Unraveling Climate Change through Data Visualization

Jorge Martínez-Rey
graphicprototype.net

DSPT Day
October 26th, 2019

The Problem

Total Population

7.7 Bn

United Nations

The Problem

Science

The Opportunity

People

Access to Internet

4.3 Bn

International Telecommunication Union

Total Population

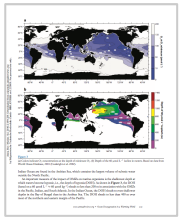
7.7 Bn

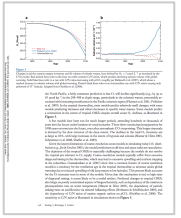
United Nations

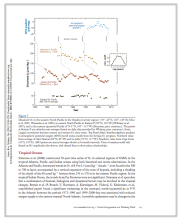
The Problem

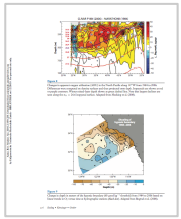
Science

Dataviz on scientific papers





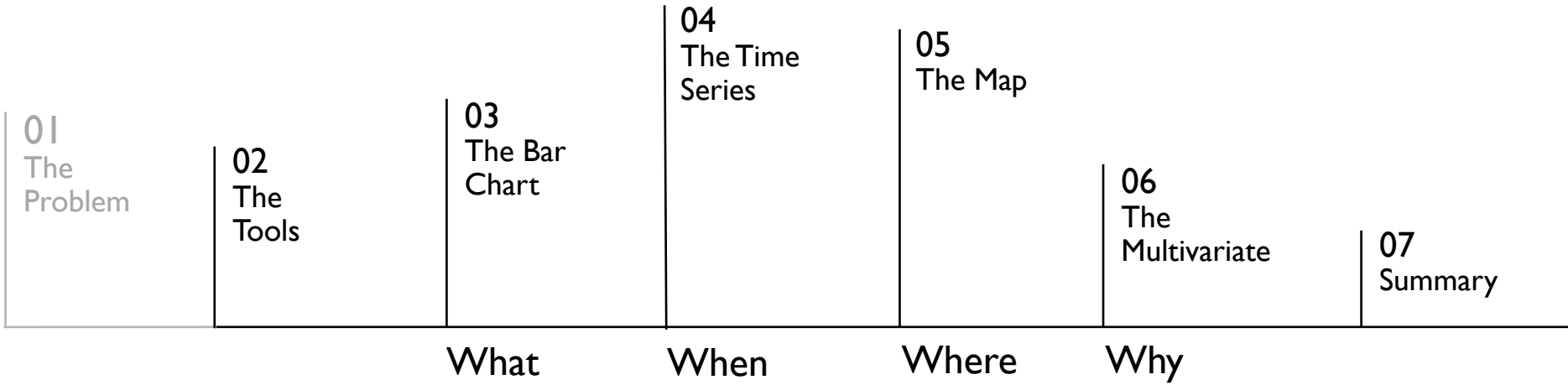




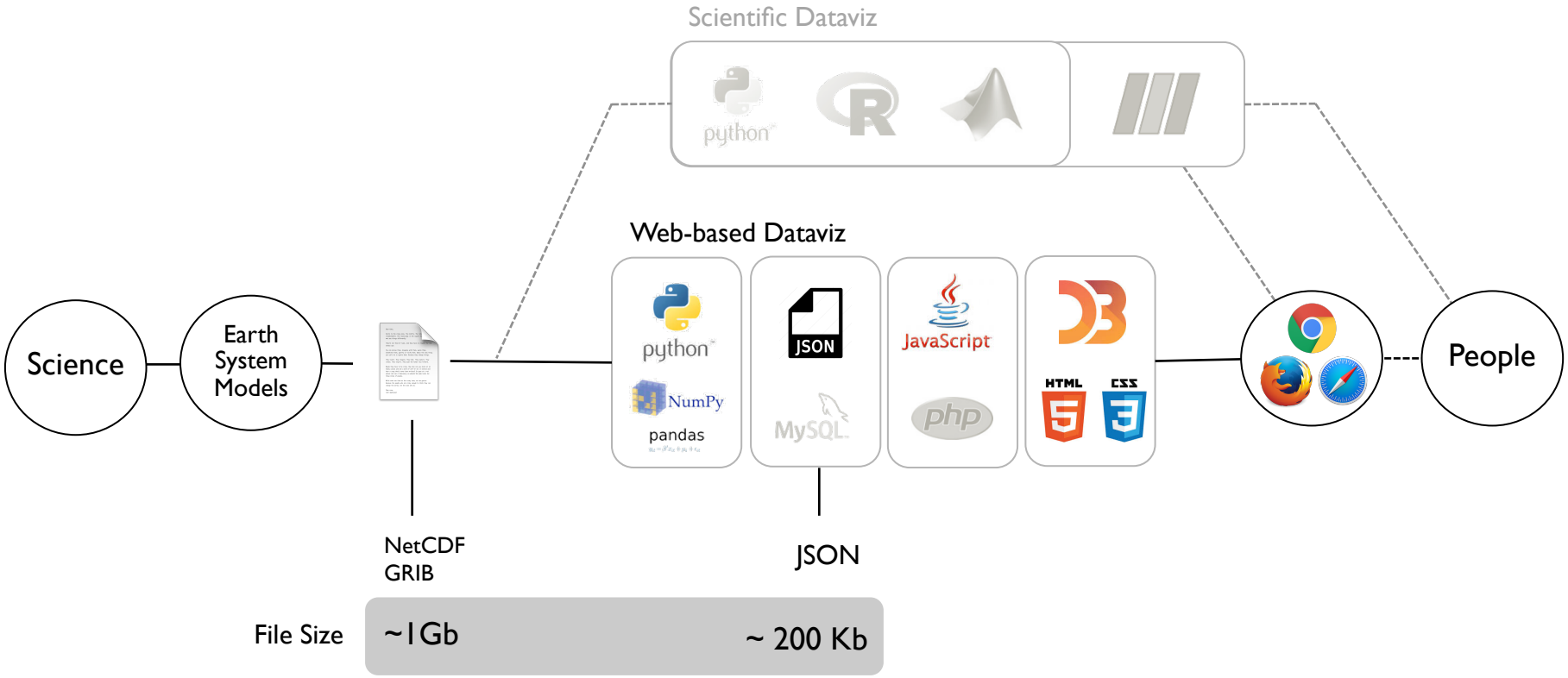
1 Bar Chart 4 Time Series 4 Maps

The Opportunity

Outline

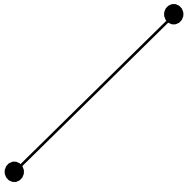

