

- 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

弦图所示内容为一组实体之间的关系。例如, 假定有一组不同发色的人: 黑色、金色、棕色和红色。该组每个人都为约会对象准备了中意的发色; 在所有29630个(假设)黑发人群中, 40%的人(11975)倾向于同具有相同发色的对象进行约会。但这种倾向是非对称的: 例如, 只有10%的金发人会选择黑发人群作为约会对象, 而20%的黑发人会金发人作为约会对象。



通过在不同的弧线之间画出二次贝塞尔曲线(Bézier curves), 将上述关系表示在一张弦图中。源弧线和目标弧线分别代表总人口的两个镜像子集, 如喜欢金发的黑发人口数量, 以及喜欢黑发的金发人口数量。另一个例子, 我们看这个软件依赖弦图的例子: [software dependencies](#).

弦布局同弦形([chord shape](#))和弧形([arc shape](#))协同工作, 用于生成数据对象。该对象作为弦形状的输入, 对弦进行描述。同时, 该布局还能生成对不同群组的描述, 用作弧形的输入。

`d3.layout.chord()`

构建新的弦布局。在默认情况下, 输入数据并未分类, 并且各群组之间没有填充。和其他布局不同, 该弦布局并不是应用于数据的函数; 相反, 数据通过设置关联矩阵([matrix](#))来指定, 通过 [chords](#) 和 [groups](#) 访问器检索。

`chord.matrix([matrix])`

指定矩阵 *matrix* 之后, 设定该布局用到的输入数据矩阵。如果没有指定矩阵 *matrix*, 返回当前数据矩阵, 默认为 `undefined`。输入矩阵的数字必须为“方形矩阵”([square matrix](#)), 例如:

```
[[11975, 5871, 8916, 2868],
 [ 1951, 10048, 2060, 6171],
 [ 8010, 16145, 8090, 8045],
 [ 1013, 990, 940, 6907]]
```

矩阵的每一行对应一个特定分组, 如上文所述某个发色。矩阵中每一列 *i* 同第 *i* 行相对应; 每个单元格 *ij* 对应表示第 *i* 组到第 *j* 组之间的关系。

`chord.padding([padding])`

指定填充 *padding* 之后, 将不同组之间设定角度填充为以弧度([radians](#))为单位的指定的值。如果没有指定填充, 返回当前填充, 默认值为0。你可能希望计算填充是分组数量(关联矩阵中行和列的数量)的函数。

`chord.sortGroups([comparator])`

如果已经指定 *comparator*, 使用指定 *comparator* 函数为布局设定分组(行)的排列顺序。为每两行调用 *comparator* 函数, 传递的入参是行和行的总和。通常, 需要将 *comparator* 按照 [d3.ascending](#) 或 [d3.descending](#) 进行指定。如果没有指定 *comparator*, 则返回当前分组排列顺序, 该顺序默认值为空。

`chord.sortSubgroups([comparator])`

如果已经指定 *comparator*, 使用指定 *comparator* 函数为布局设定分组(行内各列)的排列顺序。为每对单元格调用 *comparator* 函数, 值为各单元格的值。通常, 需要将 *comparator* 以升序或降序进行指定。如果没有指定 *comparator*, 则回当前子分组排列顺序, 该顺序默认值为空。

<#> chord.sortChords([comparator])

如果已经指定comparator，运用指定comparator函数为弦布局设定弦（Z顺序）的排列顺序。为每两条弦调用comparator函数，入参为源单元格和目标单元格的最小值。通常，要将comparator以升序或降序进行指定。如果没有指定comparator，返回当前chord排列顺序，默认值为空。

<#> chord.chords()

给定布局的当前配置和关联矩阵，返回计算过的弦对象。如果弦对象已计算完毕，本方法返回缓存值。如果布局属性有任何改变，则清空之前计算的弦。此时，如果下次调用该方法，需要对布局进行重新计算。返回对象具有下列属性：

- source -描述源对象。
- target -描述目标对象。

这两个对象描述下列实体：

- index -行索引，`_i_`。
- subindex索引-列索引，`_j_`。
- startAngle-弧的起始角，在radians内。
- endAngle-弧的终止角，在radians内。
- value -关联单元格`_ij_`的数值。

需要注意的是，这些对象同弦很方便为弦([chord](#))生成器匹配默认的访问器；但仍可以对访问器进行重写或者修改返回对象，实现布局微调。

<#> chord.groups()

给定布局的当前配置和关联矩阵，返回计算过的分组对象。如果分组对象已计算完毕，本方法返回缓存值。如果布局属性有任何改变，则清空之前计算的分组。此时，如果下次调用该方法，需要对布局进行重新计算。返回对象具有下列属性：

- index -行索引，`_i_`。
- startAngle -弧的起始角，在radians内。
- endAngle -弧的终止角，在radians内。
- value -相关行`_i_`的值的总和。

需要注意的是，这些对象同弧度([arc](#))生成器的默认访问器具有较好的吻合度；但仍可以对访问器进行重写或者修改返回对象，实现布局微调。

- 张烁译 20140428
- 咕噜校对 2014-11-30 09:39:57