x = \rho \sin\theta, \qquad y = -\rho \cos\theta$$

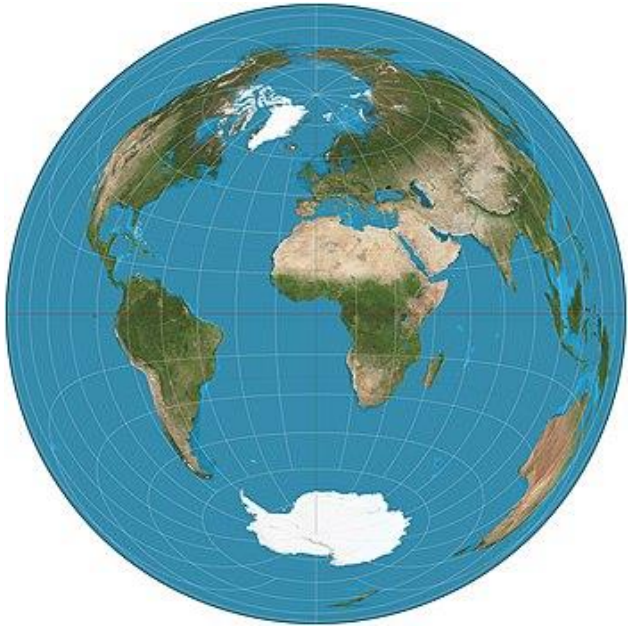
$$\cos\frac{\rho}{R} = \sin\varphi_0 \sin\varphi + \cos\varphi_0 \cos\varphi \cos(\lambda - \lambda_0)$$

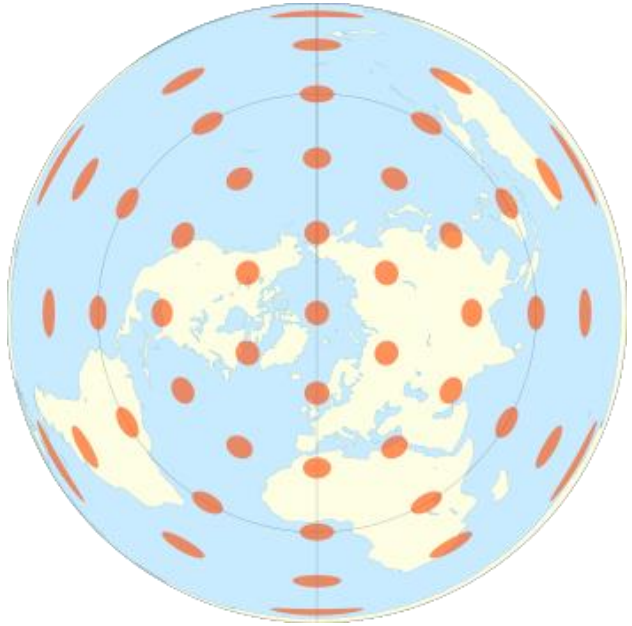
$$\tan\theta = \frac{\cos\varphi \sin(\lambda - \lambda_0)}{\cos\varphi_0 \sin\varphi - \sin\varphi_0 \cos\varphi \cos(\lambda - \lambda_0)}$$

Other equal area

Azimuthal equal area





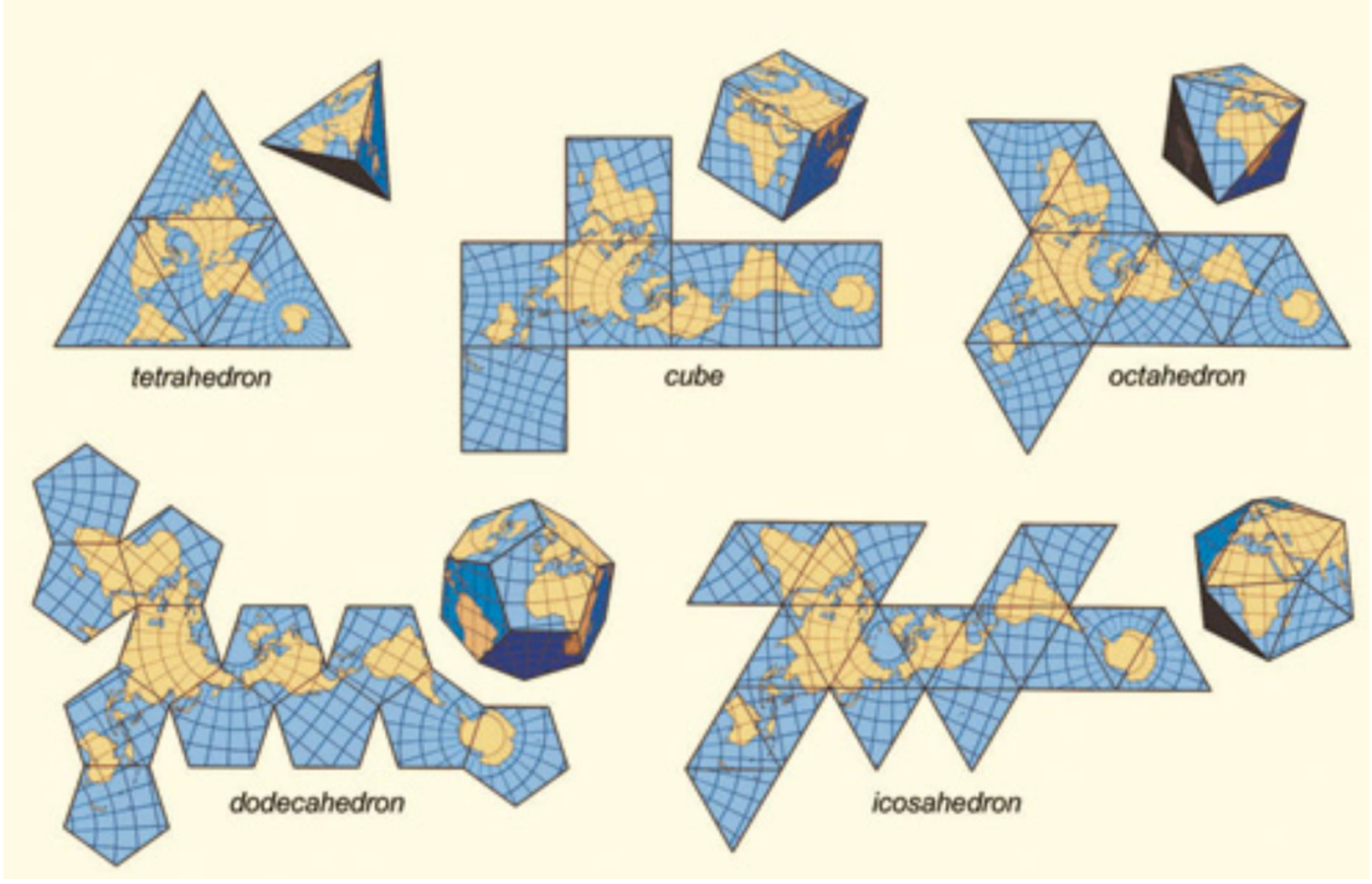


Lambert cylindrical

Equal Earth

Bottomley

Unfolding a sphere



Other projections

Scatterplot maps

GeoJSON

- Format for encoding geographic shapes
  - Shapes encoded as latitude and longitude coordinates


```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "geometry": {
        "type": "Point",
        "coordinates": [102.0, 0.5]
      },
      "properties": {
        "prop0": "value0"
      }
    },
    {
      "type": "Feature",
      "geometry": {
        "type": "LineString",
        "coordinates": [
          [102.0, 0.0],
          [103.0, 1.0],
          [104.0, 0.0],
          [105.0, 1.0]
        ]
      },
      "properties": {
        "prop0": "value0",
        "prop1": 0.0
      }
    },
    {
      "type": "Feature",
      "geometry": {
        "type": "Polygon",
        "coordinates": [
          [
            [100.0, 0.0],
            [101.0, 0.0],
            [101.0, 1.0],
            [100.0, 1.0],
            [100.0, 0.0]
          ]
        ]
      },
      "properties": {
        "prop0": "value0",
        "prop1": { "this": "that" }
      }
    }
  ]
}
```

GeoJSON


```
Object {
 type: "FeatureCollection"
 totalFeatures: 214
 features: ▼Array(214) [
   0: ▶Object {type: "Feature", id: "nyu_2451_34509.1", geometry: Object, geometry_name: "geom", properties:
   1: ▶Object {type: "Feature", id: "nyu_2451_34509.2", geometry: Object, geometry_name: "geom", properties:
   2: ▶Object {type: "Feature", id: "nyu_2451_34509.3", geometry: Object, geometry_name: "geom", properties:
   3: ▶Object {type: "Feature", id: "nyu_2451_34509.7", geometry: Object, geometry_name: "geom", properties:
   4: ▶Object {type: "Feature", id: "nyu_2451_34509.8", geometry: Object, geometry_name: "geom", properties:
   5: ▶Object {type: "Feature", id: "nyu_2451_34509.34", geometry: Object, geometry_name: "geom", properties:
   6: ▶Object {type: "Feature", id: "nyu_2451_34509.4", geometry: Object, geometry_name: "geom", properties:
   7: ▶Object {type: "Feature", id: "nyu_2451_34509.5", geometry: Object, geometry_name: "geom", properties:
   8: ▶Object {type: "Feature", id: "nyu_2451_34509.6", geometry: Object, geometry_name: "geom", properties:
   9: ▶Object {type: "Feature", id: "nyu_2451_34509.9", geometry: Object, geometry_name: "geom", properties:
   10: ▶Object {type: "Feature", id: "nyu_2451_34509.10", geometry: Object, geometry_name: "geom", properties
   11: ▶Object {type: "Feature", id: "nyu_2451_34509.11", geometry: Object, geometry_name: "geom", properties
   12: ▶Object {type: "Feature", id: "nyu_2451_34509.12", geometry: Object, geometry_name: "geom", properties
   13: ▶Object {type: "Feature", id: "nyu_2451_34509.30", geometry: Object, geometry_name: "geom", properties
   14: ▶Object {type: "Feature", id: "nyu_2451_34509.13", geometry: Object, geometry_name: "geom", properties
   15: ▶Object {type: "Feature", id: "nyu_2451_34509.14", geometry: Object, geometry_name: "geom", properties
   16: ▶Object {type: "Feature", id: "nyu_2451_34509.15", geometry: Object, geometry_name: "geom", properties
   17: ▶Object {type: "Feature", id: "nyu_2451_34509.16", geometry: Object, geometry_name: "geom", properties
   18: ▶Object {type: "Feature", id: "nyu_2451_34509.31", geometry: Object, geometry_name: "geom", properties
   19: ▶Object {type: "Feature", id: "nyu_2451_34509.17", geometry: Object, geometry_name: "geom", properties
   … more
 ]
 crs: ▶Object {type: "name", properties: Object}
 bbox: ▶Array(4) [-74.258918762207, 40.4938278198242, -73.6479721069336, 40.9173812866211]
```

GeoJSON


```
▼ Object {
  type: "Feature"
  id: "nyu_2451_34509.1"
  geometry: ▼ Object {
    type: "MultiPolygon"
    coordinates: ▼ Array(1) [
      0: ▶ Array(3) [Array(67), Array(6), Array(9)]
    ]
  }
  geometry_name: "geom"
  properties: ▶ Object {zcta: "10001", bcode: "36061", note: null, bbox: Array(4)}
}
```


```
nycGeo.features[0]
```

```
projection = ƒ(t)
```

```
projection =  d3.geoAlbers()
      .fitSize([mapWidth, mapHeight], nycGeo)
```

```
path = ƒ(t)
```

```
path = d3.geoPath().projection(projection)
```

```
<svg width=${mapWidth} height=${mapHeight}>
  <path
    d=${path(nycGeo)}
    fill="#d3d3d3"
    stroke="white"
  />
</svg>
```

```
projection = ƒ(t)
```

```
projection =  d3.geoAlbers()
      .fitSize([mapWidth, mapHeight], nycGeo)
```

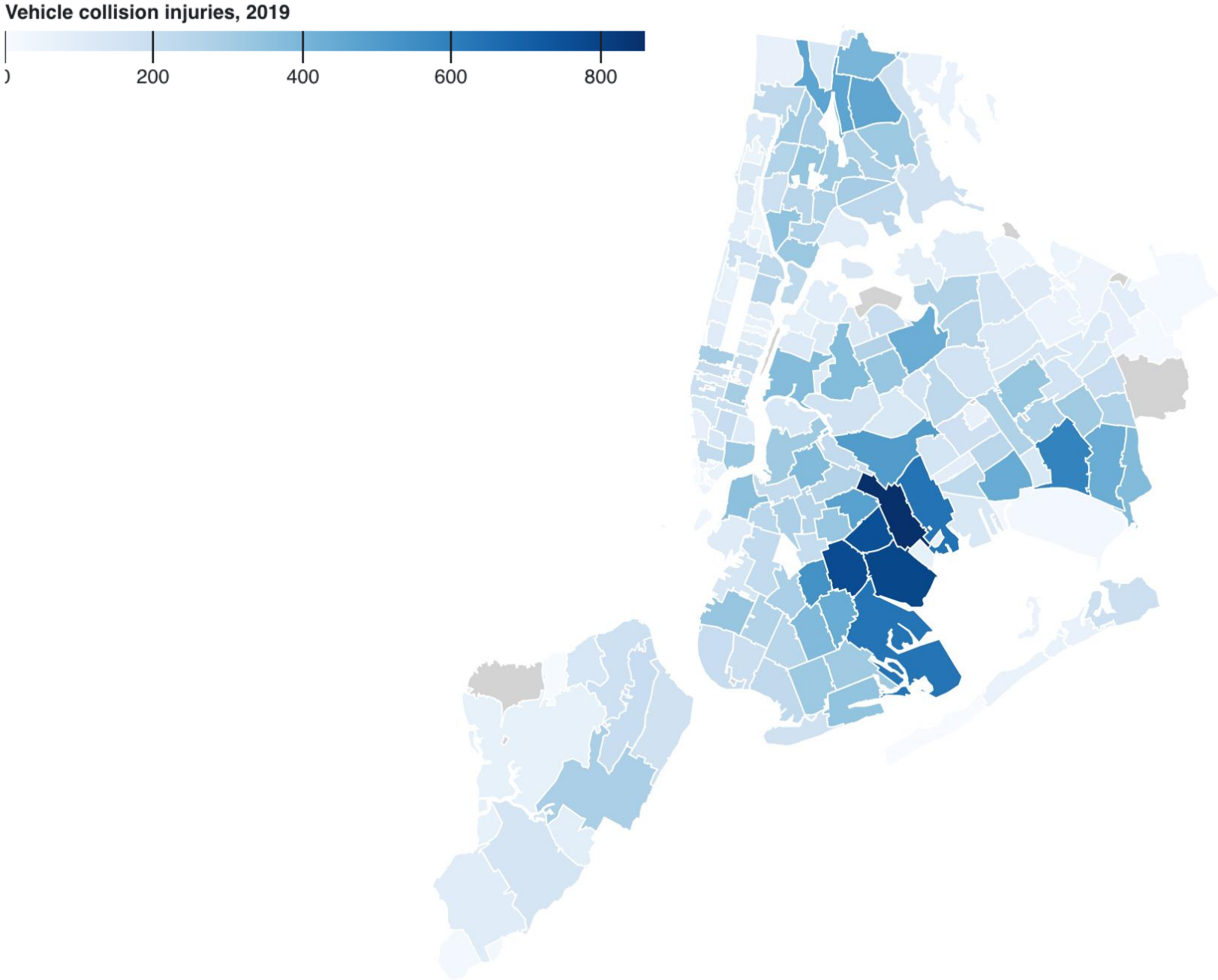
```
path = ƒ(t)
```

```
path = d3.geoPath().projection(projection)
```

```
<svg width=${mapWidth} height=${mapHeight}>
  <path
    d=${path(nycGeo.features[0])}
    fill="#d3d3d3"
    stroke="white"
  />
</svg>
```

```
{
  const svg = d3.create('svg')
      .attr('width', mapWidth)
      .attr('height', mapHeight);

  // draw one svg path per zip code
  svg.selectAll("path")
    .data(nycGeo.features)
    .join("path")
      .attr("d", path) // same as `d => path(d)`
      .attr("fill", '#d3d3d3')
      .attr("stroke", "white");

  return svg.node();
}
```

```
{
  const svg = d3.create('svg')
      .attr('width', mapWidth)
      .attr('height', mapHeight);

  // draw one svg path for the entire map
  svg.append("path")
      .attr("d", path(nycGeo))
      .attr("fill", '#d3d3d3')
      .attr("stroke", "white");

  return svg.node();
}
```


