

- 如发现翻译不当或有其他问题可以通过以下方式联系译者:
- 邮箱: zhang_tianxu@sina.com
- QQ群: [D3数据可视化](#)205076374, [大数据可视化](#)436442115

对于地图的可视化, D3支持少数的组件显示和操作地理数据。这些组件使用GeoJSON格式——在JavaScript中标准的地理特征表示方法。(可以参见TopoJSON格式,它是GeoJSON的扩展格式,表示的更加紧密。)。要把图形文件转换为GeoJSON,使用ogr2ogr的GDAL包的一部分。

这有一些你可能感兴趣的其他工具:

- TopoJSON –简化图形文件, 拓扑结构和GeoJSON压缩。
- Shapely –操作平面几何图形对象。
- ColorBrewer –地图颜色的比例尺。
- PostGIS –一个地理空间数据库。

显示地理数据最主要的机制就是用到d3.geo.path。这个类是类似于d3.svg.line和其他SVG形状生成程序: 给定一个几何形状或功能的对象, 它生成适合于SVG路径元素的“d”属性的路径数据串。该d3.geo.path类可以直接渲染到画布上, 在动画投影时可以提供更好的性能。

d3.geo.path()

创建一个新的地理路径生成器, 使用默认设置: albersUsa投影和4.5个像素点的半径。

path(feature[, index])

为给定的功能feature返回路径数据串, 它可以是任何GeoJSON的特征或几何对象:

- Point –单个位置。
- MultiPoint –一组位置。
- LineString –一组位置形成一条连续的线。
- MultiLineString - 位置数组的数组, 形成多条线。
- Polygon –位置数组的数组, 形成一个多边形 (可能是由小洞组成的)。
- MultiPolygon –位置的多维数组, 形成了多个多边形。
- GeometryCollection –几何对象的数组。
- Feature - 包含了几何对象其中的一个特征。
- FeatureCollection –特征对象的数组。

“Sphere”类型也支持, 对于绘制地球的轮廓是非常有用的。sphere球体没有坐标。一个可选的index 索引可以被指定, 传递给pointRadius存取器, Index在路径生成器被selection.attr调用的时候自动地传递。重要的是: 一多边形内部是所有的点, 这些点是以顺时针方向围绕着多边形。如果你的GeoJSON输入有错误顺序的多边形, 那么你必须扭转它们, 通过ST_ForceRHR, 你也可以将你的GeoJSON转换为TopoJSON, 这样的话, 它就能自动生成。显示多个功能, 你可以将它们放置在一个单一的特征集和单一路径元素上:

```
svg.append("path")
  .datum({type:"FeatureCollection",features:features})
  .attr("d",d3.geo.path());
```

另外, 您也可以创建多个不同的路径元素:

```
svg.selectAll("path")
  .data(features)
```

```
.enter().append("path")
.attr("d", d3.geo.path());
```

对于集合来说利用不同的路径元素通常地比单个路径元素要慢。但是，如果你想与特征分别进行交互的话，不同的路径元素是最好的（例如，使用CSS：悬停或单击事件）。

path.projection([projection])

如果 指定了投影projection，设置为路径生成器使用的投影为指定的投影函数。如果projection 投影没有指定，那么返回当前的投影，默认为albersUsa。projection 投影通常是D3一个内置地理投影；然而，任何函数都可以使用。投影函数采用了两元素的数字数组来代表坐标的位置，[longitude, latitude]，并返回类似的表示该投影像素位置[x, y]的两元素数字数组。例如，一个基本的spherical Mercator投影：

```
function mercator(coordinates) {
  return [
    coordinates[0] / 360,
    (-180 / Math.PI * Math.log(Math.tan(Math.PI / 4 + coordinates[1] * Math.PI / 360))) / 360
  ];
}
```

