



UNIVERSITY OF SAN FRANCISCO  
CHANGE THE WORLD FROM HERE

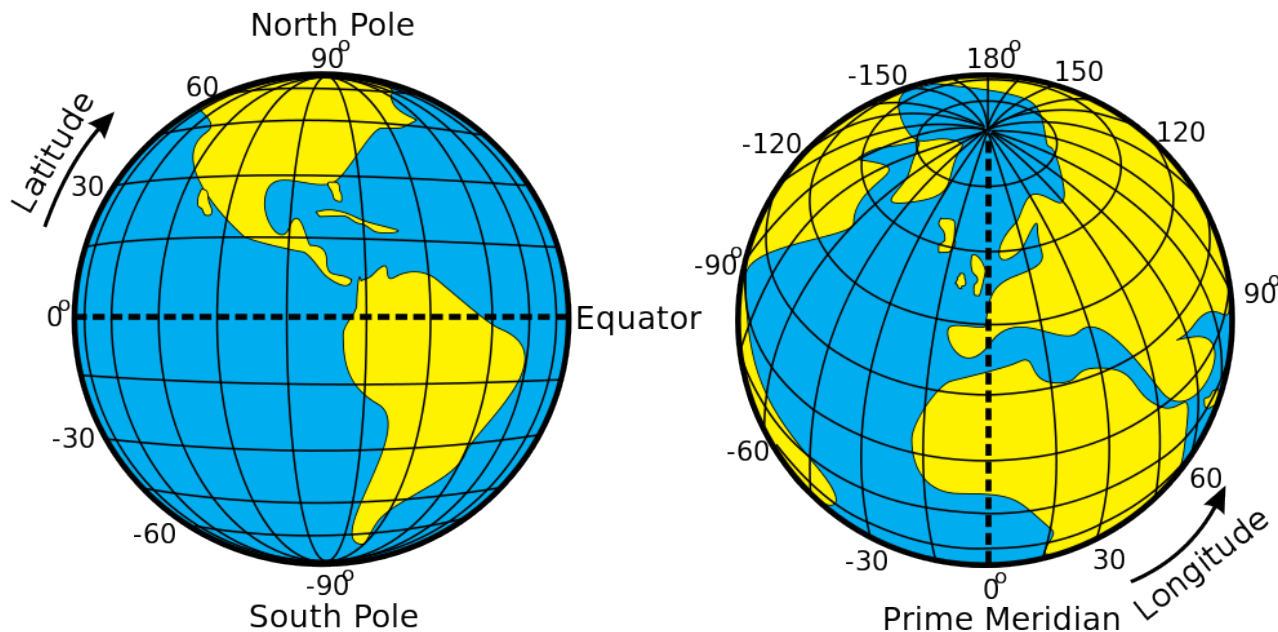
# Geospatial Data

Data Formats and Map Projections

# GEOSPATIAL DATA



# Latitude and Longitude



[http://en.wikipedia.org/wiki/File:Latitude\\_and\\_Longitude\\_of\\_the\\_Earth.svg](http://en.wikipedia.org/wiki/File:Latitude_and_Longitude_of_the_Earth.svg)

# Latitude and Longitude

- **DMS (Degree, Minutes, Seconds):**

- Latitude:  $40^{\circ}26'47''\text{N}$
- Longitude:  $79^{\circ}58'36''\text{W}$

- **Decimal Degree:**

- Latitude: 40.446195
- Longitude: -79.948862

[http://en.wikipedia.org/wiki/Geographic\\_coordinate\\_conversion](http://en.wikipedia.org/wiki/Geographic_coordinate_conversion)