Vehicle collision injuries, 2019
0
200
400
600
800

```
▼ Object {
  10000: 32
  10001: 187
  10002: 323
  10003: 203
  10004: 37
  10005: 16
  10006: 26
  10007: 62
  10009: 114
  10010: 183
  10011: 151
  10012: 135
  10013: 209
  10014: 71
  10016: 298
  10017: 153
  10018: 127
  10019: 294
  10020: 12
  10021: 90
  … more
}
```

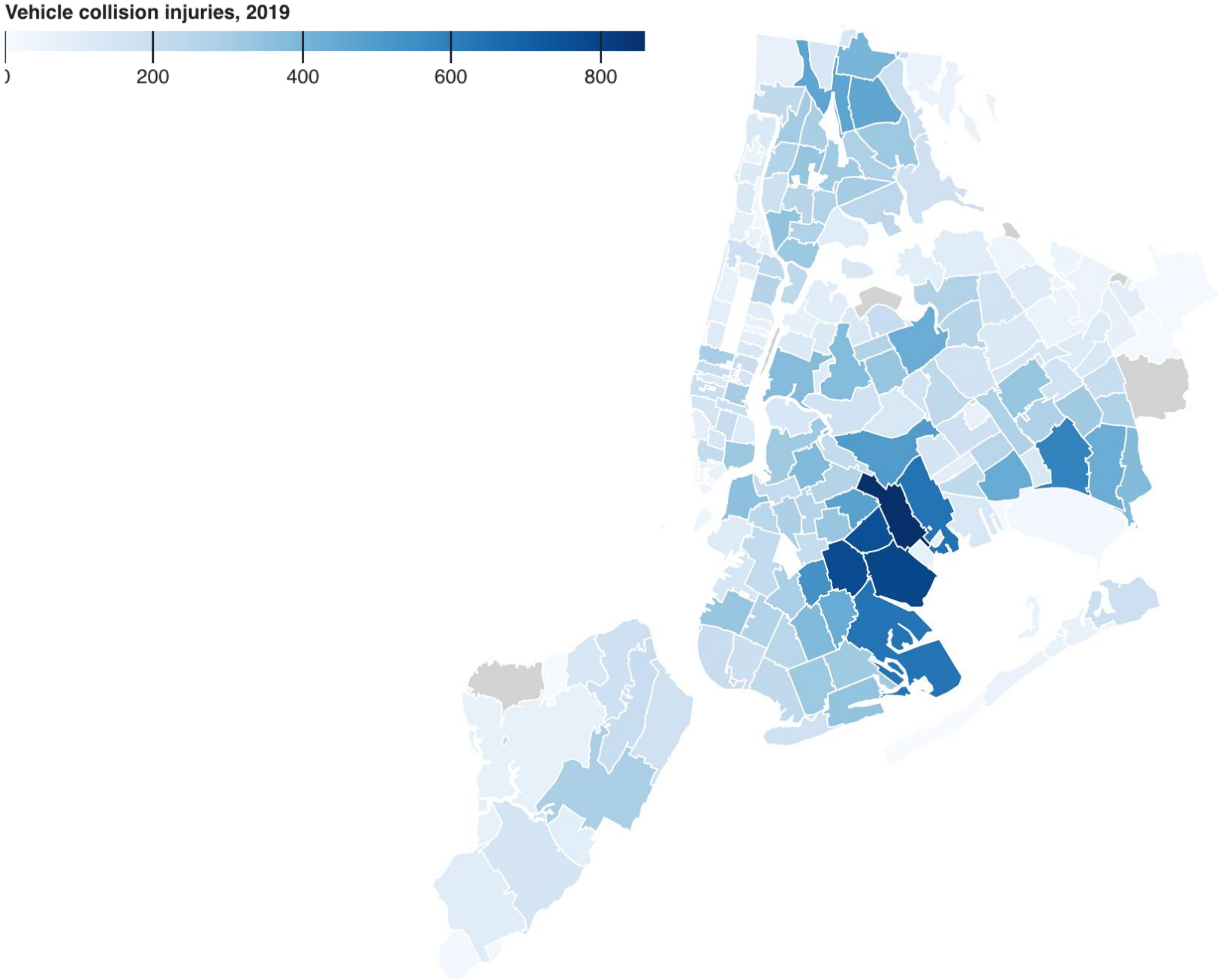
```
const color = d3.scaleSequential(d3.interpolateBlues);
```

```
color = d3.scaleSequential()
  .domain([0, maxInjuriesForZip])
  .interpolator(d3.interpolateBlues)
  .unknown('#d3d3d3')
```

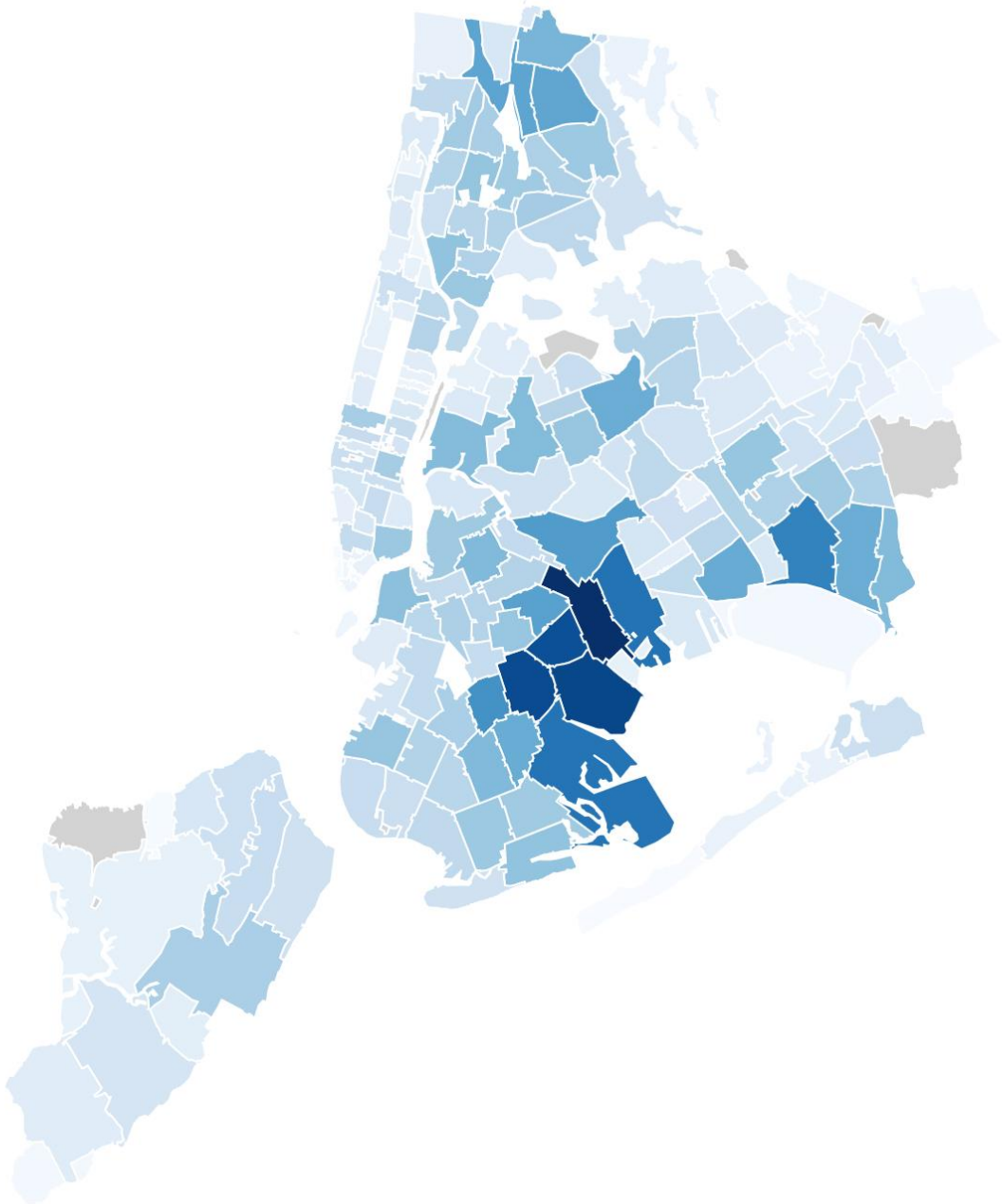
```
zipToInjuries
```

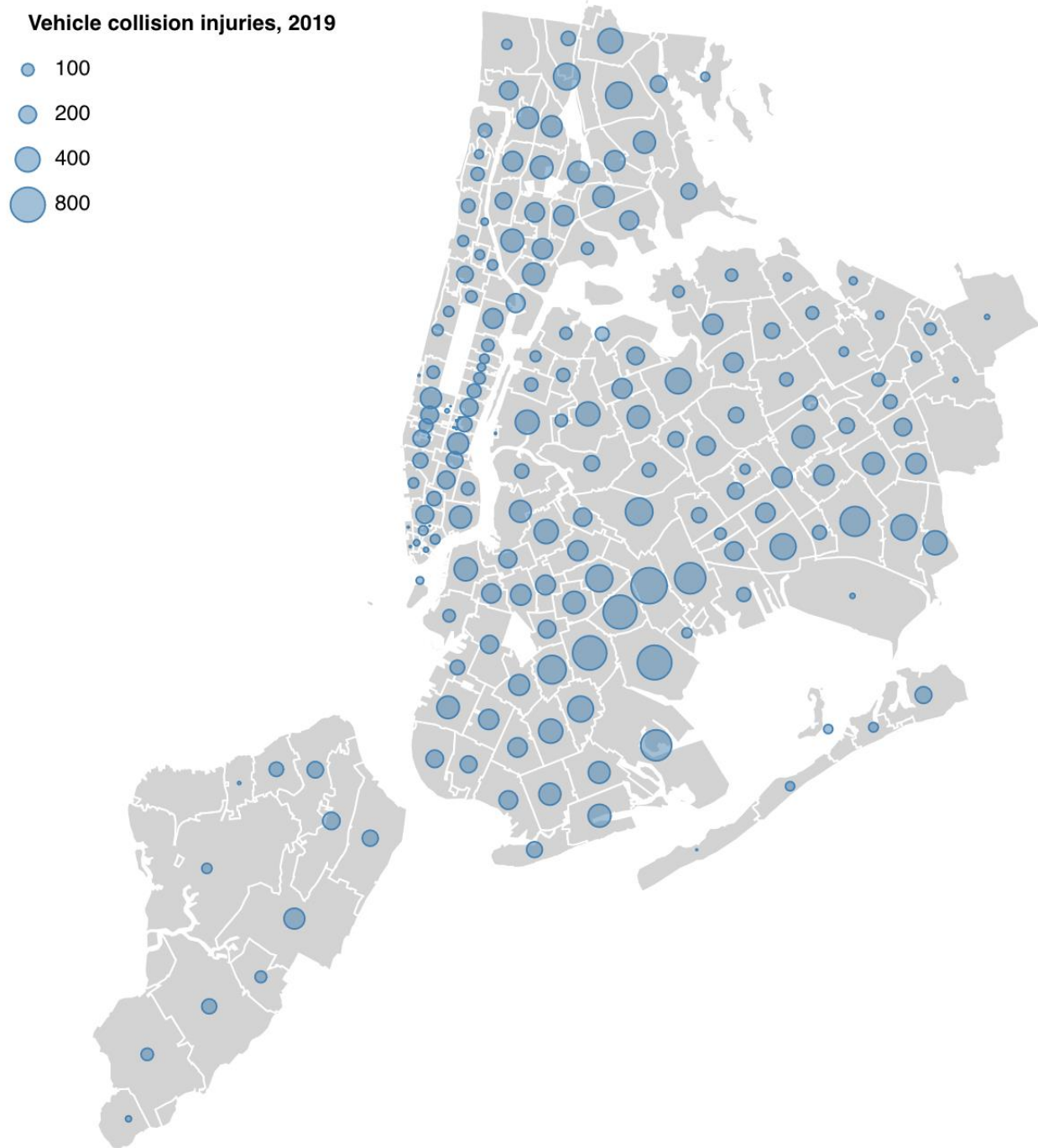
```
maxInjuriesForZip = 860
```

```
maxInjuriesForZip = d3.max(Object.values(zipToInjuries))
```


Vehicle collision injuries, 2019
0
200
400
600
800

```
{
  const svg = d3.create('svg')
      .attr('width', mapWidth)
      .attr('height', mapHeight);

  // create one path for each zip code
  svg.selectAll("path")
    .data(nycGeo.features)
    .join("path")
      // use the geographic path generator to set the shape of the path
      .attr("d", path)
      // use the color scale to set the color of the path
      .attr("fill", d => color(zipToInjuries[d.properties.zcta]))
      .attr("stroke", "white");

  return svg.node();
}
```



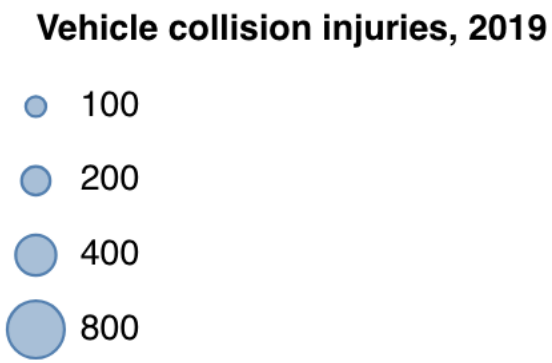

Vehicle collision injuries, 2019
100
200
400
800

```
maxRadius = 10
```

```
maxRadius = 10
```

```
radius = f(n)
```

```
// we want to map number of injuries to the area of the circle,
// so we use scaleSqrt since the area of a circle is pi * (r^2)
radius = d3.scaleSqrt()
  .domain([0, maxInjuriesForZip])
  .range([0, maxRadius])
```