01
The
Problem
02
The
Tools
03
The Bar
Chart
04
The Time
Series
05
The Map
06
The
Multivariate
07
Summary
What
When
Where
Why

The Tools


Scientific Dataviz
python
Web-based Dataviz
Science
Earth System Models
python
NumPy
pandas
JSON
MySQL
JavaScript
php
HTML
CSS
People
NetCDF
GRIB
JSON
File Size
~1Gb
~ 200 Kb

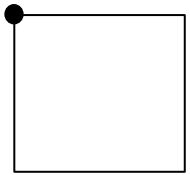
The Tools

@mbostock



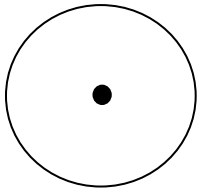
Line

x1,y1 (d)
x2,y2 (d)
color (d)
width (d)
opacity (d)



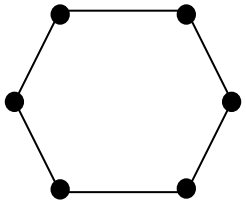
Rectangle

x,y (d)
width (d)
length (d)
border (d)
opacity (d)
fill color (d)



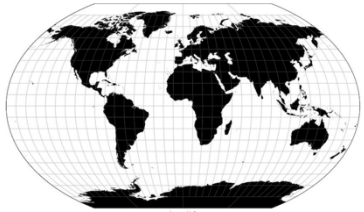
Circle

x,y (d)
radius (d)
border (d)
opacity (d)
fill color (d)