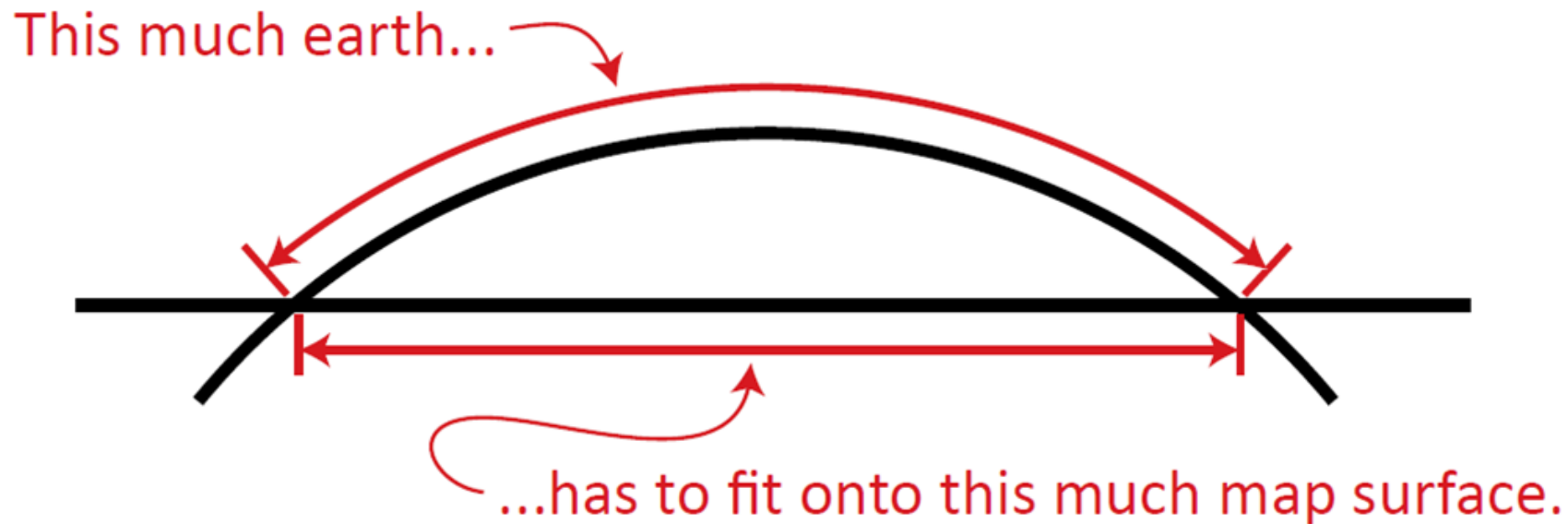


# MAP PROJECTIONS



# Map Projections

- World is 3D, maps are 2D
  - Earth is a *ellipsoid(ish)*, not a perfect *sphere*
- Must project 3D coordinates onto 2D surface
  - True of rendering any 3D object in 2D
- All projections cause some sort of distortion



<http://mjfooster83.github.io/projections/index.html#/38>

# Projection Distortion

- **Distance** between two points
- **Direction** between two points
- **Shape** of regions
- **Area** of regions





# Types of Projections

- **Azimuthal**
  - Preserves the azimuth (direction) from center
- **Conformal**
  - Local angles are correct, preserving small shapes
- **Equal-Area**
  - All regions have correct area
- **Equidistant**
  - Distances from center (or along certain lines, like along meridians) are correct

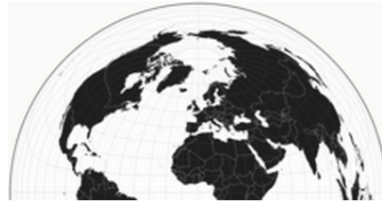
<http://egsc.usgs.gov/isb/pubs/MapProjections/projections.html>