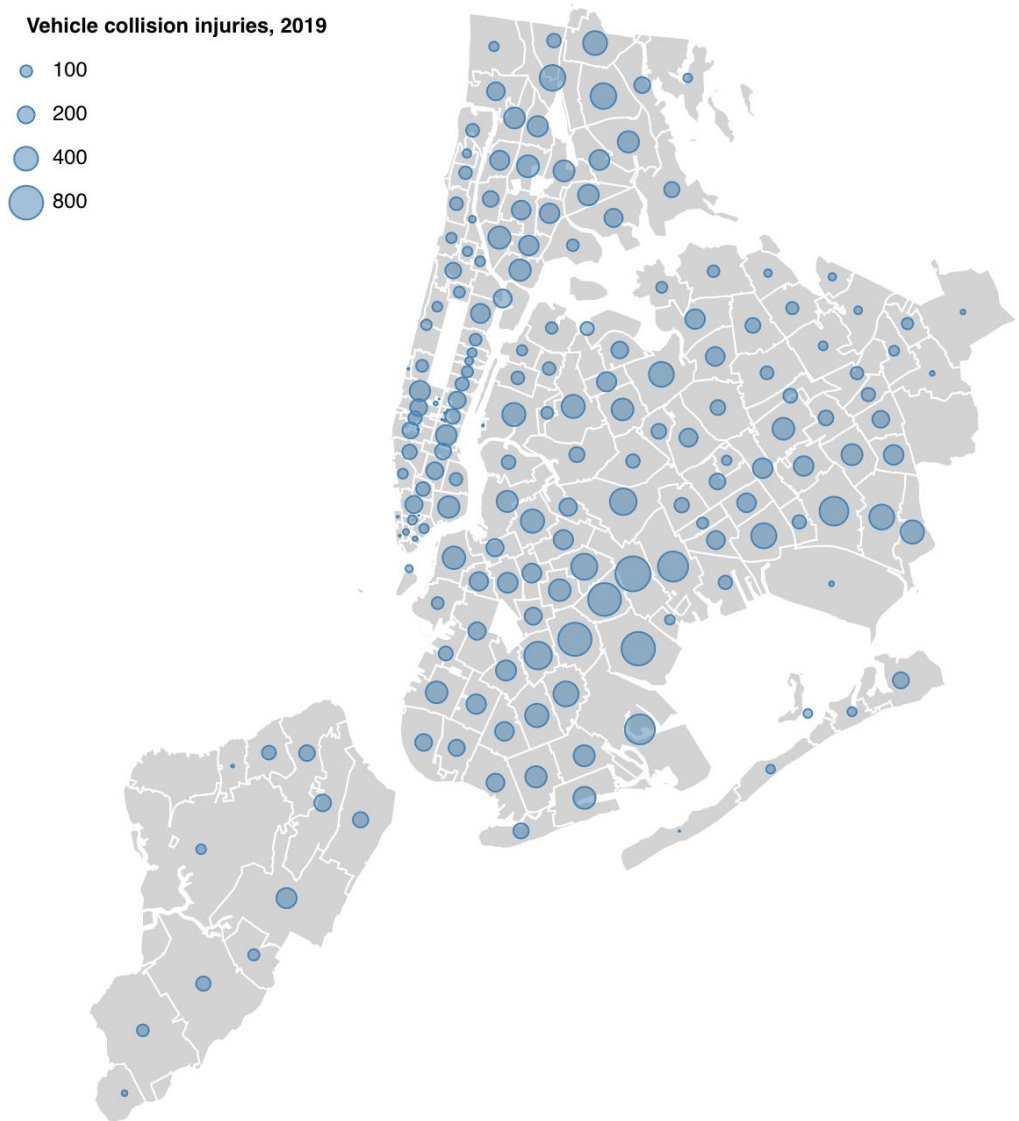


```
const svg = d3.create('svg')
    .attr('width', mapWidth)
    .attr('height', mapHeight);

// draw map

svg.selectAll("path")
  .data(nycGeo.features)
  .join("path")
    .attr("d", path)
    .attr("fill", '#d3d3d3')
    .attr("stroke", "white");
```

▸ Array(2) [238.85631317949486, 244.91887758343373]

```
path.centroid(nycGeo.features[0])
```

```
// draw circles

svg.append("g")
  .selectAll("circle")
  .data(nycGeo.features)
  .join("circle")
    .attr("stroke", "steelblue")
    .attr("fill", "steelblue")
    .attr("fill-opacity", 0.5)
    // set the position of the circles
    .attr("transform", d => `translate(${path.centroid(d)})`)
    // set the size of the circles
    .attr("r", d => radius(zipToInjuries[d.properties.zcta]));
```



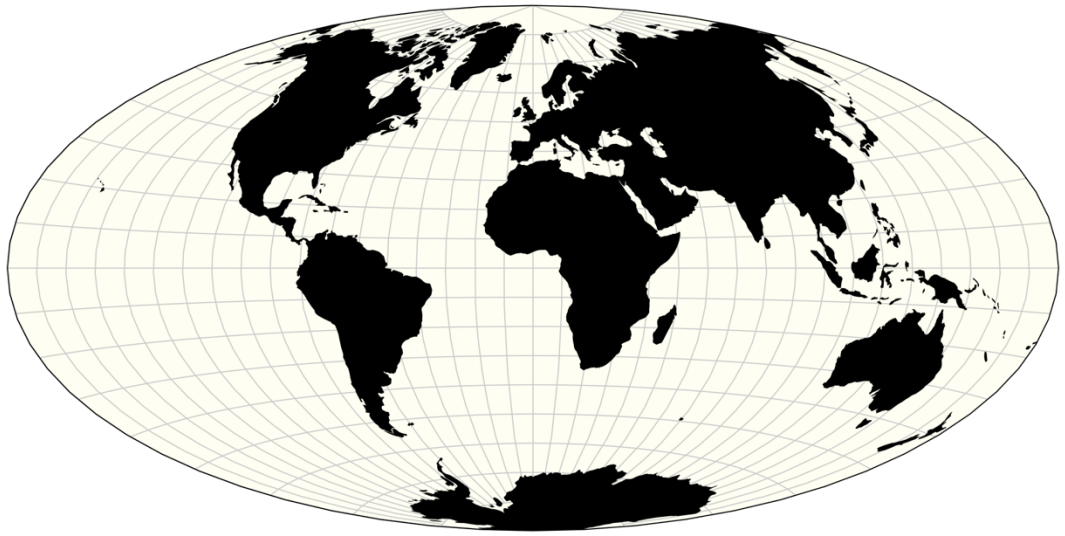
Projections

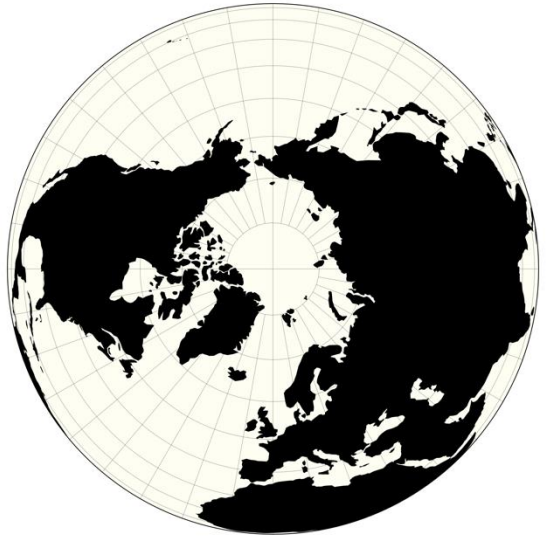
```
const path = d3.geoPath(projection);
```

```
svg
  .append("g")
  .selectAll("path")
  .data(l.features)
  .enter()
  .append("path")
  .attr("id", d => d.properties.name)
  .attr("fill", "black");
```

```
graticule = ▼Object {
  type: "MultiLineString"
  coordinates: ▶Array(53) [Array(3),
}
```

```
graticule = d3.geoGraticule10()
```





Approaches to Mapping Data

Symbol Maps → plot data over a map

Choropleth Maps → colored regions

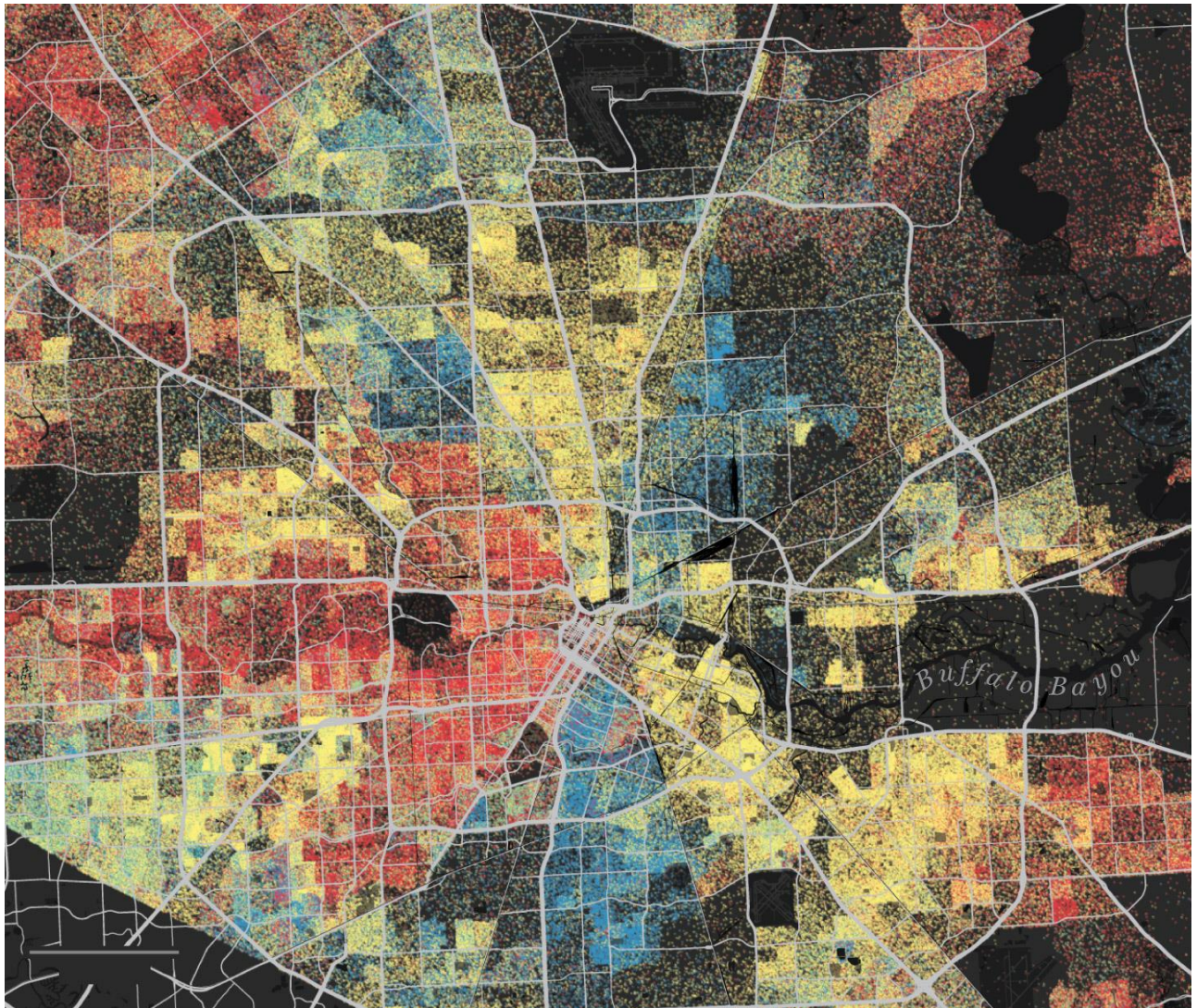
Heatmaps & Contours → show densities

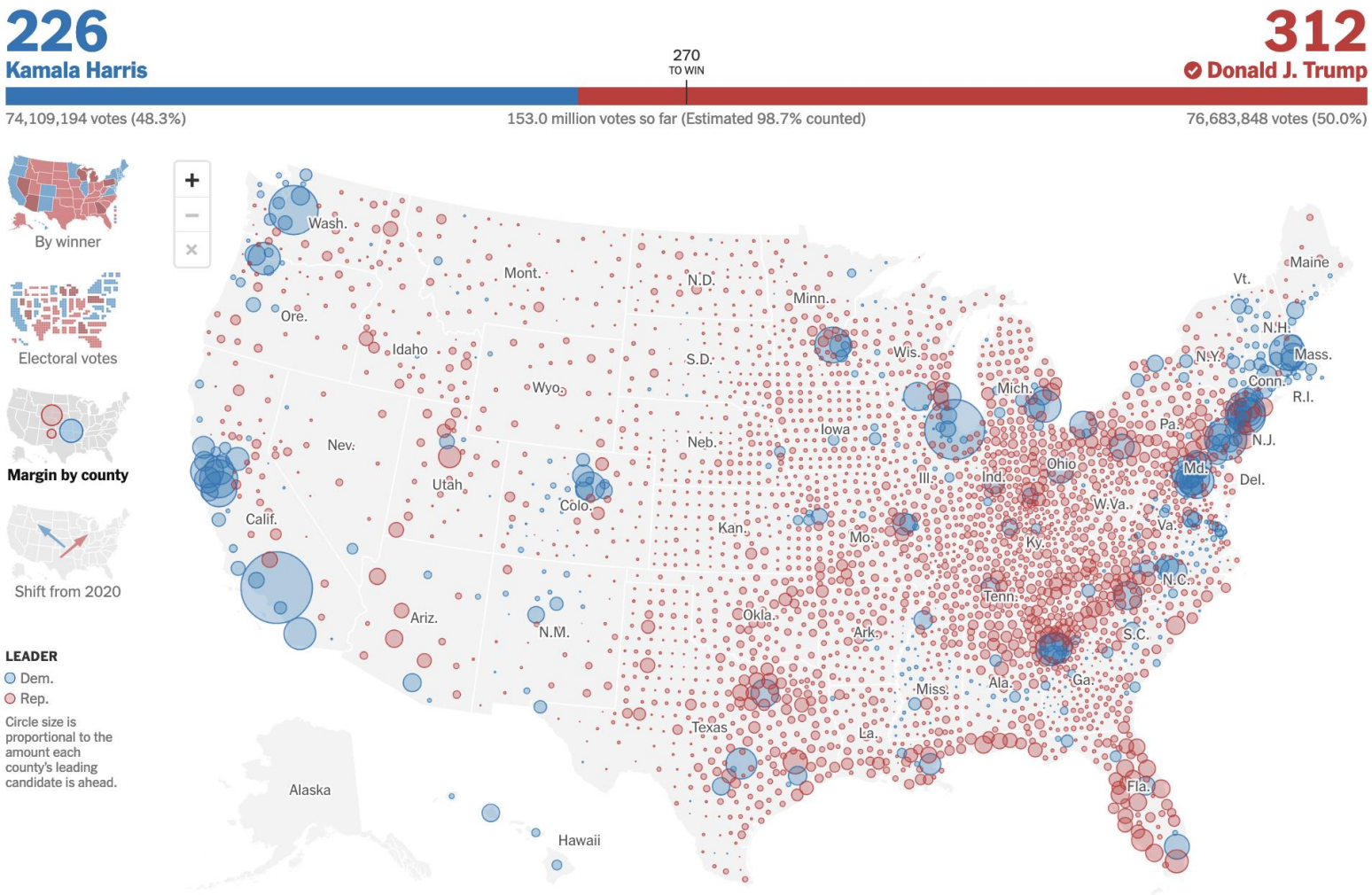
Cartograms → distort to show quantities

Flow Maps → flux across regions

Generalization → distort/abstract to aid tasks

Symbol map (scatterplot)




226
Kamala Harris
270
TO WIN
312
Donald J. Trump
74,109,194 votes (48.3%)
153.0 million votes so far (Estimated 98.7% counted)
76,683,848 votes (50.0%)
By winner
Electoral votes
Margin by county
Shift from 2020
LEADER
Dem.
Rep.
Circle size is proportional to the amount each county's leading candidate is ahead.
Wash.
Ore.
Calif.
Idaho
Nev.
Mont.
Wyo.
Utah
Colo.
Ariz.
N.M.
N.D.
S.D.
Neb.
Kan.
Okla.
Texas
Minn.
Iowa
Mo.
Ark.
La.
Wis.
Ill.
Mich.
Ind.
Ohio
Ky.
Tenn.
Miss.
Ala.
Ga.
Fla.
S.C.
N.C.
Va.
W.Va.
Pa.
N.Y.
Md.
Del.
N.J.
Conn.
R.I.
Mass.
Vt.
N.H.
Maine
Alaska
Hawaii

226
Kamala Harris

270
TO WIN

312
Donald J. Trump

74,109,194 votes (48.3%) | 153.0 million votes so far (Estimated 98.7% counted) | 76,683,848 votes (50.0%)