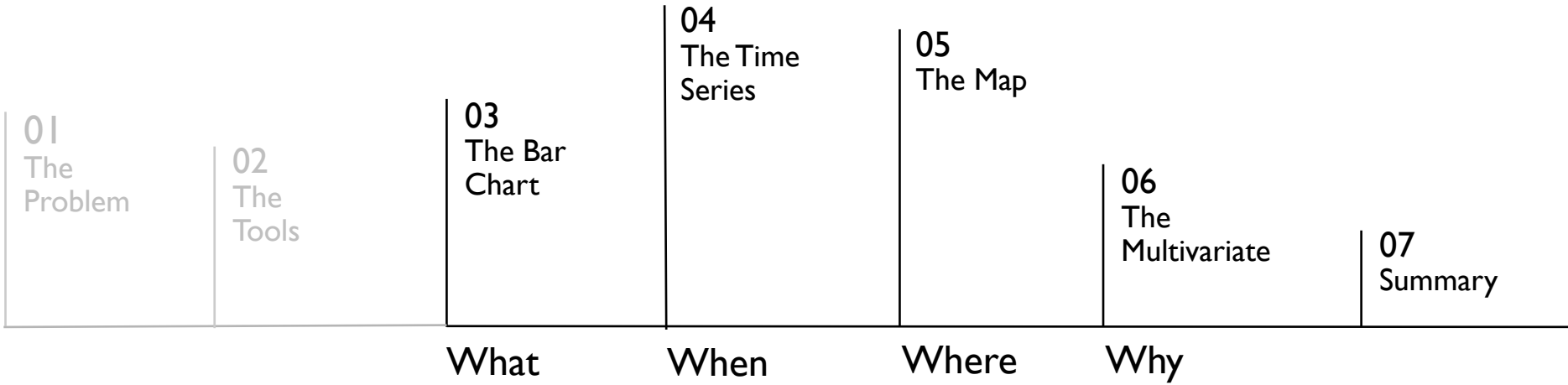


Polygon

x_i,y_i (d)
border (d)
opacity (d)
fill color (d)



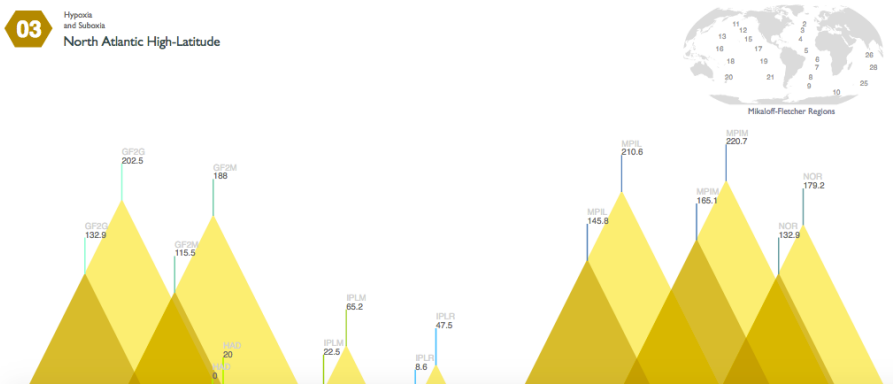
Map Projections


01
The
Problem
02
The
Tools
03
The Bar
Chart
04
The Time
Series
05
The Map
06
The
Multivariate
07
Summary
What
When
Where
Why

The Bar Chart

Accepted
Rejected

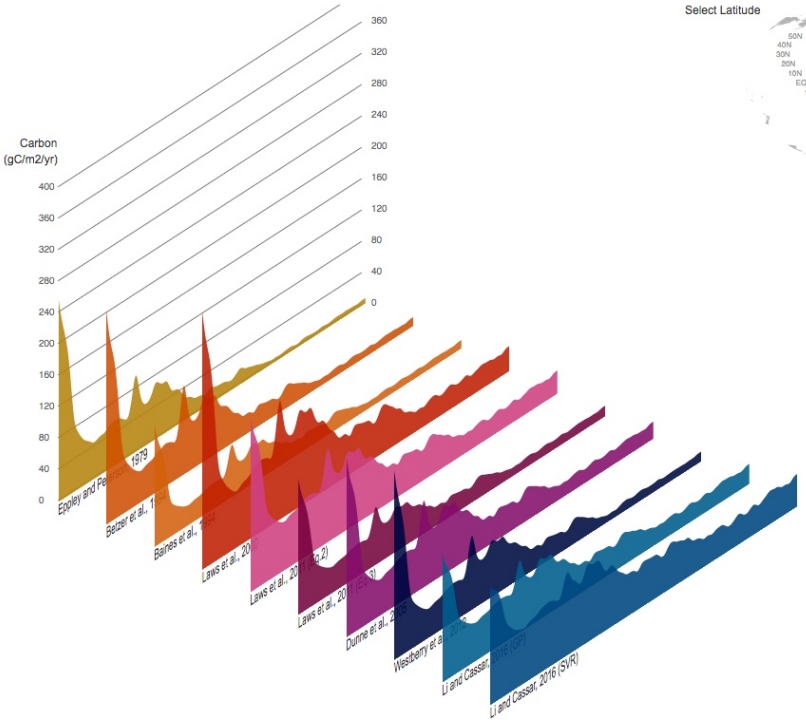
Type Stacked Dots
Data Model output of N_2-fix rates