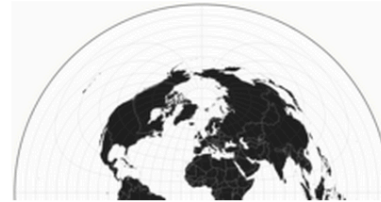
d3.geo.albersUsa



d3.geo.azimuthalEqualArea



d3.geo.azimuthalEquidistant



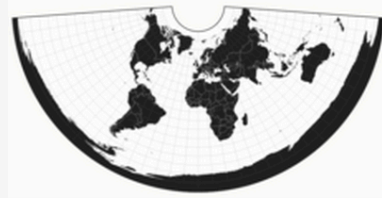
d3.geo.conicEqualArea



d3.geo.conicConformal



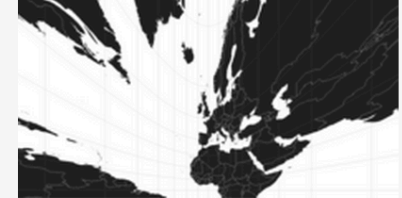
d3.geo.conicEquidistant



d3.geo.equirectangular



d3.geo.gnomonic



d3.geo.mercator



d3.geo.orthographic



d3.geo.stereographic



d3.geo.transverseMercator



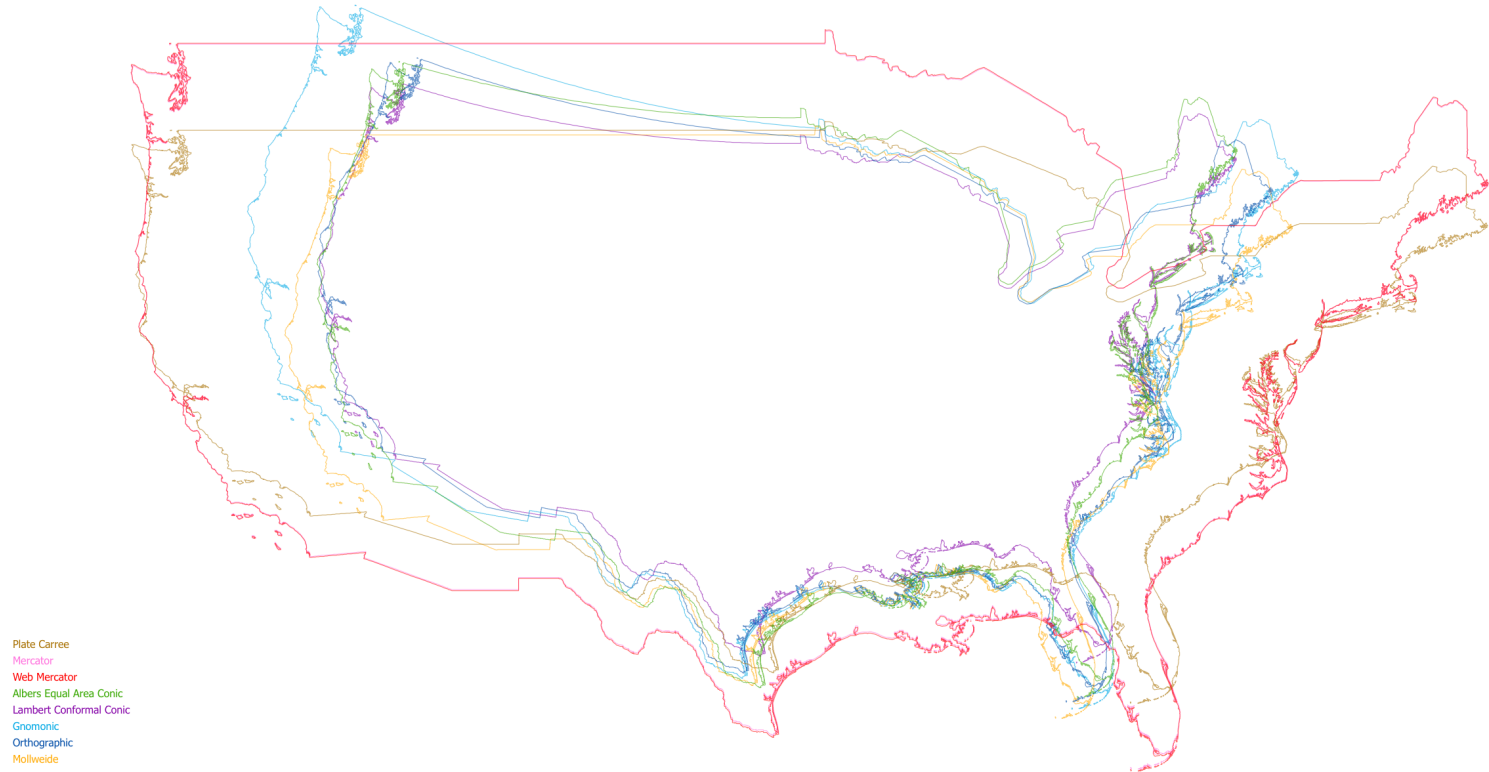
<https://github.com/mbostock/d3/wiki/Geo-Projections>



<https://github.com/mbostock/d3/wiki/Geo-Projections>

## Continental USA in Eight Projections

All shapes drawn at the same scale, with the same center



<http://sandbox.azavea.com/projection-overlays/contiguous-usa.html>

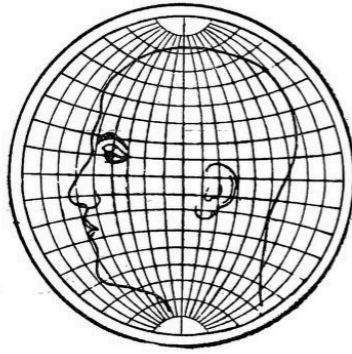


FIG. 42.—Man's head drawn on globular projection.

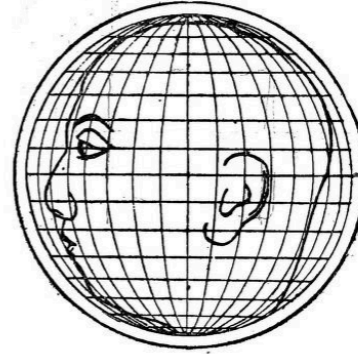


FIG. 43.—Man's head plotted on orthographic projection.

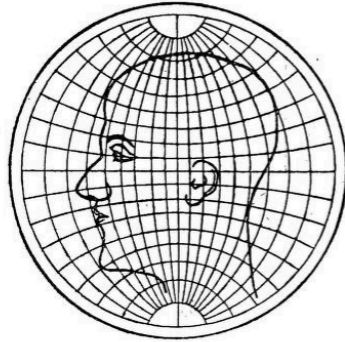


FIG. 44.—Man's head plotted on stereographic projection.