By winner


Electoral votes

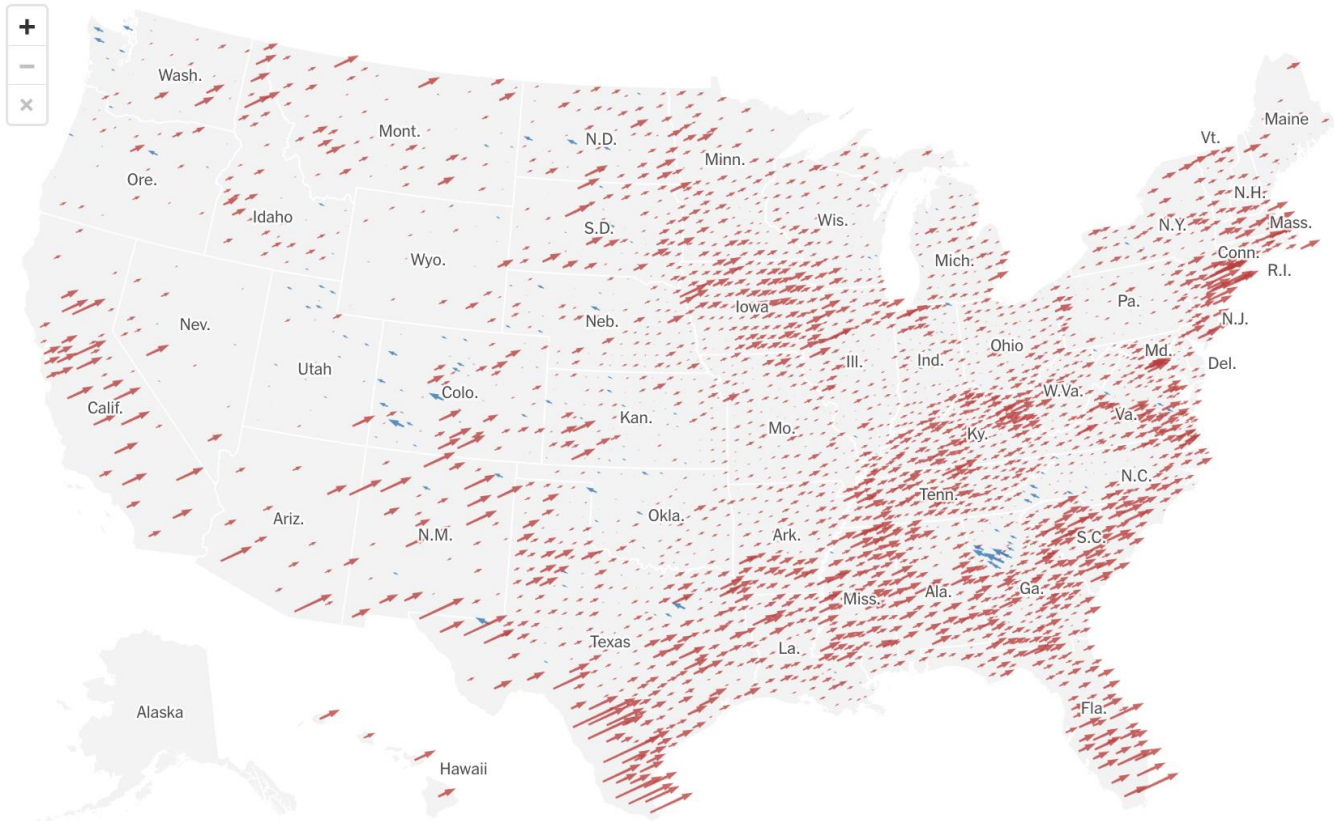

Margin by county


Shift from 2020

SHIFT IN MARGIN

More Dem. More Rep.

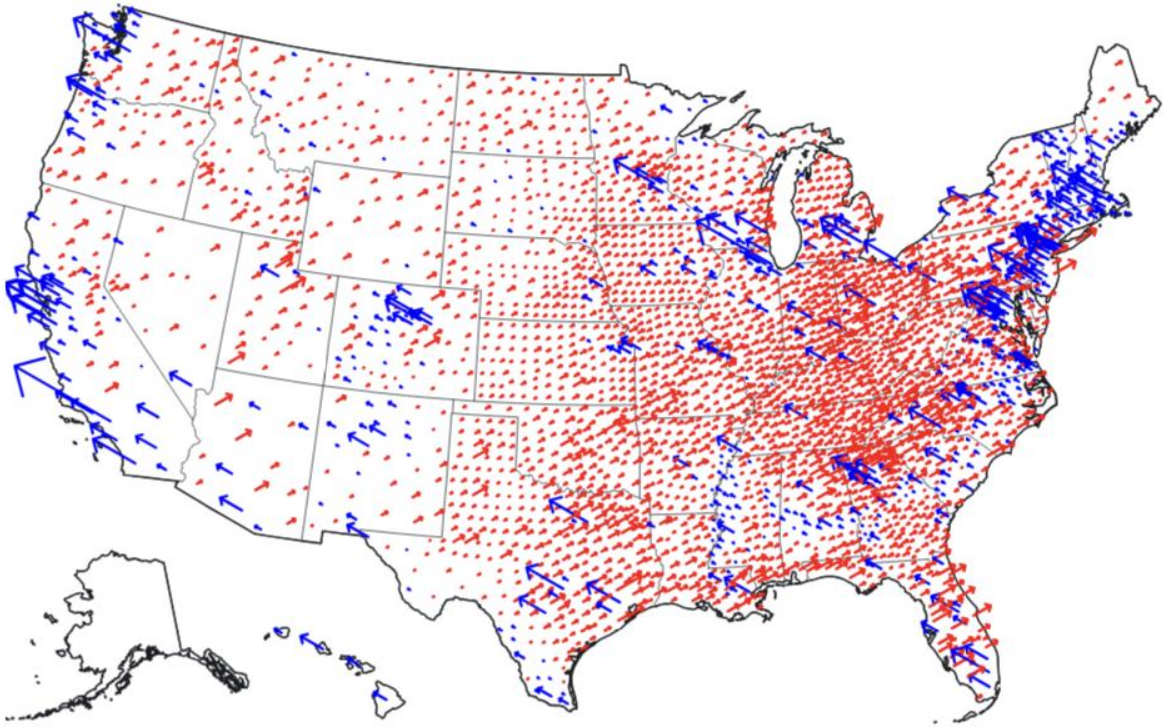
Compared with 2020 presidential vote in places that have reported almost all of their votes.



In observable plot

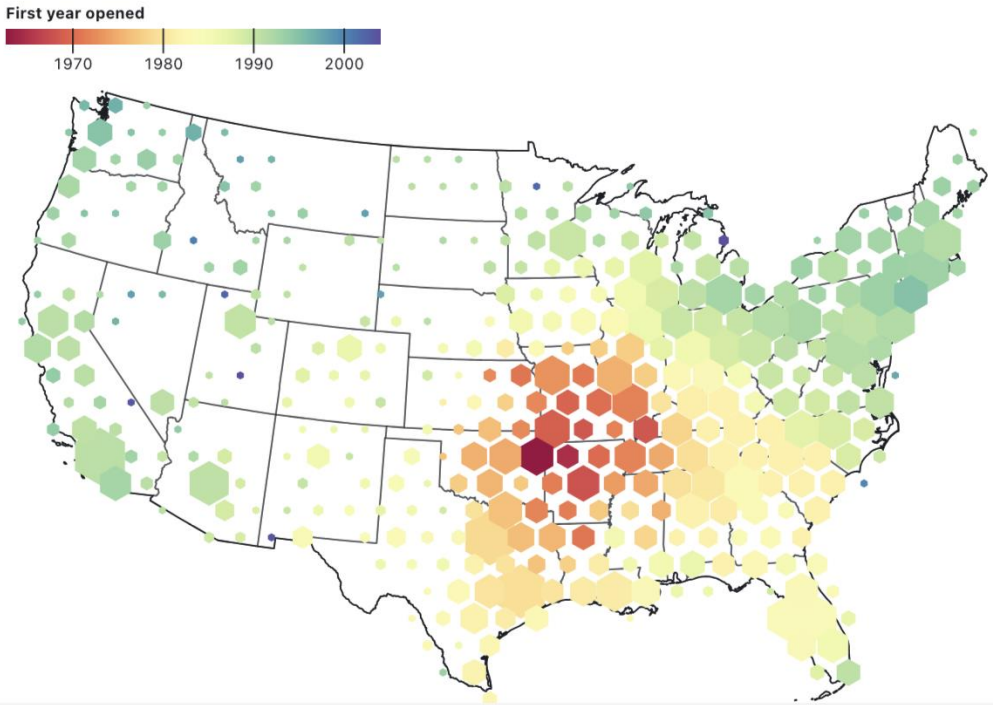
```
Plot.plot({
  projection: {type: "orthographic", rotate: [-longitude, -30]},
  r: {transform: (d) => Math.pow(10, d)}, // convert Richter to amplitude
  style: "overflow: visible;", // allow dots to escape
  marks: [
    Plot.geo(land, {fill: "currentColor", fillOpacity: 0.2}),
    Plot.sphere(),
    Plot.dot(earthquakes, {x: "longitude", y: "latitude", r: "magnitude", stroke: "red", fill: "red", fillOpacity: 0.2})
  ]
})
```

In observable plot



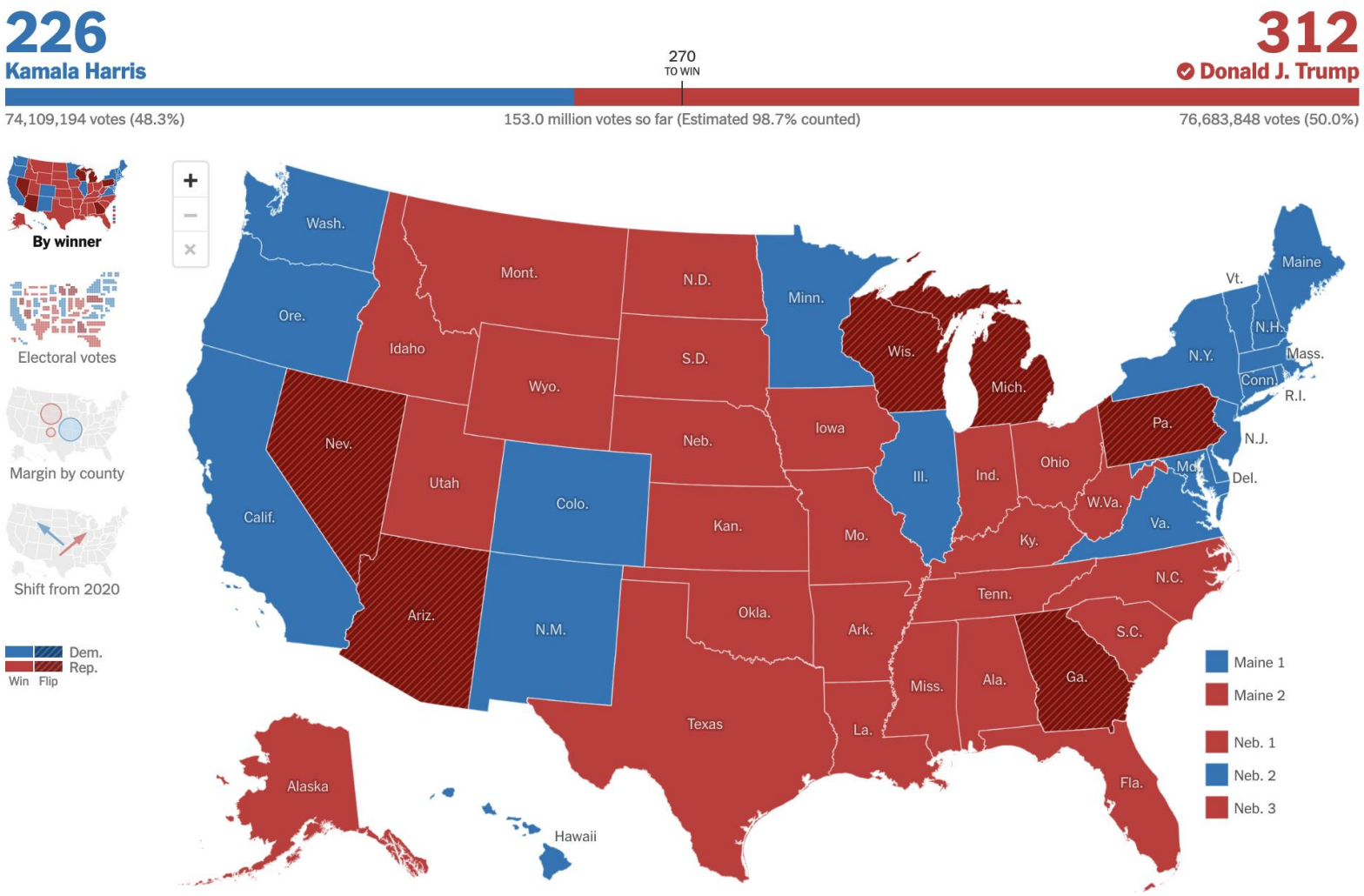
```
Plot.plot({
  projection: "albers-usa",
  length: {type: "sqrt", transform: Math.abs},
  marks: [
    Plot.geo(statemesh, {strokeWidth: 0.5}),
    Plot.geo(nation),
    Plot.vector(
      counties,
      Plot.centroid({
        anchor: "start",
        length: (d) => d.properties.margin2020 * d.properties.votes,
        stroke: (d) => d.properties.margin2020 > 0 ? "red" : "blue",
        rotate: (d) => d.properties.margin2020 > 0 ? 60 : -60
      })
    )
  ]
})
```

With binning

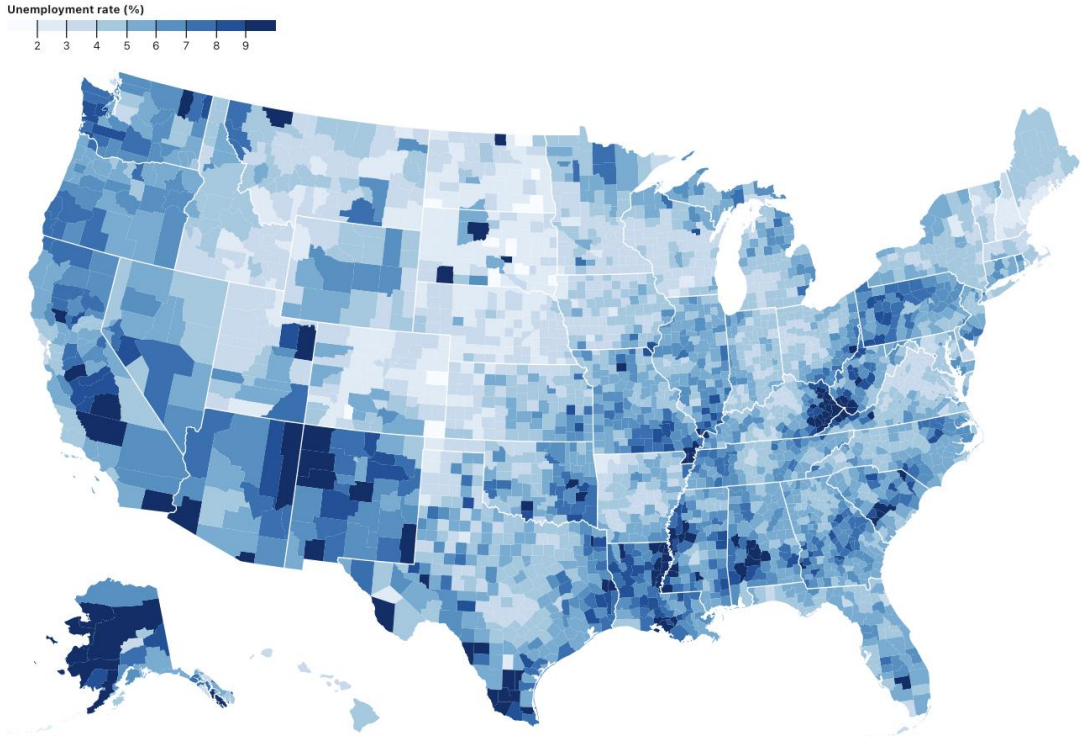



```
Plot.plot({
  projection: "albers",
  r: {range: [0, 16]},
  color: {scheme: "spectral", label: "First year opened", legend: true},
  marks: [
    Plot.geo(statemesh, {strokeOpacity: 0.5}),
    Plot.geo(nation),
    Plot.dot(walmarts, Plot.hexbin({r: "count", fill: "min"}, {x: "longitude", y: "latitude", fill: "date"}))
  ]
})
```