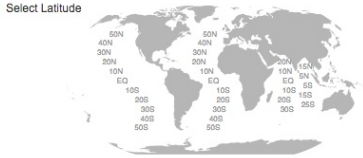


Type Polygons
Data Model output of oceanic hypoxic and suboxic volumes

```
svg.selectAll('polygon').data(data).enter().append('polygon')

.attr("points", function(d) {
   var a=(d.model_num)*k + "," + 100
   var b=(d.model_num)*k + (d.value *.75) + "," + (100 - (d.value *1.5))
   var c=(d.model_num)*k + (d.value *1.5) +  "," + 100
   return  (a + " " + b + " " + c);})
```

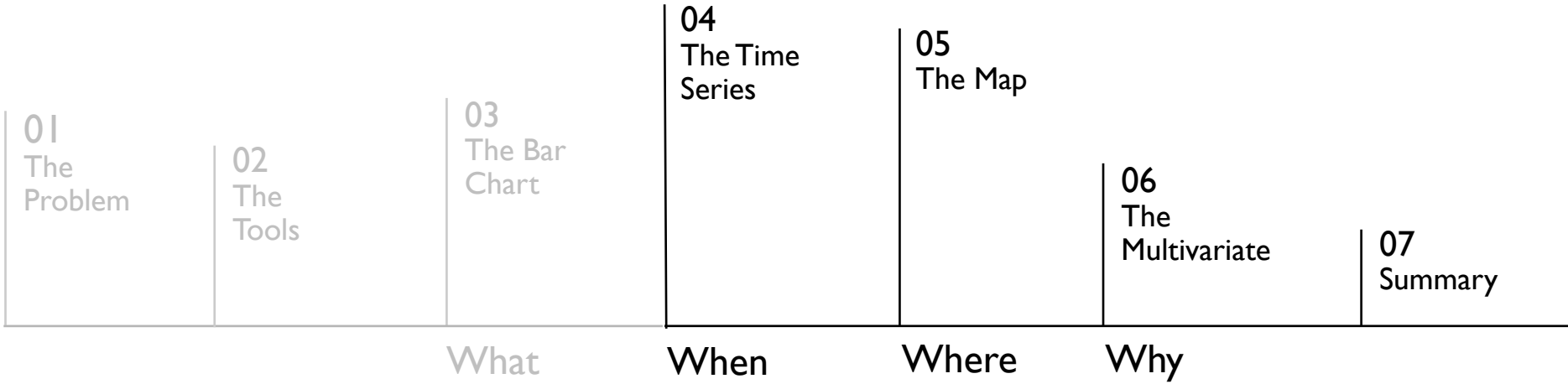
The Bar Chart



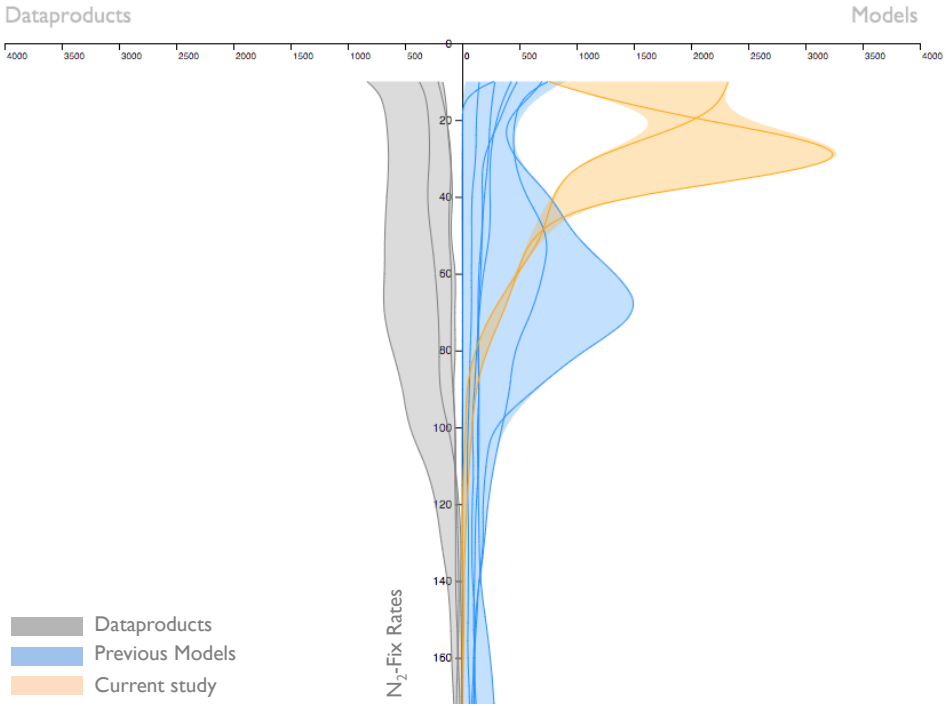


- Type Joy Plot
  Data Model output of oceanic carbon export

```
d3.json(json_file, function(error, data) {