从内在的说，这个点投影函数是包裹着一个备用的流变换，它可以执行自适应重采样。然而，备用的流不执行任何裁剪或切割。为了更好地控制流的变换，projection 投影可以被指定为一个对象来实现流方法（参见实例：<http://bl.ocks.org/mbostock/5663666>）。该流的方法需要一个输出流作为输入，并返回一个投影在输入的几何体上的包装流，换句话说，它实现了projection.stream。如果投影为null，则路径使用恒等变换，其中输入的几何体不被投影并且不是直接以原始坐标来渲染。这对已经投影的几何体快速渲染有用，或对正方形投影的快速渲染有用。

path.context([context])

如果指定了context，设置渲染上下文并返回生成的路径。如果context 为空，当在一个给定的功能调用时，路径生成器将返回一个SVG路径字符串。如果context 非空，路径生成器将替换调用函数为指定的上下文来渲染几何图形。

context 必须实现以下方法：

```
beginPath()
moveTo(x, y)
lineTo(x, y)
arc(x, y, radius, startAngle, endAngle)
closePath()
```

可以注意到，这是画布元素的二维渲染上下文的子集，这样画布context 可以被传递到路径生成器，在这种情况下，几何形状将直接渲染到画布上。如果context 没有指定，则返回当前渲染的上下文，默认为null。

path.area(feature)

对指定的功能feature计算投影面积（使用方形像素）。Point，MultiPoint，LineString和LineString特征有为零的区域。对多边形Polygon和多边形集合MultiPolygon的特征，该方法首先计算外部环形的面积，再减去内部有任何孔的区域。这种方法通过投影流观察进行任何的剪裁和重新采样。

path.centroid(feature)

对指定的功能feature计算投影重心（以像素为单位），这是非常方便的。比如说，标记州或国家的边界，或显示符号映射。非连续统计图，示例了围绕其质心的每个状态。这种方法观察通过投影流运行的任何剪裁和重新采样。非连续

统计图: <http://mbostock.github.com/d3/ex/cartogram.html>

path.bounds(feature)

对特定的功能计算投影的边框（像素），这是非常方便的。比如说，放大到一个特定的形态。这种方法观察通过投影流运行的任何剪裁和重新采样。

path.pointRadius([radius])

如果指定半径，那么设置的半径为特定的数，用来显示点Point和多点MultiPoint的功能。如果未指定半径，则返回当前的半径。而半径通常指定为一个数字常数，它也可以被指定为对每个特征进行计算的函数，传递来自路径函数的feature和index参数。例如，如果你的GeoJSON数据有额外的属性，你就可以访问内置的半径函数中的属性来改变点的大小；或者，你可以用d3.svg.symbol和一个投影来更好地控制显示。Shape Generators

注：在D3中生成一个巨大的弧线，只需要简单地把一个LineString类型的几何对象传递给d3.geo.path。D3的投影采用great-arc插值器生成中间点（使用了自适应重采样）。因此没有必要使用形状生成器来创造大的弧线。

d3.geo.graticule

构造一个特征生成器用来创建地理刻度。

graticule()

返回一个MultiLineString几何对象表示这个刻度的所有经线和纬线。

graticule.lines()

返回LineString几何对象，用于这个刻度的每一个经线和纬线。

graticule.outline()

返回一个多边形Polygon几何对象，代表了这个地理坐标的轮廓，IE浏览器沿着它的经线和纬线界定其范围。

graticule.extent(extent)

如果extent指定了，设置此刻度的主要和次要范围。如果没有指定范围，返回当前次要的范围，默认为 $\langle (-180^\circ, -80^\circ - \epsilon), (180^\circ, 80^\circ + \epsilon) \rangle$ 。

graticule.majorExtent(extent)

如果extent指定了，设置此刻度的主要范围。如果未指定范围，返回当前主要的范围，默认为 $\langle (-180^\circ, -90^\circ + \epsilon), (180^\circ, 90^\circ - \epsilon) \rangle$ 。

graticule.minorExtent(extent)

如果extent指定了，设置此刻度的较小范围。如果没有指定范围，返回当前较小的范围，默认为 $\langle (-180^\circ, -80^\circ - \epsilon), (180^\circ, 80^\circ + \epsilon) \rangle$ 。

graticule.step(step)