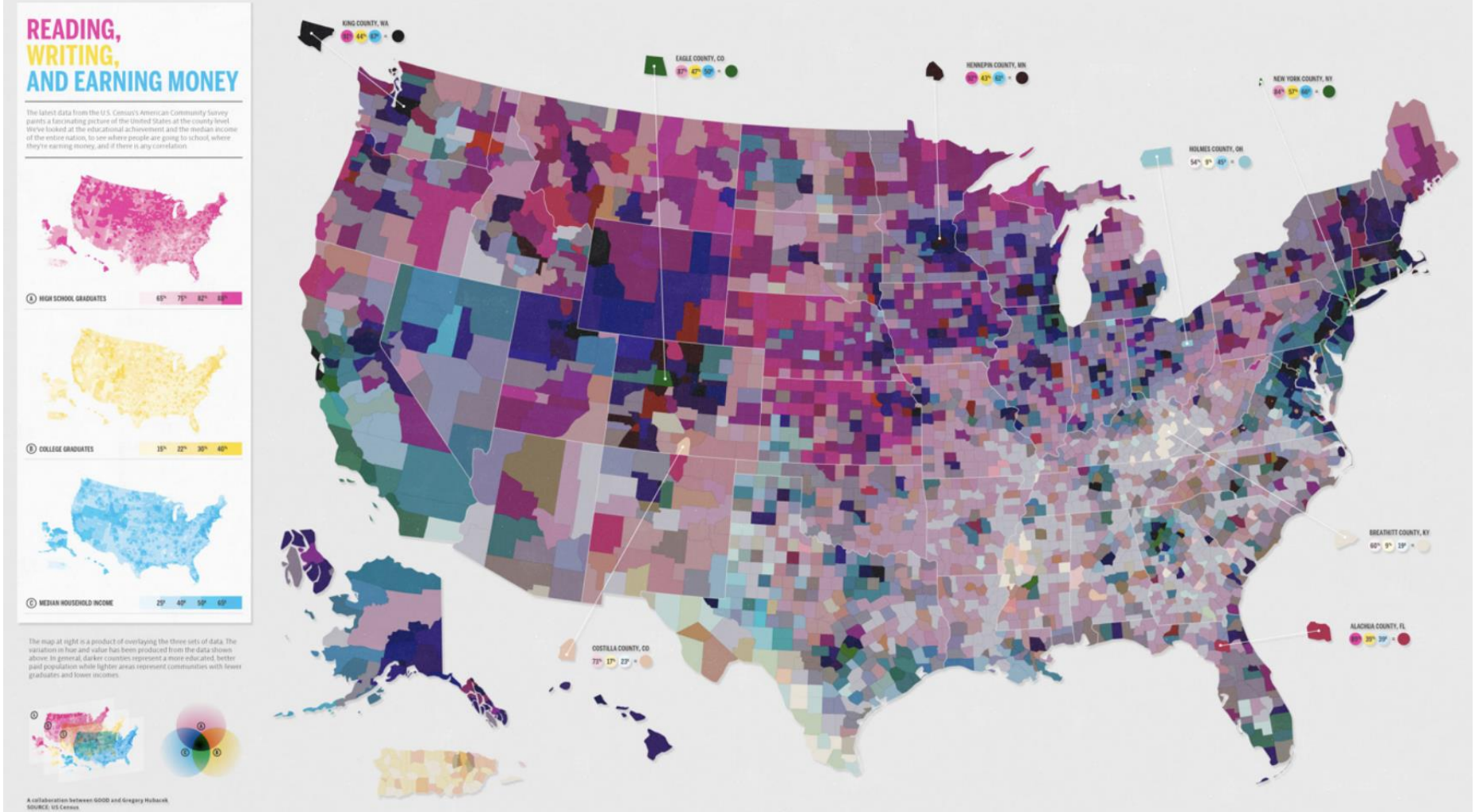
Choropleth


226
Kamala Harris
270
TO WIN
312
Donald J. Trump
74,109,194 votes (48.3%)
153.0 million votes so far (Estimated 98.7% counted)
76,683,848 votes (50.0%)
By winner
Electoral votes
Margin by county
Shift from 2020
Dem.
Rep.
Win Flip
Wash.
Ore.
Calif.
Idaho
Nev.
Mont.
Wyo.
Utah
Ariz.
Colo.
N.M.
N.D.
S.D.
Neb.
Kan.
Okla.
Texas
Minn.
Iowa
Mo.
Ark.
La.
Wis.
Ill.
Mich.
Ind.
Ohio
Ky.
Tenn.
Miss.
Ala.
Ga.
Fla.
S.C.
N.C.
Va.
W.Va.
Pa.
N.Y.
Vt.
Maine
N.H.
Mass.
Conn.
R.I.
N.J.
Md.
Del.
Alaska
Hawaii
Maine 1
Maine 2
Neb. 1
Neb. 2
Neb. 3

Choropleth in observable plot



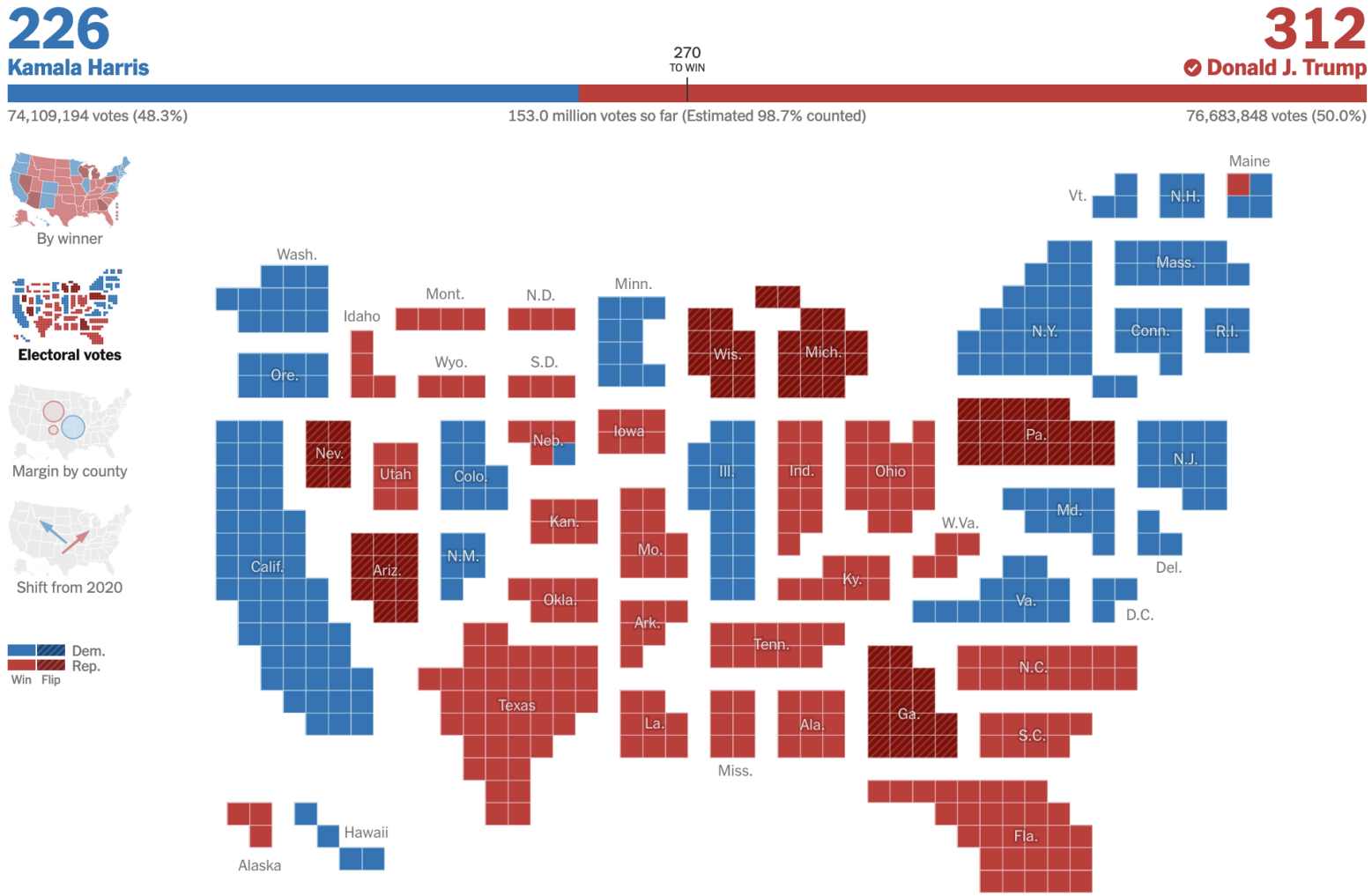

```
Plot.plot({
  projection: "identity",
  width: 975,
  height: 610,
  color: {scheme: "Blues", type: "quantize", n: 9, domain: [1, 10], label: "Unemployment rate (%)", legend: true},
  marks: [
    Plot.geo(topojson.feature(us, us.objects.counties), {fill: (map => d => map.get(d.id))(new Map(data.map(d => [d.id, d.rate])))}),
    Plot.geo(topojson.mesh(us, us.objects.states, (a, b) => a !== b), {stroke: "white"})
 ]
})
```



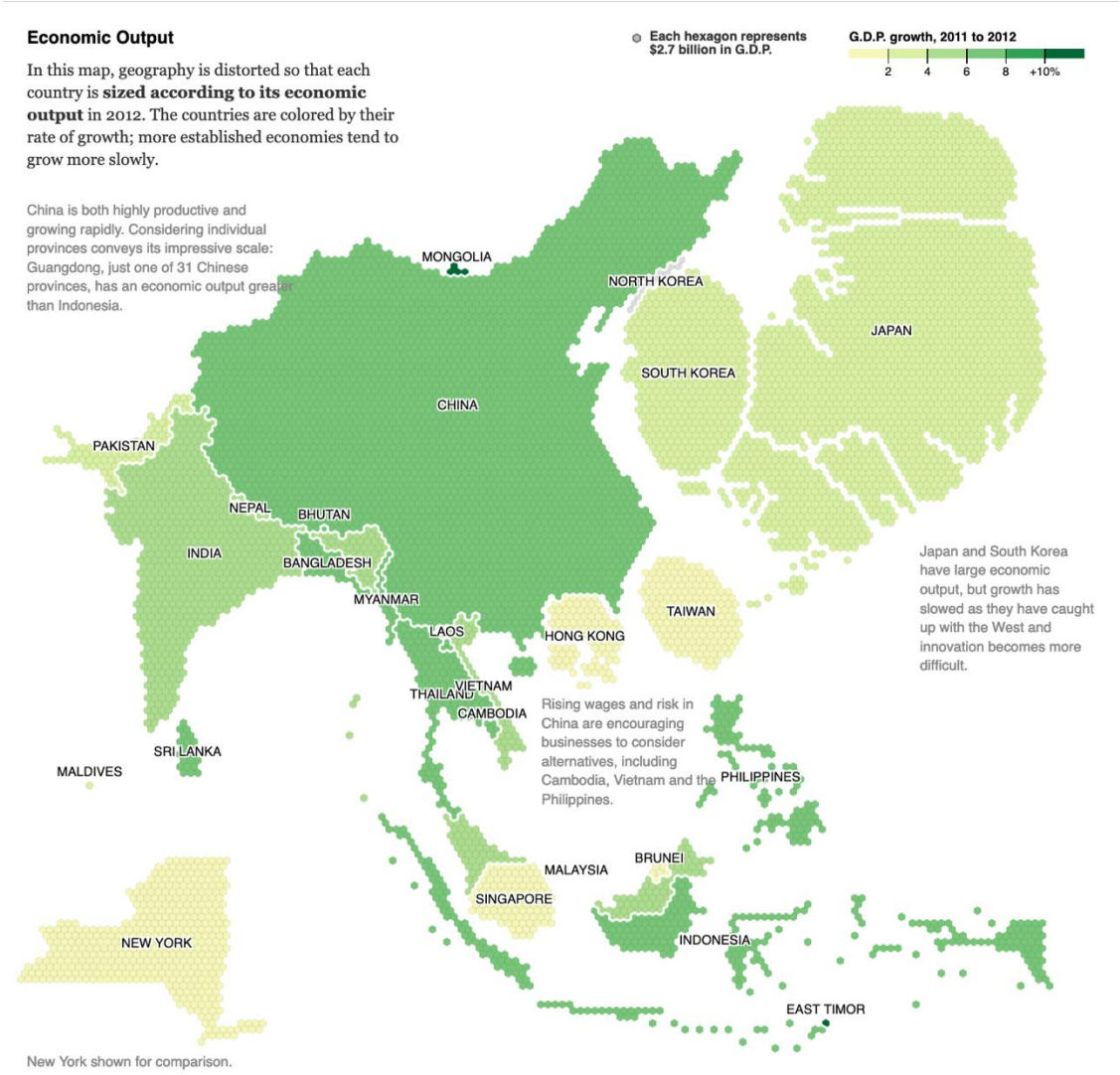
Choose colors with care

Focus on the foreground

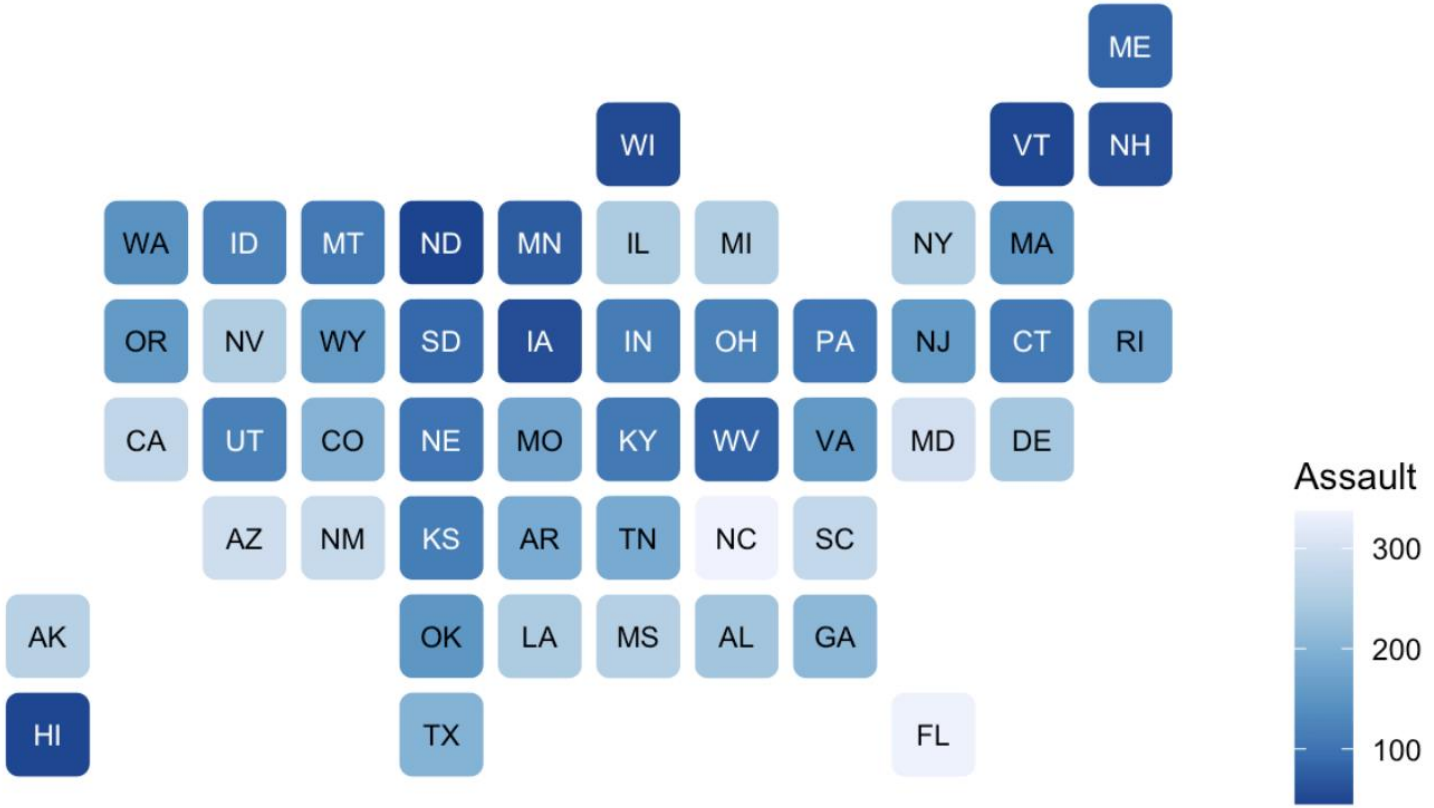
Cartogram


226
Kamala Harris
270
TO WIN
312
Donald J. Trump
74,109,194 votes (48.3%)
153.0 million votes so far (Estimated 98.7% counted)
76,683,848 votes (50.0%)
By winner
Electoral votes
Margin by county
Shift from 2020
Dem.
Rep.
Win Flip
Wash.
Ore.
Idaho
Mont.
Wyo.
N.D.
S.D.
Minn.
Iowa
Wis.
Mich.
Nev.
Utah
Colo.
Neb.
Kan.
Mo.
Ill.
Ind.
Ohio
Pa.
N.Y.
Vt.
N.H.
Maine
Mass.
Conn.
R.I.
N.J.
Md.
Del.
W.Va.
Ky.
Va.
D.C.
Calif.
Ariz.
N.M.
Okla.
Ark.
Tenn.
N.C.
S.C.
Texas
La.
Miss.
Ala.
Ga.
Fla.
Alaska
Hawaii

