

[Wiki](#) ▶ [\[\[API--中文手册\]\]](#) ▶ [\[\[地理\]\]](#) ▶ **地理投影**

- 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

D3默认包括了一些常见投影，如下所示。众多的（不太常用的）投影在扩展地理投影插件和多面体投影插件中是可用的。

由D3提供的大多数投影都是通过d3.geo.projection来创建并配置的，你可以旋转这个地球，缩放或转换画布等。除非你正在执行一个新的原始投影，否则你可能不会用D3.geo.projection来构造，但是有可能使用这个配置方法。

d3.geo.projection(raw)

从指定的原始raw 点投影的函数，构造一个新的投影。例如，一个Mercator投影可实现为：

```
var mercator = d3.geo.projection(function(λ, φ) { return [λ, Math.log(Math.tan(π/4 + φ/2))]; });
```

（可以参考src/geo/mercator.js全面的实现。）如果原始函数支持反转方法，则返回的投影将会显示一个对应的反转方法。
src/geo/mercator.js: <https://github.com/mbostock/d3/blob/master/src/geo/mercator.js>

projection(location)

投影从球面坐标（单位：度）转到笛卡尔坐标（单位：像素），返回数组[x, y]，给出输入数组[longitude, latitude]。如果指定的位置没有定义投影的位置，那么返回可能为null。例如，当位置在投影的裁剪边界之外的時候。

projection.invert(point)

投影反向是从直角坐标（像素），到球面坐标（度），返回一个数组[longitude, latitude]，给定输入数组[x, y]。但并非所有的投影都会实现反转，对于可逆投影，这种方法是undefined。

projection.rotate([rotation])

如果rotation 旋转指定了，设置投影的三轴旋转为指定的角度λ，φ和γ（偏航角，倾斜角和滚动角，或等效地经度，纬度和滚动），以度并返回投影。如果rotation 未指定，返回当前缺省的转动值[0, 0, 0]。如果rotation 指定且只有两个值，而不是3个值，那么滚动的角度被假设为0°。

projection.center([location])

如果center 重心指定了，则设置投影中心为指定的位置，经度和纬度的两元数组，并返回投影。如果没有指定中心，则返回当前的中心，默认为(0°, 0°)。

projection.translate([point])

如果point 点指定了，则设置投影转变的位移为指定的二元数组[x, y]并返回的投影。如果未指定点，则返回当前变换的位移，默认为[480, 250]。变换的位移确定投影的中心像素坐标。默认转换位移的位置是(0°, 0°)，在一个960×500区域的中心。

projection.scale([scale])

如果scale 比例尺指定了，则设置投影的比例尺为特定的值，并返回投影。如果未指定比例尺，返回默认的值150。比例尺因子线性地对应于投影点之间的距离。然而，比例尺因子不能一直越过投影。

projection.clipAngle(angle)

如果angle 角度指定了，则设置投影的裁剪圆半径，指定角度并返回投影。如果角度为null，切换到子午线切割，而不是小圈的裁剪。如果没有指定角度，返回当前裁剪的角度，默认为空。小圈裁剪是独立于通过clipExtent的视窗裁剪。antimeridian cutting: <http://bl.ocks.org/mbostock/3788999>

projection.clipExtent(extent)

如果extent 范围指定了，则设置投影的剪辑视窗范围为指定的边界，以像素为单位并返回投影。extent 范围边界被指定为一个数组[[x0, y0], [x1,y1]]，其中x0 是视窗的左侧，y0 是顶部，x1 为右侧和y1 是底部。如果范围为null，则视窗裁剪不执行。如果extent 范围没有指定，则返回当前视窗裁剪的范围，默认为null。视窗裁剪是独立于通过clipAngle的小圈剪裁。

projection.precision(precision)

如果precision 精度指定了，则设置投影自适应重采样的临界值为指定的值，以像素为单位，并返回投影。此值对应于Douglas-Peucker 距离。如果precision 精度没有指定，则返回投影当前重采样的精度，其精度默认为Math.SQRT(1/2)。0的精度禁止自适应重采样。

projection.stream(listener)

返回一个投影流，包装了特定的监听器。任何几何形状的对封装器，都是在被传输到包监听之前投影的。一个典型的投影包含多个流的变换：输入的几何形首先被转换为弧度，在三轴上旋转，在小圆圈或沿着子午线切割，最后投射到具有自适应重采样，缩放和平移的笛卡尔平面上。

d3.geo.projectionMutator(rawFactory)

构造一个新的投影，是从指定的原始点投影函数factory开始。此函数不直接返回投影，而是返回一个变换的方法，且你可以随时调用原始投影函数发生变换的方法。例如，假设你正在实施Albers equal-area conic投影，它需要配置投影的两大相似之处。使用闭包，可以实现原始投影，如下所示： // φ0 and φ1 are the two parallels
functionalbersRaw(φ0,φ1){ returnfunction(λ,φ){ return[/* compute x here */ , /* compute y here */]; }; }