如果step 指定了，设置这个刻度的主要和次要的步长。如果没有指定step，则返回当前次要的step，默认为 $\langle 10^\circ, 10^\circ \rangle$ 。

graticule.majorStep(step)

如果step 指定了，设置这个刻度的主要step。如果没有指定step，返回当前的主要step，它默认为 $\langle 90^\circ, 360^\circ \rangle$ 。

graticule.minorStep(step)

如果step 指定了，设置这个刻度次要的step。如果没有指定step，返回当前的次要step，默认为 $\langle 10^\circ, 10^\circ \rangle$ 。

graticule.precision(precision)

如果指定了precision，设置这个刻度的精度，以度为单位。如果没有指定精度，则返回当前精度，默认值为 2.5° 。

d3.geo.circle

为在一个给定的地理位置创建带有给定半径（以度为单位）的圆中心构建一个特征生成器。

circle(arguments...)

返回一个接近圆形的GeoJSON多边形。原点访问器指定如何为给定的参数确定原点，默认的访问使用的是常量 $\langle 0^\circ, 0^\circ \rangle$ 。

circle.origin([origin])

如果指定了origin，设置圆的原点。一个两元坐标数组应该被指定，或访问函数。如果origin 没有指定，则返回当前的原点，默认值为 $\langle 0^\circ, 0^\circ \rangle$ 。

circle.angle([angle])

如果指定了angle，用度数来设置圆的角半径。如果angle 没有指定，则返回当前的半径，默认为 90° 。

circle.precision([precision])

如果precision 指定，以度为单位设置圆形片段的精度的插值器。当一个特征被圆形裁剪时，这些内插节段就被插入了。如果没有指定precisionis，则返回当前精度，默认值为 6° 。

Spherical Math

d3.geo.area(feature)

返回指定功能的球形区域（以球面度为单位）。可以参考path.area，其在Cartesian plane平面上计算投影区域。

d3.geo.centroid(feature)

返回指定功能的球形重心。可参考path.centroid，其在Cartesian plane平面上计算投影重心。

d3.geo.bounds(feature)

返回指定功能的球形边框。边框是由一个二维数组来表示: `[[left, bottom], [right, top]]`，其中`left`是最小经度，`bottom`是最小纬度，`right`是最大经度和`top`是最大纬度。可参考`path.bounds`，其在Cartesian plane平面上计算投影的边框。

d3.geo.distance(a, b)

返回在两个点a和b之间弧度的最大距离，每个点都被指定为一个数组`[longitude, latitude]`，和表示十进制度的坐标。

d3.geo.length(feature)

返回弧度指定功能的大弧great-arc的长度。对于多边形，返回外环周长加上任何内置的环。

d3.geo.interpolate(a, b)

返回给定的两个位置a和b的插值器。每个位置必须被表示为一个`[longitude, latitude]`的二元数组。返回的插值器是一个函数，它可以接受单个参数t作为输入，其中t的范围是0~10，0返回a位置的，而1返回b的位置。中间值的插入是从a到b的大弧形。

d3.geo.rotation(rotate)

设置一个旋转为 `[λ, φ, γ]`形式的数组。数组的元素是以度为单位的角度，并指定按照下面的顺序来旋转：纵向，横向和原点。如果数组的最后一个元素，`γ`，省略了，那么这个默认值就为0。返回一个函数，该函数就如所述的那样转动到一个指定的位置。

rotation(location)

在上述描绘的顺序下，这个旋转是在一个指定的位置，根据特定的角度来旋转的。位置被指定为一个数组`[longitude, latitude]`，用度数表示的坐标。返回一个新的数组来表示旋转的位置。

rotation.invert(location)

该旋转是根据指定的角度和位置来旋转的，但与上述描述的反向顺序来旋转。一个位置指定为一个数组`[longitude, latitude]`，用度数来表示坐标。返回一个新的数组来表示旋转的位置。

- 翟琰琦译 20141124
- 何凯琳 译2014年11月27日 20:45:29
- Gulu 校对 2014-12-7 21:45:37