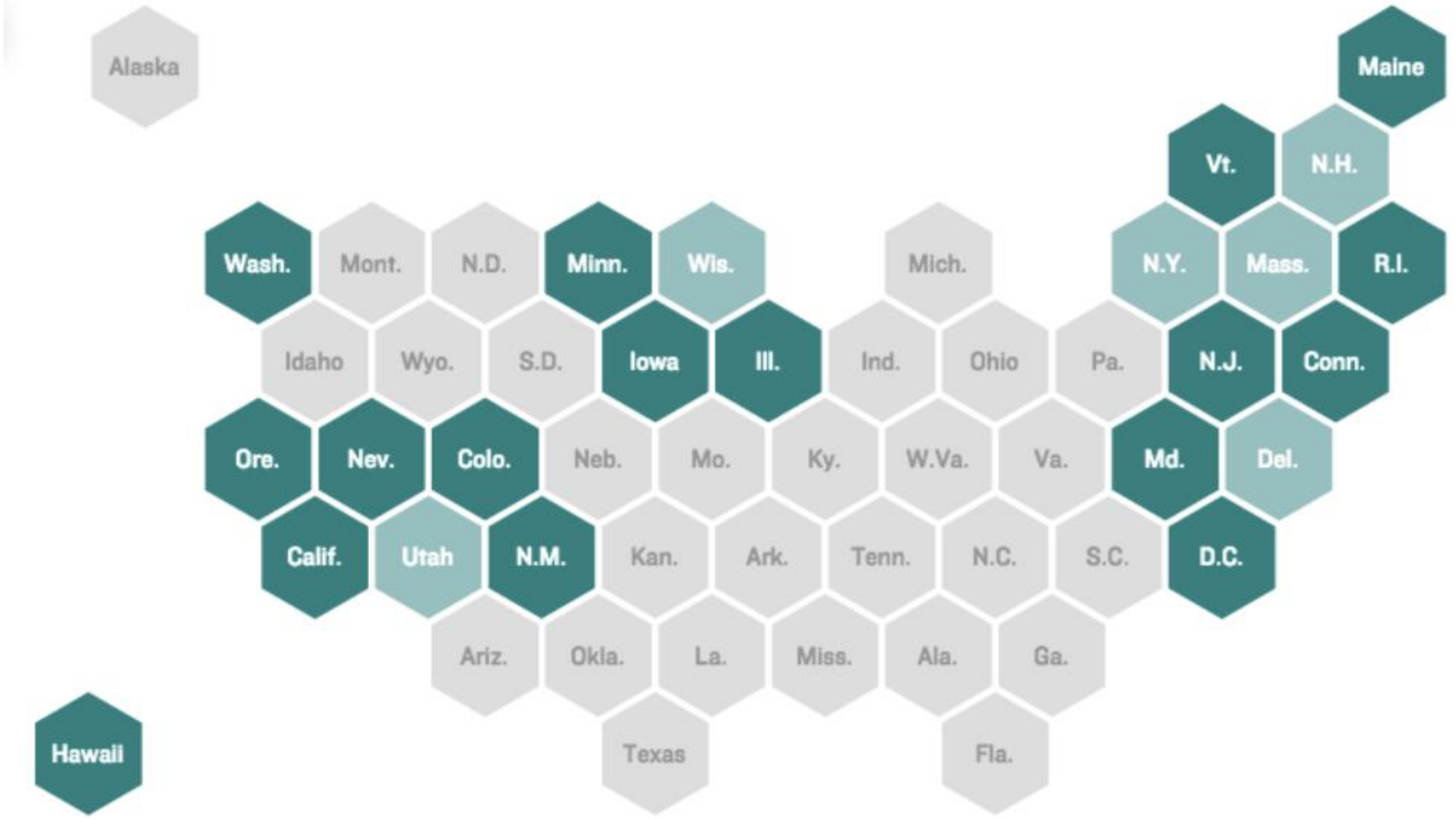
Cartogram


Economic Output
In this map, geography is distorted so that each country is sized according to its economic output in 2012. The countries are colored by their rate of growth; more established economies tend to grow more slowly.
Each hexagon represents $2.7 billion in G.D.P.
G.D.P. growth, 2011 to 2012
2
4
6
8
+10%
China is both highly productive and growing rapidly. Considering individual provinces conveys its impressive scale: Guangdong, just one of 31 Chinese provinces, has an economic output greater than Indonesia.
MONGOLIA
NORTH KOREA
JAPAN
SOUTH KOREA
CHINA
PAKISTAN
NEPAL
BHUTAN
INDIA
BANGLADESH
MYANMAR
LAOS
TAIWAN
HONG KONG
VIETNAM
THAILAND
CAMBODIA
Japan and South Korea have large economic output, but growth has slowed as they have caught up with the West and innovation becomes more difficult.
Rising wages and risk in China are encouraging businesses to consider alternatives, including Cambodia, Vietnam and the Philippines.
SRI LANKA
MALDIVES
PHILIPPINES
BRUNEI
MALAYSIA
SINGAPORE
INDONESIA
NEW YORK
EAST TIMOR
New York shown for comparison.

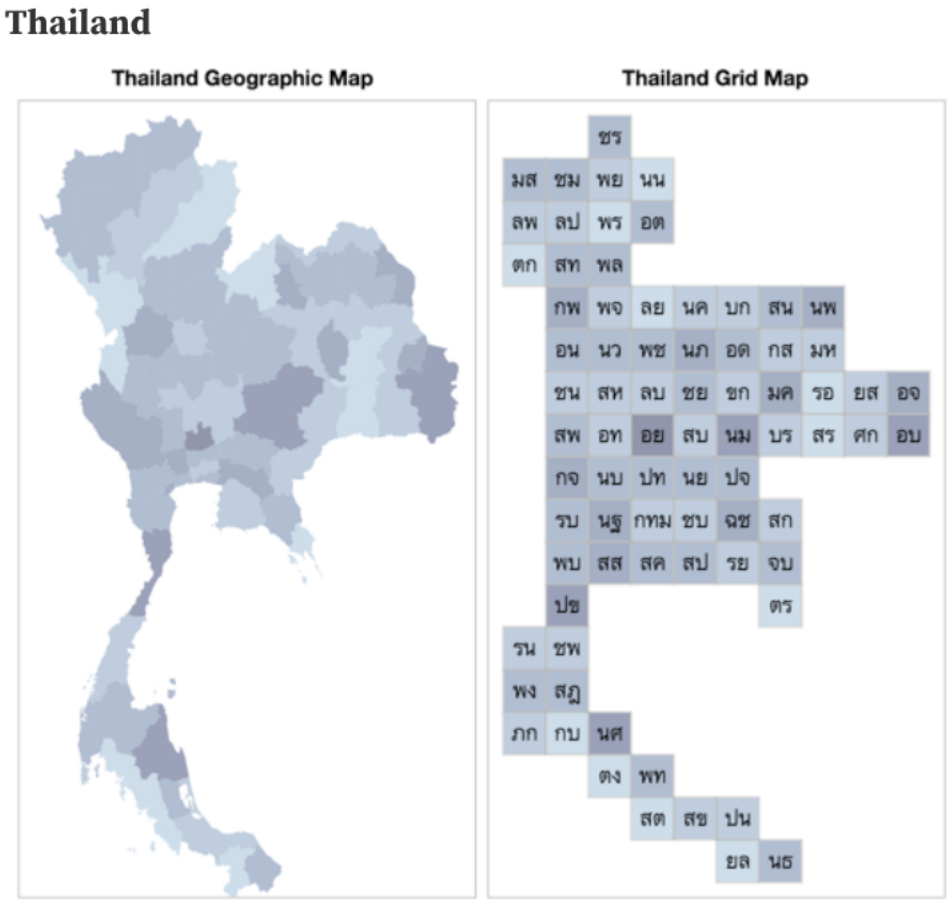
Cartogram


ME
WI
VT
NH
WA
ID
MT
ND
MN
IL
MI
NY
MA
OR
NV
WY
SD
IA
IN
OH
PA
NJ
CT
RI
CA
UT
CO
NE
MO
KY
WV
VA
MD
DE
AZ
NM
KS
AR
TN
NC
SC
AK
OK
LA
MS
AL
GA
HI
TX
FL
Assault
300
200
100

Cartogram


Alaska
Maine
Vt.
N.H.
Wash.
Mont.
N.D.
Minn.
Wis.
Mich.
N.Y.
Mass.
R.I.
Idaho
Wyo.
S.D.
Iowa
Ill.
Ind.
Ohio
Pa.
N.J.
Conn.
Ore.
Nev.
Colo.
Neb.
Mo.
Ky.
W.Va.
Va.
Md.
Del.
Calif.
Utah
N.M.
Kan.
Ark.
Tenn.
N.C.
S.C.
D.C.
Ariz.
Okla.
La.
Miss.
Ala.
Ga.
Hawaii
Texas
Fla.

Cartogram


Thailand
Thailand Geographic Map
Thailand Grid Map
ชร
มส ชม พย นน
ลพ ลป พร อต
ตก สท พล
กพ พจ ลย นค บก สน นพ
อน นว พช นภ อด กส มห
ชน สห ลบ ชย ชก มค รอ ยส อจ
สพ อท อย สบ นม บร สร ศก อบ
กจ นบ ปท นย ปจ
รบ นฐ กทม ชบ ฉช สก
พบ สส สค สป รย จบ
ปช ตร
รน ชพ
พง สฎ
ภก กบ นศ
ตง พท
สต สช ปน
ยล นธ

Cartogram

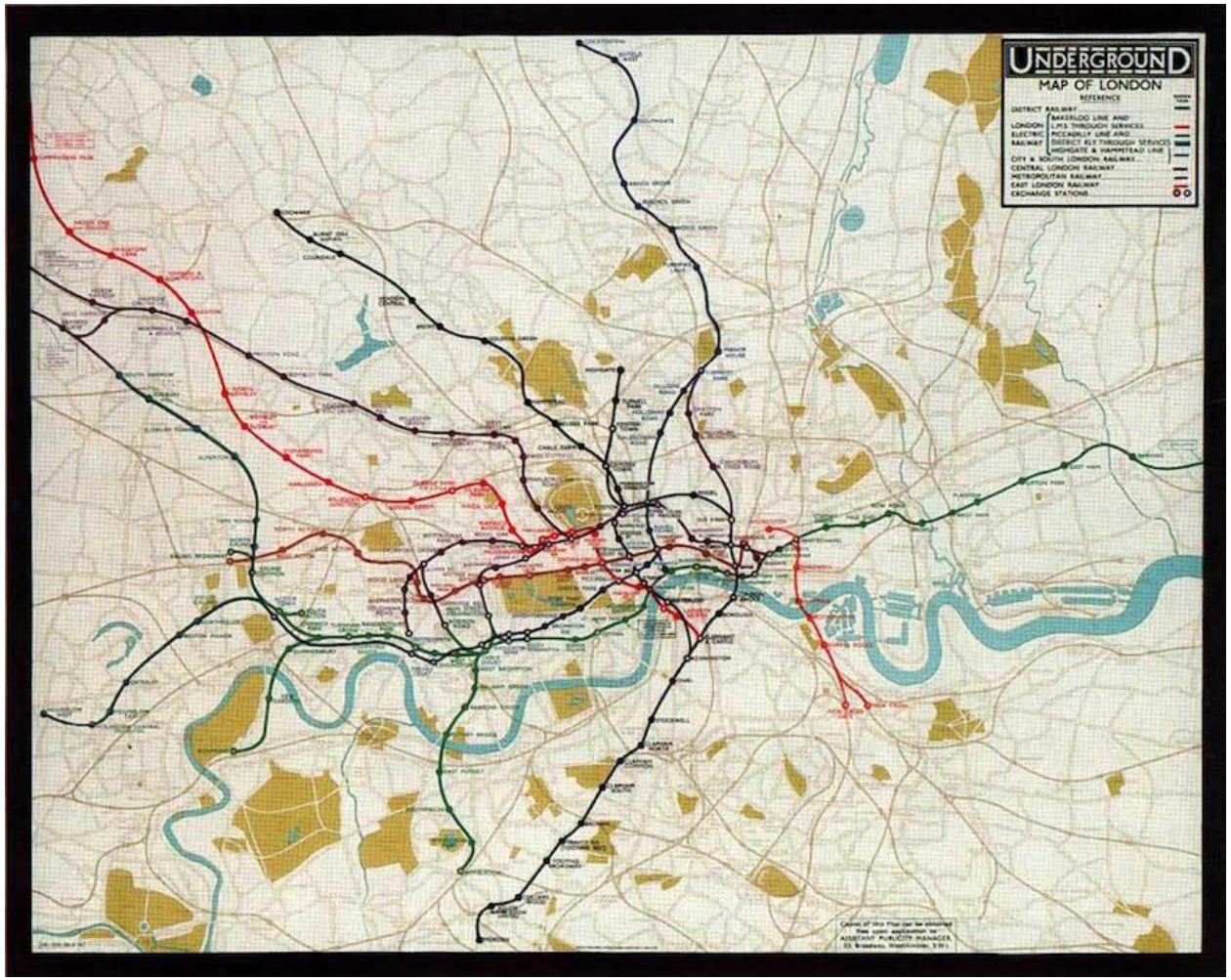

France's Ageing Population
The number of people aged 75 and over in France has risen from 4% in 1975 to 10% in 2015.
See below how this has changed in each of France's numbered departments
FRANCE TOTAL
Percent aged 75+
15%
10%
5%
0%
1980
1990
2000
2010
62
59
14
76
80
60
02
08
50
61
27
78
95
77
51
55
54
57
29
22
35
53
72
92
75
93
10
52
88
67
56
44
49
28
91
94
89
21
70
90
68
85
79
37
41
45
58
71
39
25
86
36
18
03
42
69
01
74
17
16
87
23
63
43
38
73
33
24
46
19
15
07
26
05
40
47
82
12
48
30
84
04
06
64
32
31
81
11
34
13
83
65
09
66
2A
2B