 var svglines = svg.append("g")
 .attr("transform", function(d) {

    var pitch=data[0].model
    return "translate(" + (pitch*10) + ","
    + (pitch*2) +")

    rotate(315)skewX(-45)skewY(10)

    translate(" + -pitch*7 + "," + -pitch*2 + ")
    scale(.5)";
    });

 svglines.append("path")
 .attr("d", area(data));
 ...
}
```


01
The
Problem
02
The
Tools
03
The Bar
Chart
04
The Time
Series
05
The Map
06
The
Multivariate
07
Summary
What
When
Where
Why

The Time Series



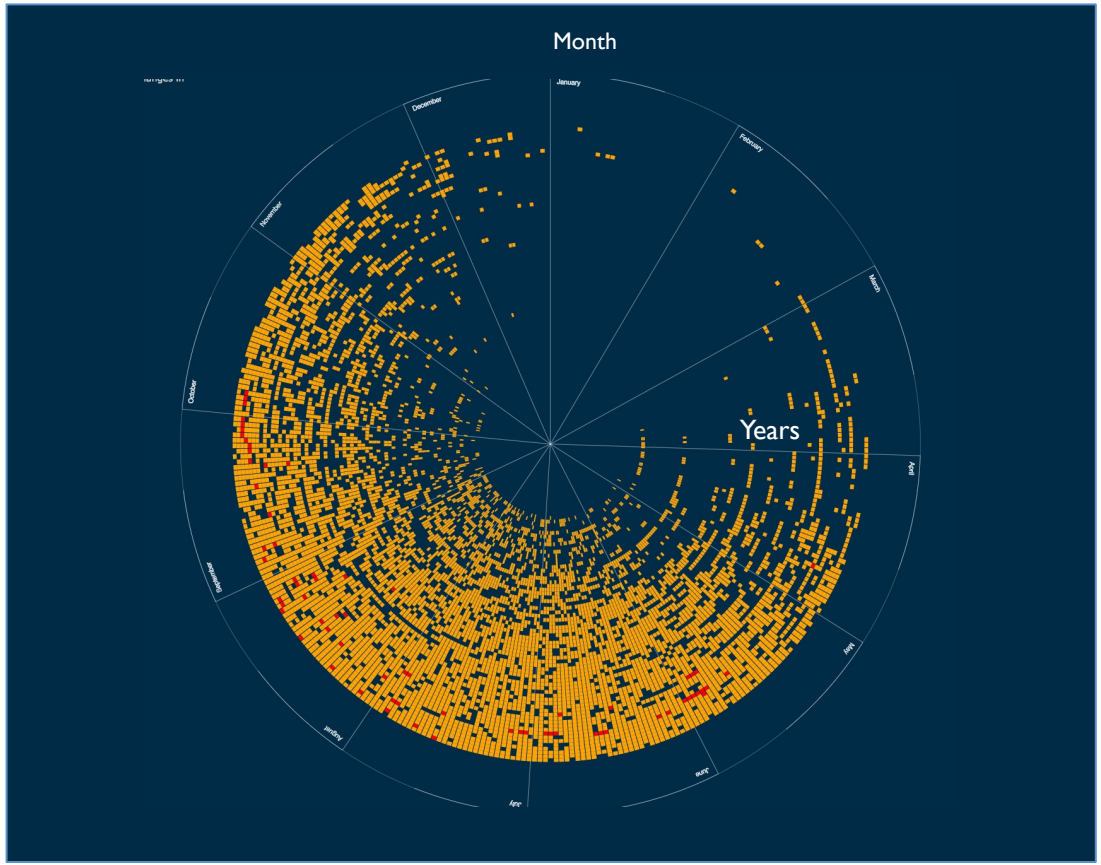
Type Histogram
Data Model output & Dataproducts of N_2-fixation rates