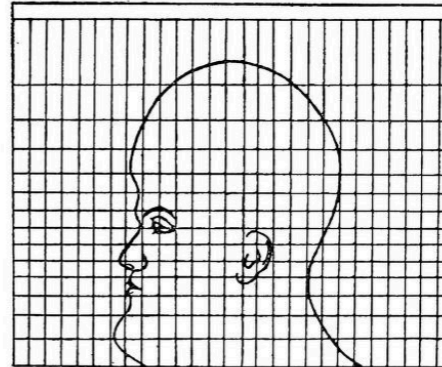
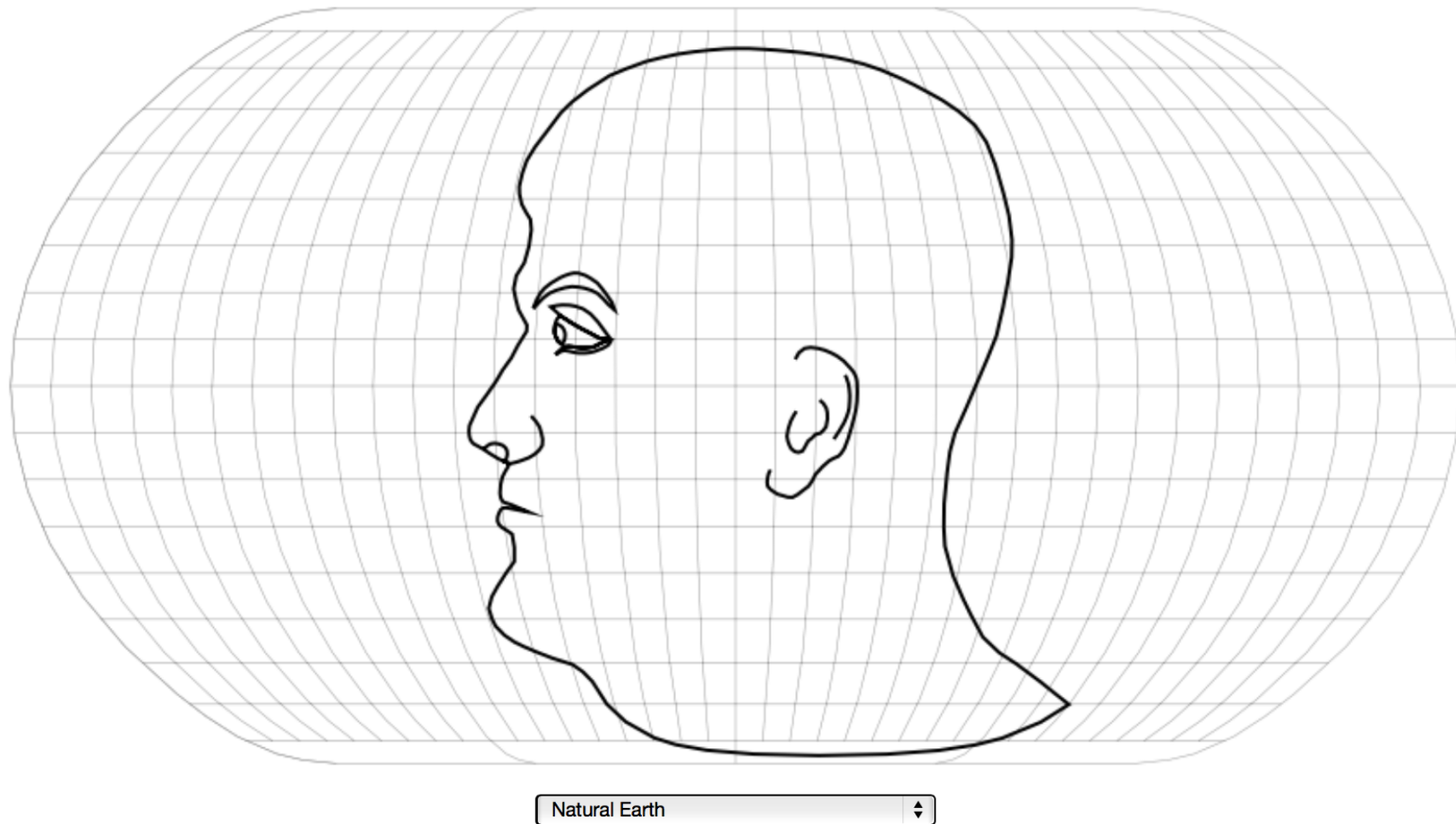