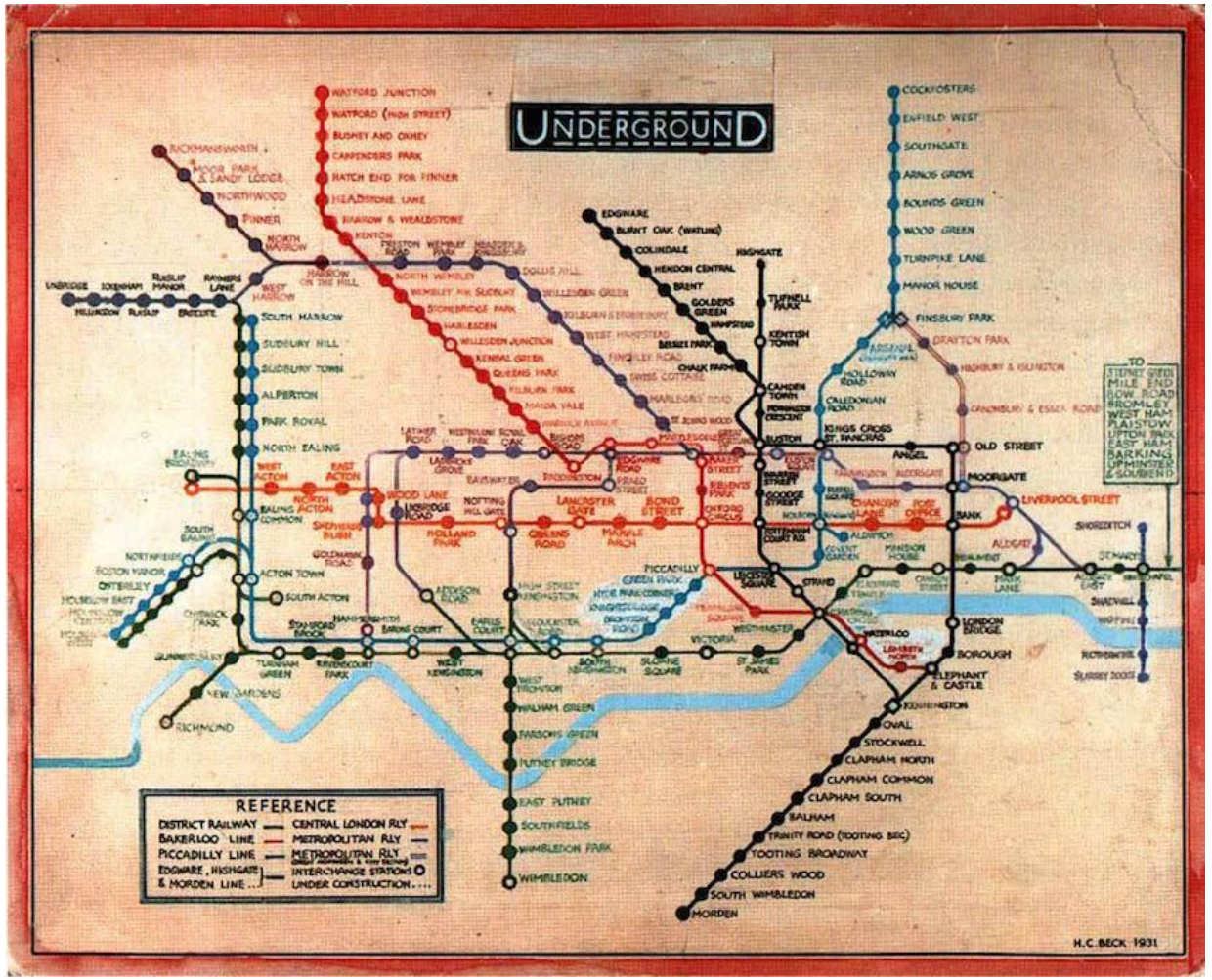
Cartogram for graph

General abstraction

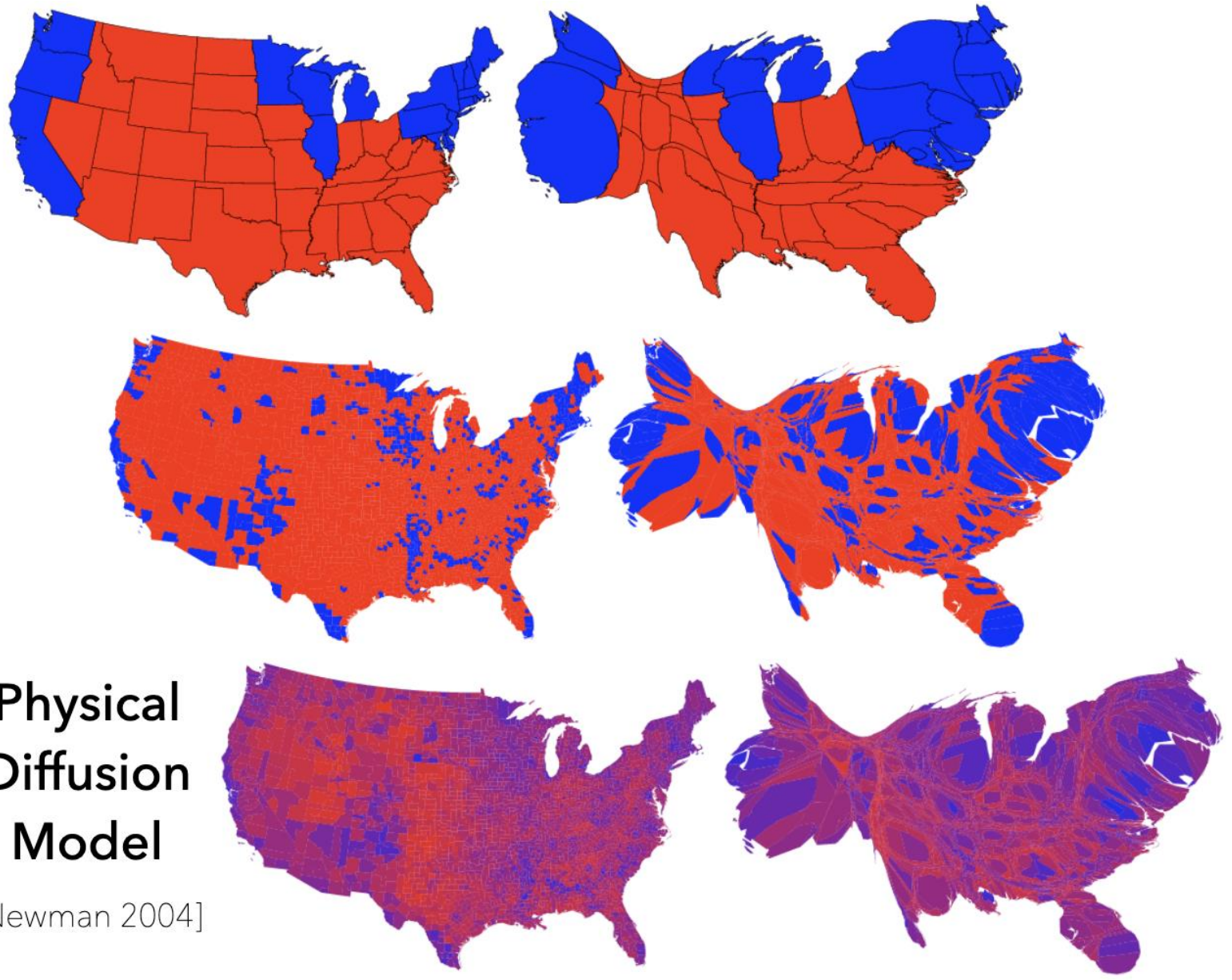
General abstraction

Abstraction





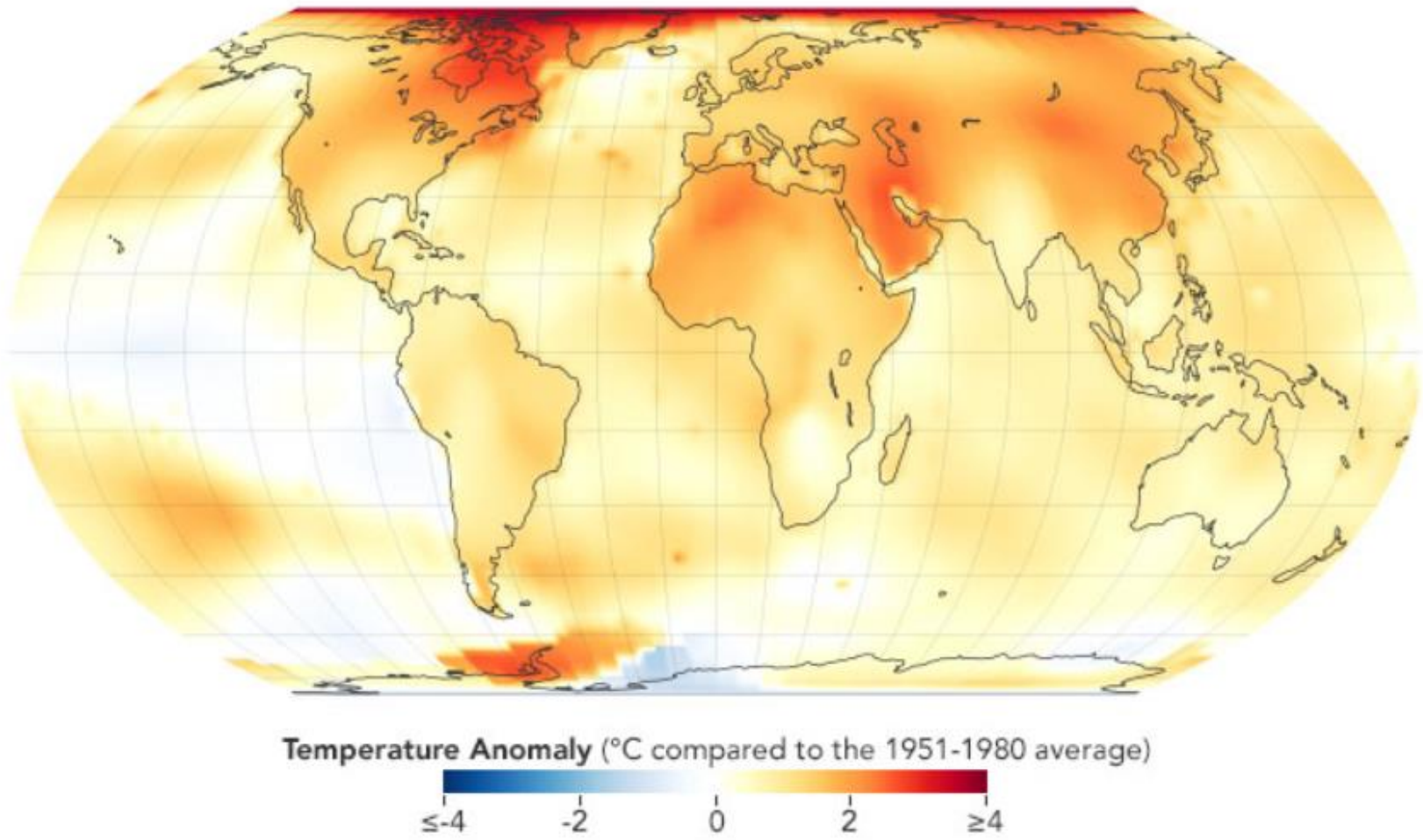
Cartogram



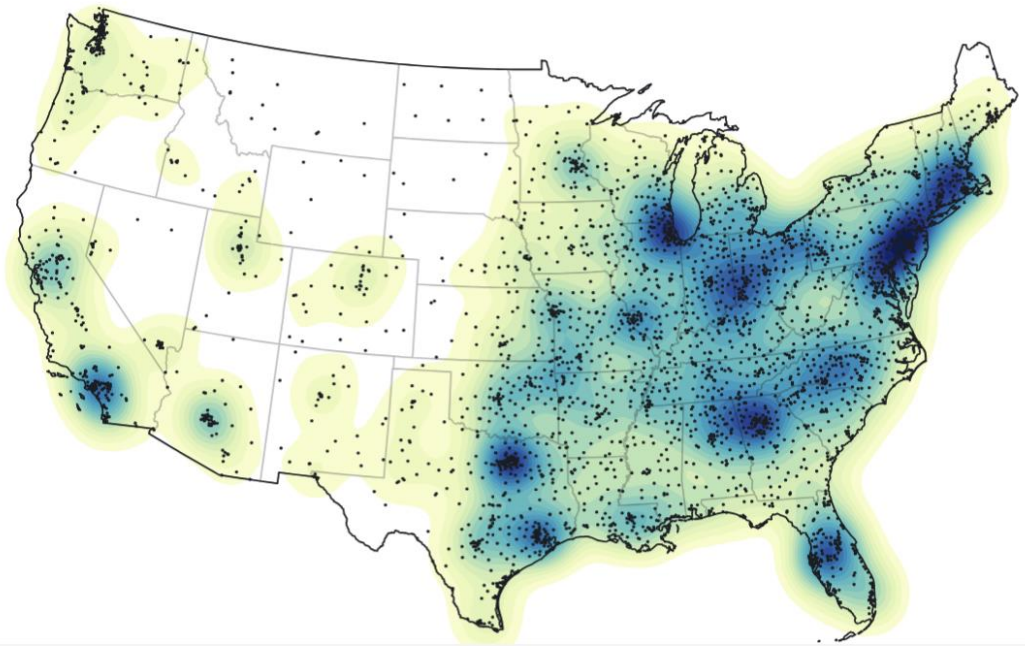
Physical Diffusion Model

[Newman 2004]

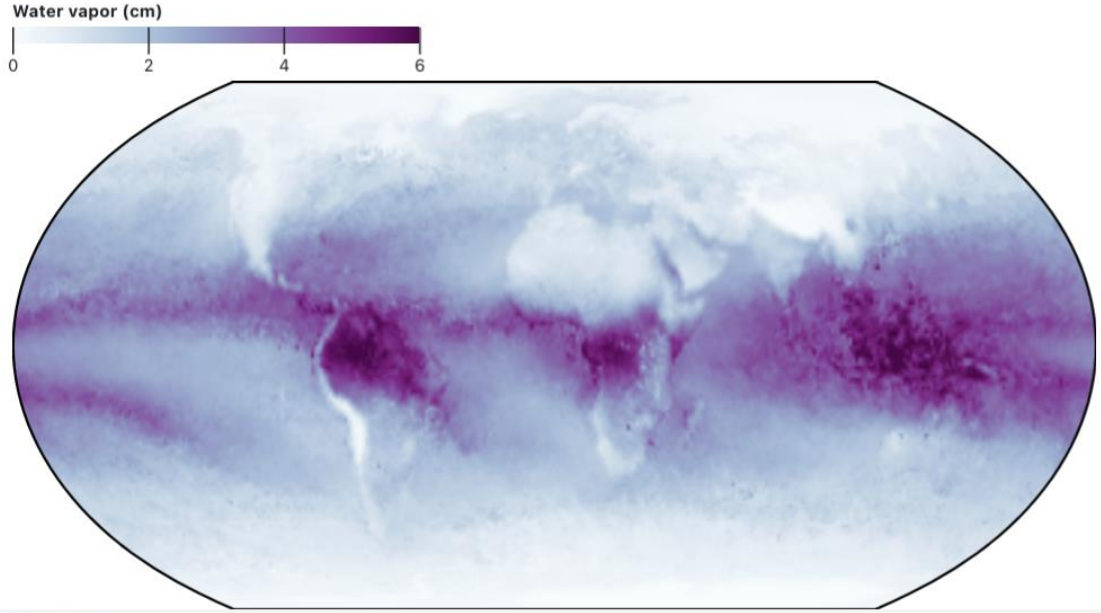
Heatmaps


Temperature Anomaly (°C compared to the 1951-1980 average)
≤-4
-2
0
2
≥4

Heatmaps



```
Plot.plot({
  projection: "albers",
  color: {scheme: "YlGnBu"},
  style: "overflow: visible;",
  marks: [
    Plot.density(walmarts, {x: "longitude", y: "latitude", bandwidth: 10, fill: "density"}),
    Plot.geo(statemesh, {strokeOpacity: 0.3}),
    Plot.geo(nation),
    Plot.dot(walmarts, {x: "longitude", y: "latitude", r: 1, fill: "currentColor"})
  ]
})
```