python

```
mat = scipy.io.loadmat(data.mat)

hist, bins = np.histogram(variable,
bins=np.linspace(5,500))

mypanda=pd.DataFrame({"hist":hist})
mypanda.reset_index().to_json...
```

```
var thelineRT = d3.line()
    .curve(d3.curveBasisOpen)
    .x(function(d) {return x(d.hist);})
    .y(function(d) {return y(d.index);})

var theareaRT = d3.area()
    .x(function(d) {return y(d.index);})
    .y0(function(d){return x(d.themin);})
    .y1(function(d){return x(d.themax);});
```

The Time Series



Type Heatmap
Data Meteorological records of temperature > 26°C from 1945 to 2017

python

```
fronts=path+'data.txt'
fq = open(fronts,'r')
for g in fq.readlines():
  theline=g.strip()
  datasplit=theline.split('\t')
  myarray.append(np.float(datasplit[1])
  ...
```

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

.innerRadius(function(d){return d.in;})
.outerRadius(function(d){return d.out;})
.startAngle(function(d) {return d.start*Math.PI;})
.endAngle(function(d) {return d.end*Math.PI;})

svg.selectAll("path").data(data).enter()
   .append("path")
   .attr("d", arc)
```

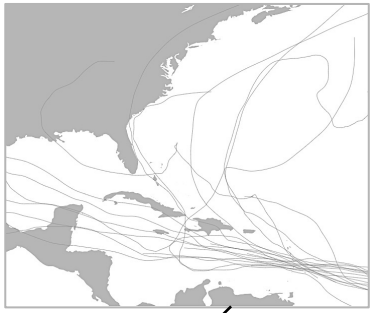
The Time Series

Hurricane Data

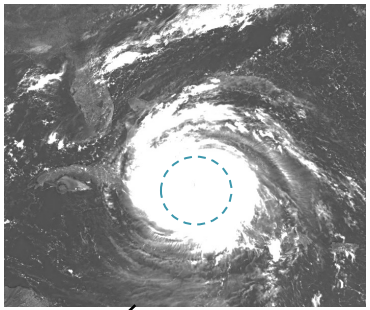
An example of the new HURDAT2 format for Hurricane Irene from 2011 follows:

```
AL092011,                 IRENE,     39,
20110821, 0000,  , TS, 15.0N,  59.0W,  45, 1006,  105,    0,    0,   45,
20110821, 0600,  , TS, 16.0N,  60.6W,  45, 1006,  130,    0,    0,   80,
20110821, 1200,  , TS, 16.8N,  62.2W,  45, 1005,  130,    0,    0,   70,
20110821, 1800,  , TS, 17.5N,  63.7W,  50,  999,  130,   20,    0,   70,
20110822, 0000,  , TS, 17.9N,  65.0W,  60,  993,  130,   30,   30,   90,
20110822, 0600,  , HU, 18.2N,  65.9W,  65,  990,  130,   60,   60,   90,
20110822, 1200,  , HU, 18.9N,  67.0W,  70,  989,  160,   60,   60,   90,
20110822, 1800,  , HU, 19.3N,  68.0W,  75,  988,  160,   60,   40,   90,
20110823, 0000,  , HU, 19.7N,  68.8W,  80,  981,  160,   70,   50,  100,
20110823, 0600,  , HU, 20.1N,  69.7W,  80,  978,  180,  120,   90,  130,
20110823, 1200,  , HU, 20.4N,  70.6W,  80,  978,  180,  120,   90,  130,
20110823, 1800,  , HU, 20.7N,  71.2W,  80,  977,  180,  120,   90,  130,
20110824, 0000,  , HU, 21.0N,  71.9W,  80,  969,  180,  150,   90,  150,
20110824, 0600,  , HU, 21.3N,  72.5W,  95,  965,  180,  150,   90,  150,
20110824, 1200,  , HU, 21.9N,  73.3W, 105,  957,  180,  150,   90,  150,
```

Trajectories



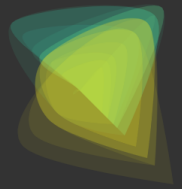
Max 34Kn Wind Radius



Lat, Lon, Time

NW, NE, SW, SE distances, Time

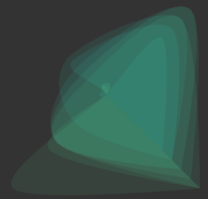
Hurricanes



Ophelia
20/09/2011

Nate
06/09/2011

Maria
06/09/2011

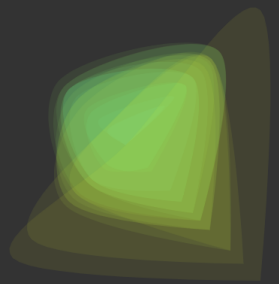


Lee
02/09/2011

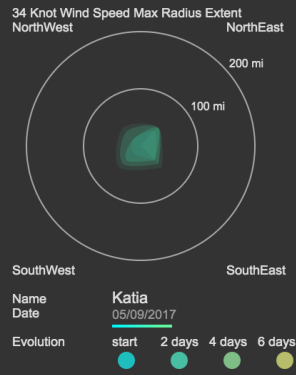
Katia
28/08/2011

Jose
26/08/2011

Ten
25/08/2011



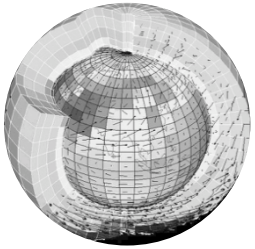
Irene
21/08/2011



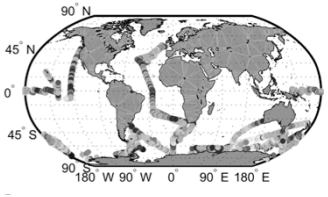