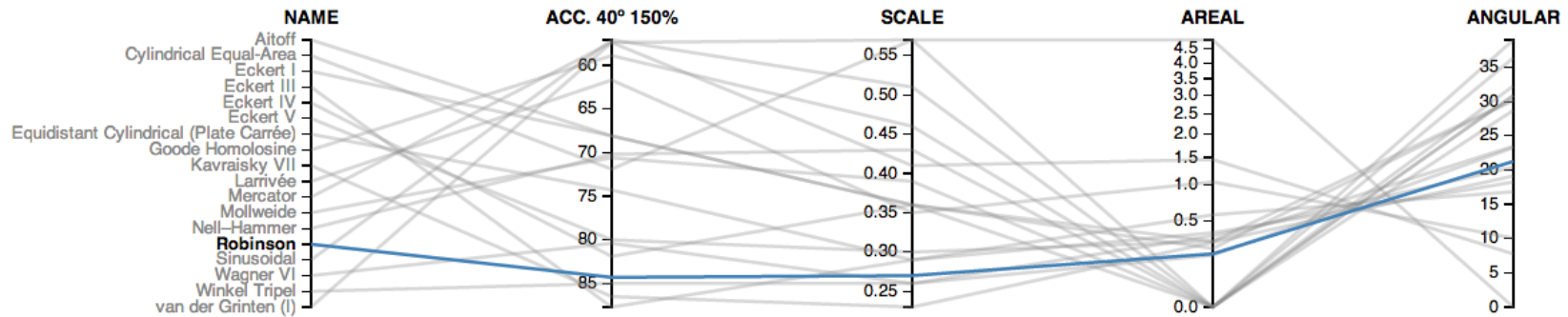
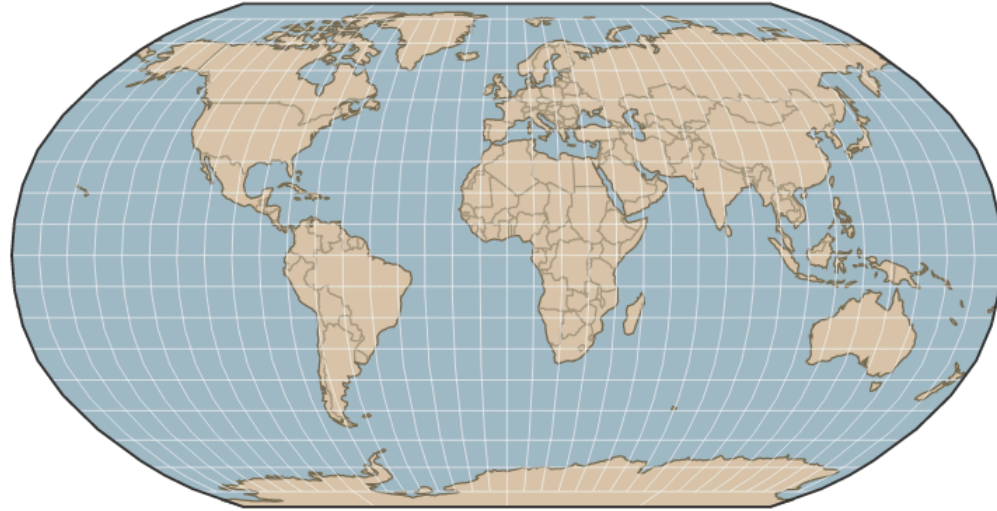