使用d3.geo.projectionMutator，可以实现一个标准的投影，使相似之处有所改变，重新分配由d3.geo.projection内部使用的原始投影： functionalbers(){ varφ0=29.5, φ1=45.5, mutate=d3.geo.projectionMutator(albersRaw),
projection=mutate(φ0,φ1);

projection.parallels=function(){ if(!arguments.length)return[φ0,φ1]; returnmutate(φ0+=+[0],φ1+=+_[1]); };

returnprojection; } 因此，在创建一个可变的投影时，变化的函数永远不会暴露出来，但可以很容易地重新创建底层的原始投影。对于完全实现，可以参考src/geo/albers.js。 Standard Projections

d3.geo.albers()

d3.geo.albers是d3.geo.conicEqualArea的一个别名，默认为美国中心：scale 1000，translate [480, 250]，rotation [96°, 0°]，center (-0.6°, 38.7°) 和parallels [29.5°, 45.5°]，使其能够适合的显示美国，中心围绕着Hutchinson, Kansas 在一个960×500的区域。中央经线和纬线是由美国地质调查局在1970年国家地图集规定的。

d3.geo.albersUsa()

埃尔伯USA投影是一个复合投影，这复合的四个埃尔伯投影的设计是用来显示美国下部48，在阿拉斯加和夏威夷旁边。虽然用于等值线图，它扩展阿拉斯加0.35×倍的区域（估计是3倍），夏威夷作为下部48显示在相同比例尺的。埃尔伯USA投影不支持旋转或者居中。

d3.geo.azimuthalEqualArea()

方位角等面积投影也适用于等值线图。这个投影的极性方面用于联合国标志。 polar aspect：
<http://bl.ocks.org/mbostock/4364903>

d3.geo.azimuthalEquidistant()

方位角的等距投影保留了从投影中心的距离，从任何投影点到投影中心的距离到大弧距离是成比例的。因此，围绕投影中心的圆圈是投影在笛卡尔平面的圆圈。这可以用于相对一个参考点的可视化距离，如可交换距离。

d3.geo.conicConformal()

兰伯特的等角的二次曲线投影投影地球形状成为一个锥形。

conicConformal.parallels([parallels])

如果parallels 指定了，则设定投影的标准平行线为特定的二元纬度数组，并返回这个投影。如果parallels 没有指定，返回当前的parallels。

d3.geo.conicEqualArea()

埃尔伯投影，作为一个区域相等的投影，被推荐用于等值线图，因为它保留了地理特征的相对区域。

conicEqualArea.parallels([parallels])

如果parallels 指定了，设置埃尔伯投影的标准平行线为指定的二元纬度数组（以度为单位），并返回投影。如果parallels 没有指定，返回当前parallels。为了最大限度地减少失真，平行线应选择为围绕投影的中心。

d3.geo.conicEquidistant()

conicEquidistant.parallels([parallels])

如果parallels 指定了，设定投影的标准parallels 到指定的二元纬度数组（度），并返回投影。如果parallels 没有指定，返回当前的parallels。

d3.geo.equirectangular()

这个正方形投影，或plate carrée投影，是最简单可行的地理投影：标识函数。它既不等面积也不等角，但有时用于光栅数据。见光栅重投影的一个例子，源图像使用正方形投影。 raster reprojection：
<http://bl.ocks.org/mbostock/4329423>

d3.geo.gnomonic()

球心投影是一个方位角投影，它投射大圆为直线。参考interactive gnomonic 的例子。 interactive gnomonic：
<http://bl.ocks.org/mbostock/3795048>

d3.geo.mercator()

这个球面的Mercator投影是常用的分片式映射库（例如OpenLayers 和Leaflet）。例如显示栅格分片与Mercator投影，可以参见d3.geo.tile插件，它是正形投影的，然而，它的推行造成了世界范围地区严重失真，因此不建议使用choropleths。

d3.geo.orthographic()

正投影是适合于显示单个半球的方位投影，透视的点在无穷远处。可以看世界观光的动画和互动正投影的例子。对于一般的透视投影，可以参考卫星投影。

d3.geo.stereographic()

立体投影是另一个角度（方位）投影。在表面上看，透视的点是球体。因此，这是常用的天体图。参考交互式立体为例。

d3.geo.transverseMercator()

横向的墨卡托投影。 Raw Projections

D3提供了几个原始的投影，当一个复合投影实现时，设计重用（如Sinu-Mollweide，它结合了原始的正弦曲线和摩尔维特投影）。原始投影在使用之前，通常是用d3.geo.projection封装着。这些点函数都是采用球面坐标 λ 和 φ （弧度）作为输入，并返回一个二元数组（也是用弧度）作为输出。许多原始投影从平面坐标映射到球面坐标，实现了一个反转的投影。

d3.geo.albers.raw(φ_0 , φ_1)

是d3.geo.conicEqualArea.raw的一个别名。

d3.geo.azimuthalEqualArea.raw

原始方位角的等面积投影。

d3.geo.azimuthalEquidistant.raw

原始的等距方位投影。

d3.geo.conicConformal.raw(φ_0 , φ_1)

返回一个原始的等角的二次曲线投影，并用弧度指定parallels。

d3.geo.conicEqualArea.raw(φ_0 , φ_1)

返回一个原始的埃尔伯投影，并用弧度指定parallels。

d3.geo.conicEquidistant.raw(φ_0 , φ_1)

返回一个原始等距圆锥投影，并用弧度指定parallels。

d3.geo.equirectangular.raw

原始的正方形投影。

d3.geo.gnomonic.raw

原始的球心投影。

d3.geo.mercator.raw

原始的墨卡托投影。

d3.geo.orthographic.raw

原始的正投影。

d3.geo.stereographic.raw

原始的立体投影。

- 何凯琳译20141129
- gulu校对2014-12-7 23:27:06