```
var points = [

  [data[i].se34 , data[i].se34],
  [-data[i].sw34 ,data[i].sw34],
  [-data[i].nw34 ,-data[i].nw34],
  [data[i].ne34 , -data[i].ne34],
  [data[i].se34 , data[i].se34]
];

var pathData = lineGenerator(points);

d3.select("#spot_"+inspot)
   .append('path')
   .attr('d', pathData)
```


01
The
Problem
02
The
Tools
03
The Bar
Chart
04
The Time
Series
05
The Map
06
The
Multivariate
07
Summary
What
When
Where
Why

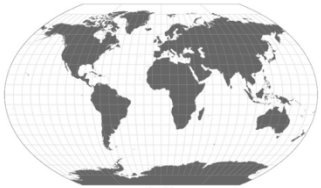
The Map



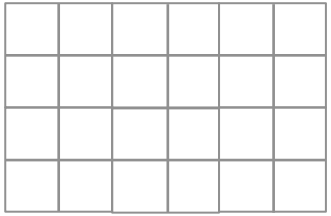
Earth System Model
NetCDF / GRIB



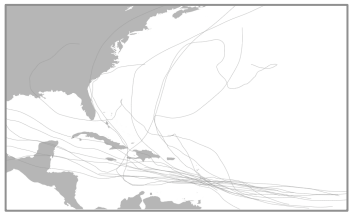
Observations
CSV



Map Projections
GeoJSON / TopoJSON

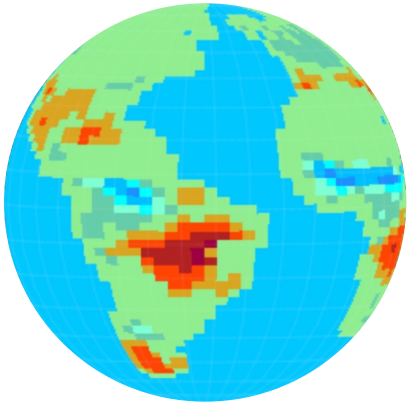


Distribution of tiles

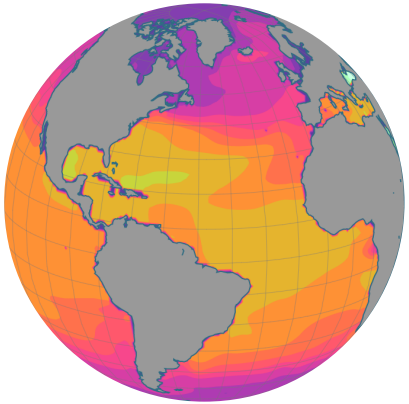


Small Features

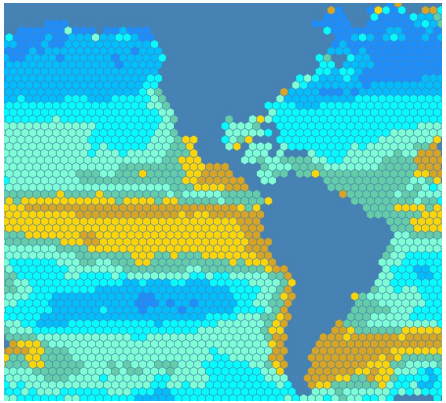
The Map



Type Binary Gridded File
Data Model Output



Type NetCDF Contours
Data Model Output



Type Hexagonal Binning
Data Model Output