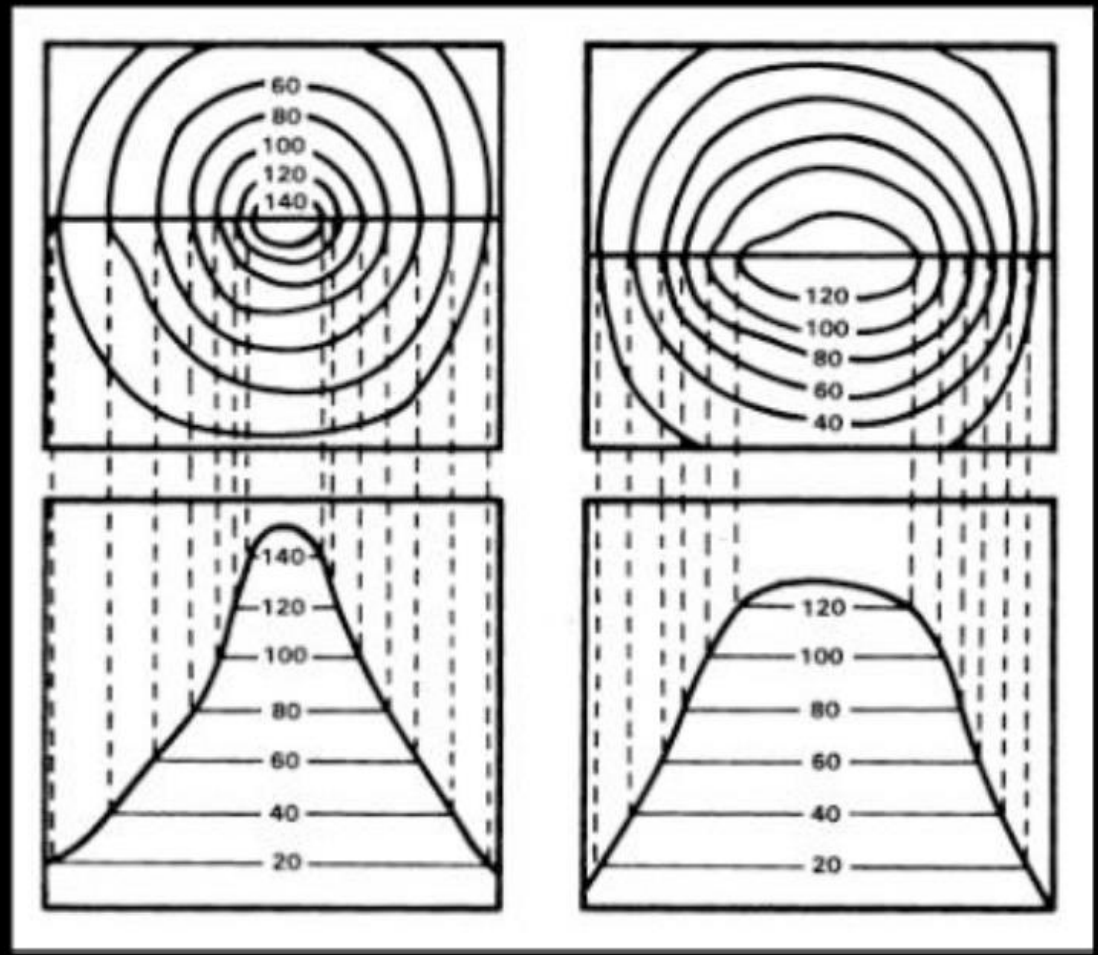

```
Plot.plot({
  projection: "equal-earth",
  color: {
    scheme: "BuPu",
    domain: [0, 6],
    legend: true,
    label: "Water vapor (cm)"
  },
  marks: [
    Plot.raster(vapor, {
      fill: Plot.identity,
      width: 360,
      height: 180,
      x1: -180,
      y1: 90,
      x2: 180,
      y2: -90,
      interpolate: "barycentric",
      clip: "sphere"
    }),
    Plot.sphere({stroke: "black"})
  ]
})
```

Break data into buckets

CRIMESPOTTING_

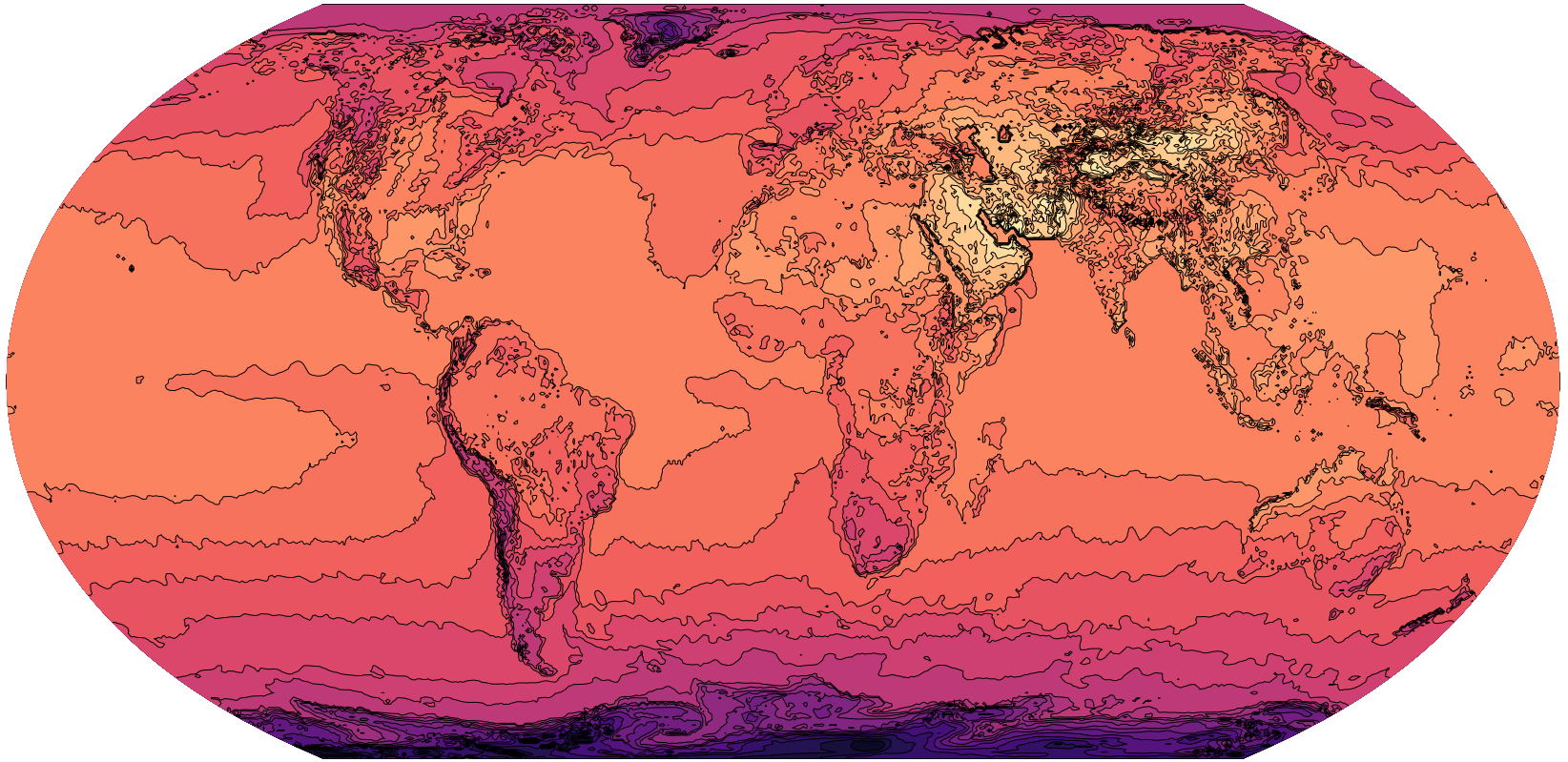
The brazen 2007 murder of journalist Chauncey Bailey in Oakland, California, led Stamen partner Mike Migurski to make the city's crime data more accessible. This heat map of downtown uses data from CrimeWatch, a community website, to show the gaps between crimes at a given intersection: white is high-crime; darker areas are safe. stamen.com

Meaningful buckets, isolines

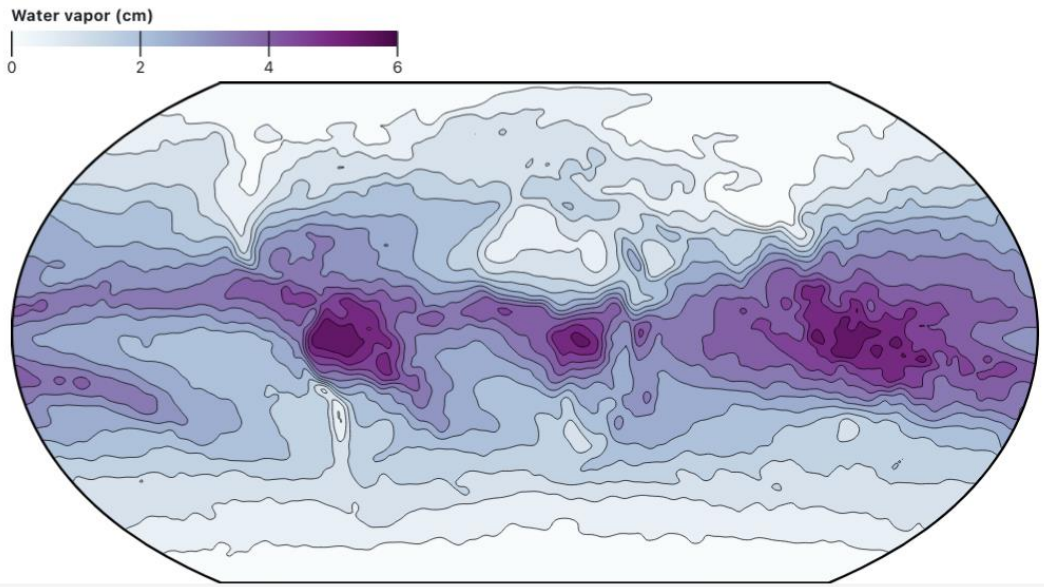


"Iso" means "same"
Isolines for elevation

Contour plots

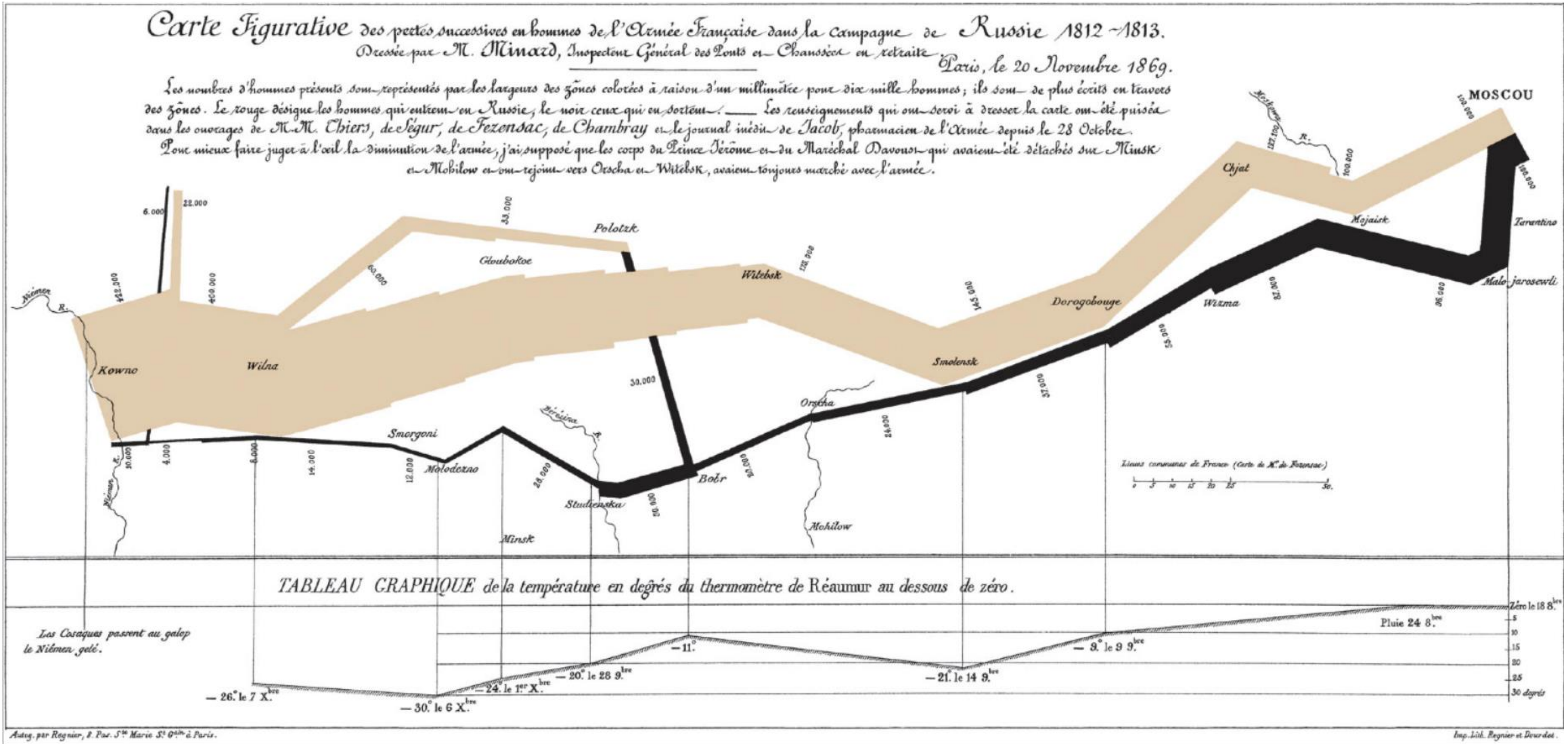


Contour plots




```
Plot.plot({
  projection: "equal-earth",
  color: {
    scheme: "BuPu",
    domain: [0, 6],
    legend: true,
    label: "Water vapor (cm)"
  },
  marks: [
    Plot.contour(vapor, {
      fill: Plot.identity,
      width: 360,
      height: 180,
      x1: -180,
      y1: 90,
      x2: 180,
      y2: -90,
      blur: 1,
      stroke: "black",
      strokeWidth: 0.5,
      clip: "sphere"
    }),
    Plot.sphere({stroke: "black"})
  ]
})
```

Flow map

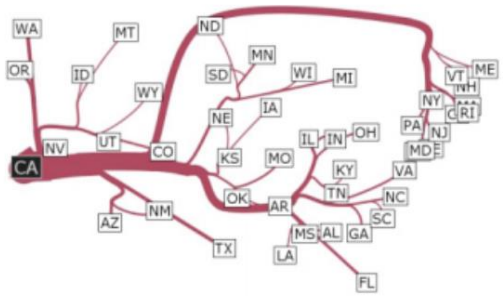

Carte Figurative des pertes successives en hommes de l'Armée Française dans la campagne de Russie 1812–1813.
Dressée par M. Minard, Inspecteur Général des Ponts et Chaussées en retraite.
Paris, le 20 Novembre 1869.
Les nombres d'hommes présents sont représentés par les largeurs des zônes colorées à raison d'un millimètre pour dix mille hommes; ils sont de plus écrits en travers des zônes. Le rouge désigne les hommes qui entrent en Russie, le noir ceux qui en sortent. —— Les renseignements qui ont servi à dresser la carte ont été puisés dans les ouvrages de M.M. Thiers, de Ségur, de Fezensac, de Chambray et le journal inédit de Jacob, pharmacien de l'Armée depuis le 28 Octobre.
Pour mieux faire juger à l'œil la diminution de l'armée, j'ai supposé que les corps du Prince Jérôme et du Maréchal Davoust qui avaient été détachés sur Minsk et Mohilow et ont rejoint vers Orscha et Witebsk, avaient toujours marché avec l'armée.
MOSCOU
Chjat
Mojaisk
Tarantino
Malo-jarosewli
Wizma
Dorogobouge
Smolensk
Orscha
Mohilow
Witebsk
Polotzk
Gloubokoe
Wilna
Kowno
Smorgoni
Molodezno
Studienska
Minsk
Botr
TABLEAU GRAPHIQUE de la température en degrés du thermomètre de Réaumur au dessous de zéro.
Les Cosaques passent au galop le Niémen gelé.
Zéro le 18 8.bre
Pluie 24 8.bre
— 9.° le 9 9.bre
— 21.° le 14 9.bre
— 11.°
— 20.° le 28 9.bre
— 24.° le 1.er X.bre
— 30.° le 6 X.bre
— 26.° le 7 X.bre
30 degrés

Flow map

Migration from California, ‘95-‘00

Tobler 1987

Phan et al. 2005



Verbeek et al. 2011

Cui et al. 2008

Holten & van Wijk 2009

Flow Map

Nov. 20, 2024
1:12 pm EST
(time of forecast download)

top speed: **41.7 mph**
average: **11.1 mph**