FIG. 45.—Man's head plotted on Mercator projection.

<http://flowingdata.com/2014/01/13/map-projections-illustrated-with-a-face/>



<http://bl.ocks.org/awoodruff/9216081>



<http://bl.ocks.org/syntagmatic/3711245>



# Mercator Projection

$$x = \lambda - \lambda_0$$

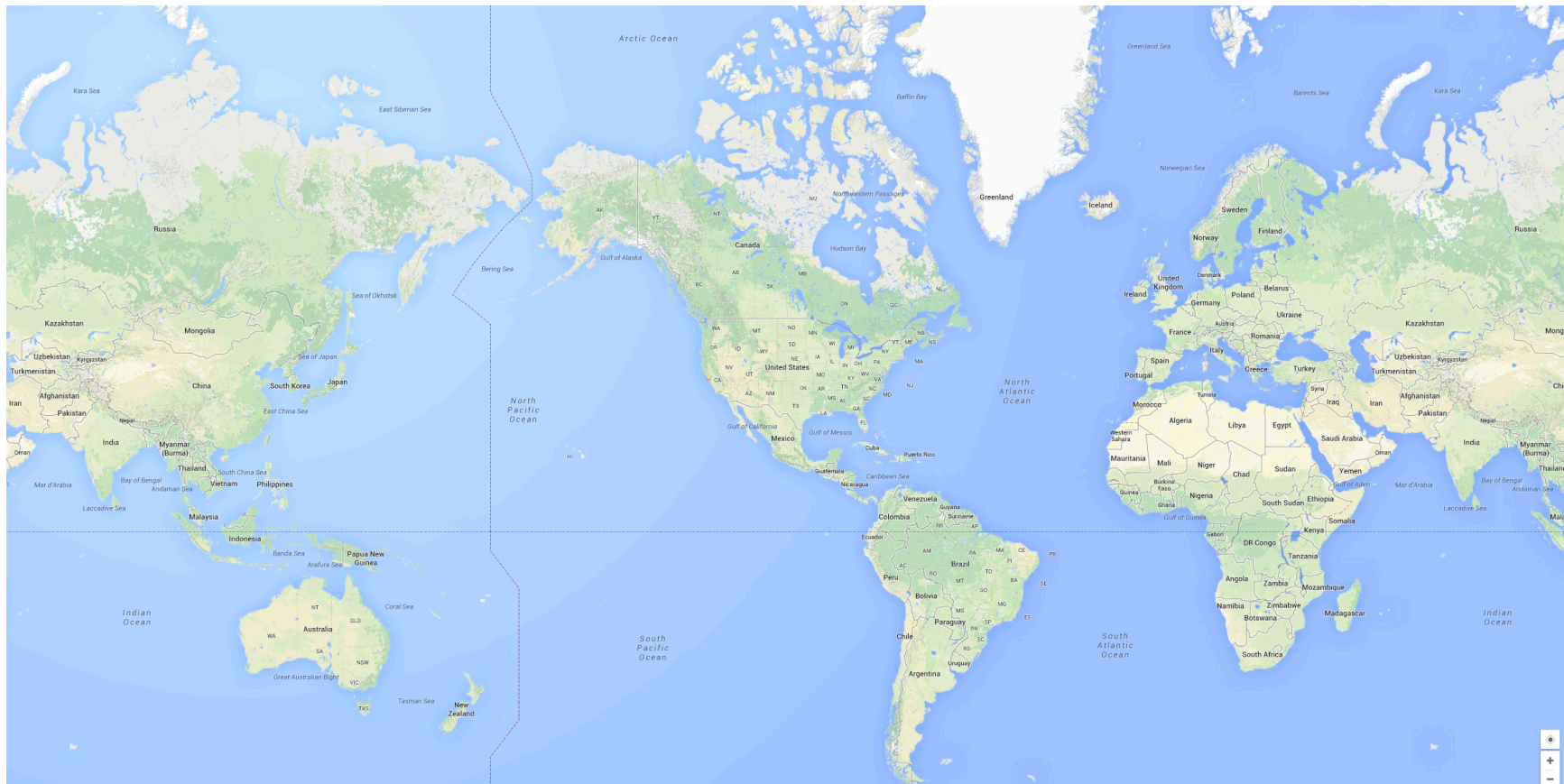
$$y = \ln (\tan \phi + \sec \phi)$$

where  $\lambda$  is the **longitude** and  $\phi$  is the **latitude**, and  $\lambda_0$  is the center longitude of the projection

<http://mathworld.wolfram.com/MercatorProjection.html>







<https://www.google.com/maps/>

# GEOSPATIAL DATA FORMATS



# Shapefiles

- Used by GIS software in early 1990s
- Geometry given as points, polylines, or polygons
- Consists of several files
  - \*.shp contains geometry information
  - \*.shx contains index information
  - \*.dbf contains attribute information
- Specified at different resolutions
  - 1:10m (state-level), 1:110m (world-level)

<http://wiki.openstreetmap.org/wiki/Shapefiles>



# GeoJSON

- JSON-based standard for geospatial data on web
- May include a geometry, feature, or collection of features
- Geometry types include Point, LineString, Polygon, MultiPoint, MultiLineString, MultiPolygon, and GeometryCollection
- Can add additional data in JSON format

<http://www.geojson.org/geojson-spec.html>



# GeoJSON

```
1  { "type": "FeatureCollection",
2    "features": [
3      { "type": "Feature",
4        "geometry": {"type": "Point", "coordinates": [102.0, 0.5]},
5        "properties": {"prop0": "value0"}
6      },
7      { "type": "Feature",
8        "geometry": {
9          "type": "LineString",
10         "coordinates": [
11           [102.0, 0.0], [103.0, 1.0], [104.0, 0.0], [105.0, 1.0]
12         ]
13       },
14       "properties": {
15         "prop0": "value0",
16         "prop1": 0.0
17       }
18     ],
```

<http://www.geojson.org/geojson-spec.html>



# Shapefiles to GeoJSON

- Shapefiles too large to load in web applications
- Most data provided as shapefiles, must convert
- Use MapShaper to simplify files
  - <http://www.mapshaper.org/>
- Use GDAL and ogr2ogr to convert
  - <http://www.gdal.org/>
  - <http://ogre.adc4gis.com/>
- See steps at <http://bost.ocks.org/mike/map/>



# TopoJSON

- Extension of GeoJSON that provides smaller files
- Details at <https://github.com/mbostock/topojson>
- Works well with D3 (created by same person)