```
oxygen=np.array(oxy)
binaryoxy=oxygen.astype(np.float32).tobytes()
```

```
d3.xhr('./oxygen.dat.npy')
   .responseType('arraybuffer')
   .get(function(error, xhr) {
   var data = new Float32Array(xhr.response);
   map.addLayer(data, layerOptions);
});
```

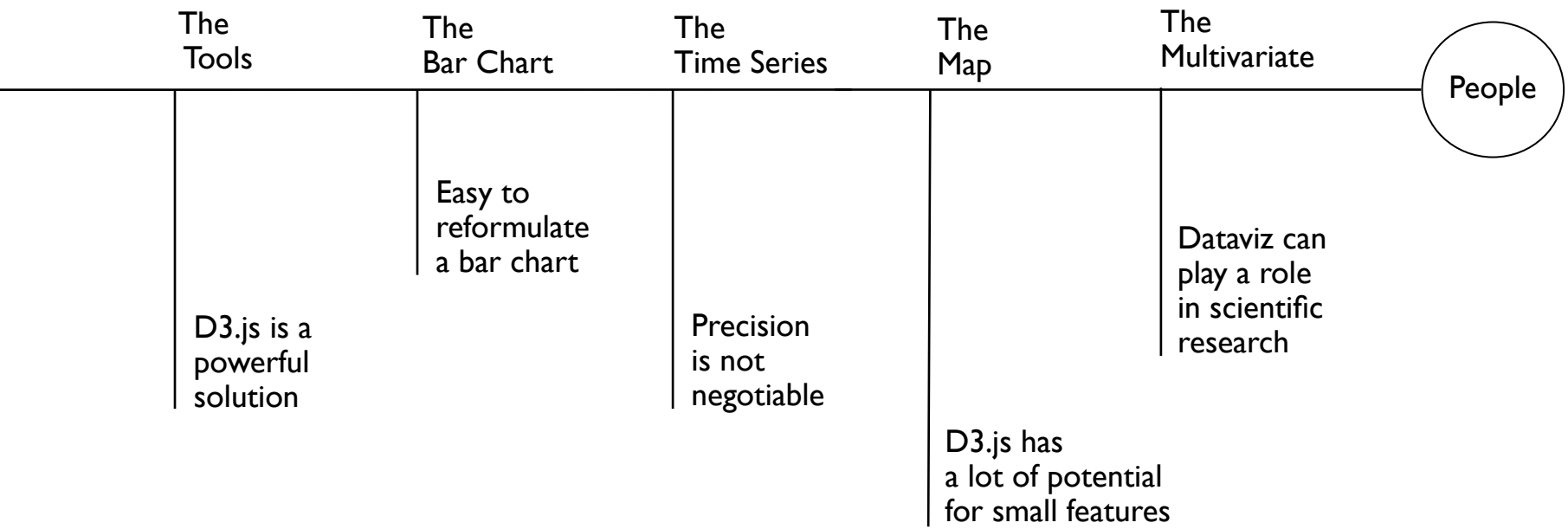
```
var hexbin = d3.hexbin()
  .size([width, height])
  .radius(theradius)
  .x(function(d) {return projection([d.lon, d.lat])[0];})
  .y(function(d) {return projection([d.lon, d.lat])[1];});
```

```
coordinates: d3.merge(d.coordinates.map(function(polygon) {
        return polygon.map(function(ring) {
          return ring.map(function(point) {
            return [point[0] / isize * 360 - 180, 90 - point[1] / jsize * 180];
          }).reverse();});
```

The Map


01
The
Problem
02
The
Tools
03
The Bar
Chart
04
The Time
Series
05
The Map
06
The
Multivariate
07
Summary
What
When
Where
Why

Summary


The Tools
The Bar Chart
The Time Series
The Map
The Multivariate
People
D3.js is a powerful solution
Easy to reformulate a bar chart
Precision is not negotiable
D3.js has a lot of potential for small features
Dataviz can play a role in scientific research