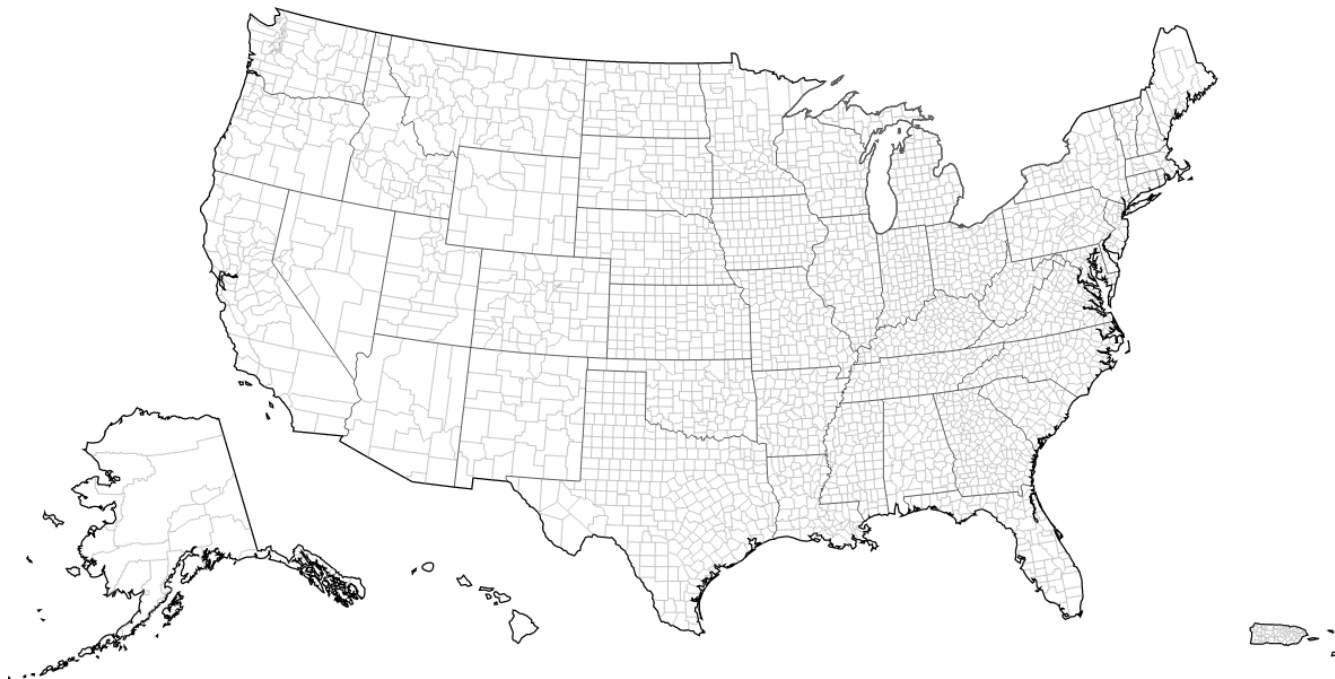


# RESOURCES



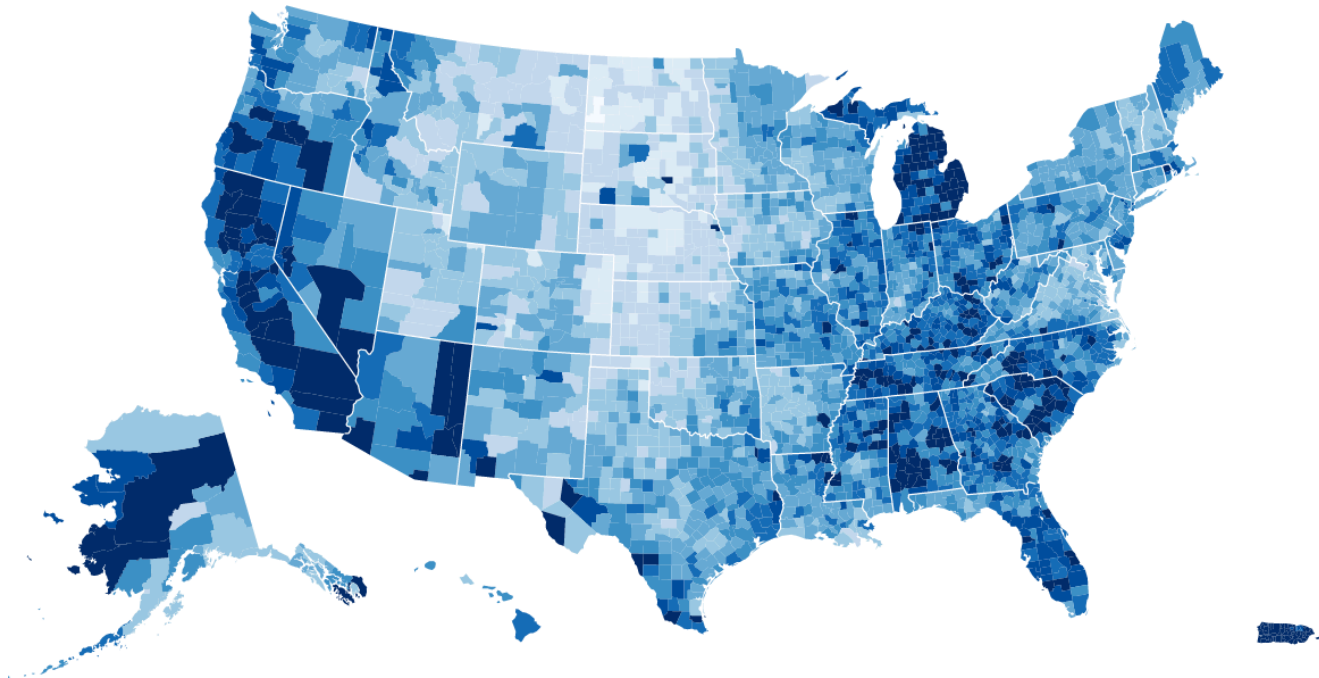


# U.S. TopoJSON



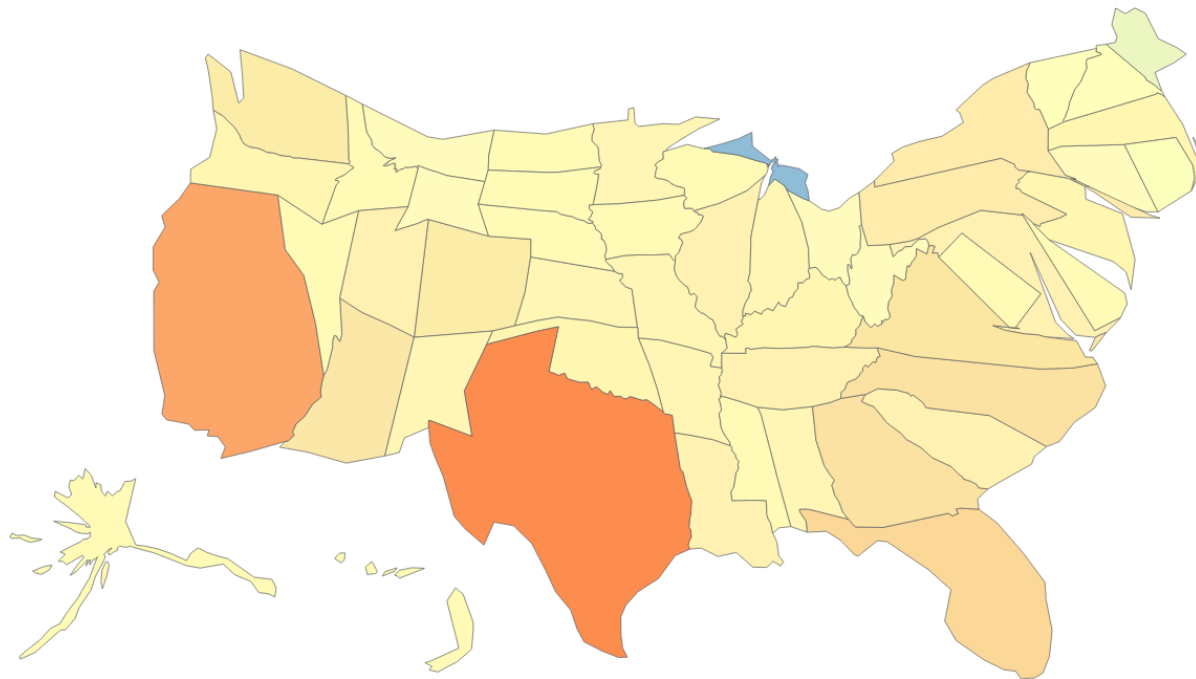
<http://bl.ocks.org/mbostock/4108203>

# D3/TopoJSON Choropleth



<http://bl.ocks.org/mbostock/4060606>

# D3/TopoJSON Cartogram



<http://prag.ma/code/d3-cartogram/>

# Other JavaScript Libraries

- Kartograph.js <http://kartograph.org/>
- Leaflet <http://leafletjs.com/>
- Polymaps <http://polymaps.org/>
- Modest Maps <http://modestmaps.com/>
- Google Maps JavaScript API  
<https://developers.google.com/maps/documentation/javascript/tutorial>



# Data Sources

- TopoJSON  
<https://github.com/mbostock/topojson/>  
<https://gist.github.com/mbostock/4090846>
- Earthquake GeoJSON Data Files  
<http://earthquake.usgs.gov/earthquakes/feed/>
- Geospatial Data.gov  
<http://geo.data.gov/geoportal/>
- Natural Earth  
<http://www.naturalearthdata.com/>



# QUESTIONS

<http://sjengle.cs.usfca.edu/>

