- 本文档是D3官方文档中文翻译，并保持与最新版同步。
- 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.js:437278817，大数据可视化：436442115
- Github小组：VisualCrew：https://github.com/VisualCrew

比例尺 是将定义域映射为值域的函数 **数值比例尺** 有连续的定义域，例如一系列数字或时间。序数比例尺有离散的定义域，例如一组名称或类别。比例尺在D3中是个可选功能。使用比例尺可以极大地简化从数据纬度到可视化展示的映射。

比例尺对象（例如，由d3.scale.linear返回的对象）的返回值既是一个对象也是一个函数。像D3 中的其他类，比例尺遵循链式语法：setter 方法直接返回比例尺自身，允许在一个简洁的语句中调用多个setter 方法。

线性比例尺

线性比例尺是最常见的比例尺，为连续地把输入域映射到连续的输出范围提供了良好的缺省选择。该映射是线性的，输出范围值*y*可以表示为输入域值*x*的线性函数为：$y = mx + b$。输入域通常是要可视化的数据维度，如学生在样本群的身高(米为单位)。输出范围通常是所需输出的可视化维度，如直方图的中条的高度(以像素为单位)。

状态维数 (state dimensions) 为 domain, range, output interpolator 和 clamping behavior.

d3.scale.**linear**()

用默认域[0,1]构造一个新的比例尺，默认的范围为[0,1]。因此，默认比例尺相当于数字恒等函数；例如linear(0.5)返回0.5。

linear(*x*)

在输入域中的输入*x*，返回输出范围对应的值。

注意：某些插值器(interpolators)会**重用返回值**。例如，如果域值是任意的对象，然后 d3.interpolateObject 自动执行，比例尺重用返回的对象。通常情况下，一个比例尺的返回值被立即用于设置属性(attribute)或样式(style)，你不必担心这一点；但是，如果你需要存储比例尺的返回值，使用字符串强制转换或酌情创建一个副本。

linear.**invert**(*y*)

返回值的输入域*X*在输出范围*y*中的相应值。这代表了逆映射的范围域。在输出范围的有效值y ， linear(linear.invert(*y*))等于*y* ;类似地，在输入域中的有效数值*x*时 ， linear.invert(linear(*x*))等于*x*。等价地，你可以通过建立一个新的比例尺，交换定义域和值域进行反转操作。翻转操作在交互中特别有用的，例如，以确定在对应于该像素位置的鼠标下的输入域的值。

注：如果输出范围是数字只能用倒置操作来支持！ D3允许的输出范围是任何类型的;引擎盖下， d3.interpolate或您选择的自定义插值是用来归一化参数*t*映射到输出范围内的值。因此，输出范围可以是颜色，字符串，或者甚至任意对象。由于没有设施，以"uninterpolate"来标注任意类型，倒置操作目前仅在数值范围的支持。

linear.**domain**([*numbers*])

如果数字*numbers*被指定，设置比例尺的输入域到数字的指定数组。数组必须包含两个或两个以上的数字。如果给定的数组中的元素不是数字，他们将被强制转换为数字;这种强迫发生同样，当规模被调用。因此，线性标尺可用于类型，如日期的对象(date objects)可以被转换成数字编码;但是，它通常是用d3.time.scale的日期更方便。 (您可以使用的valueOf实现自己的转换数量的对象。)如果未指定数字，则返回比例尺的输入域。

虽然线性标度通常只有两个数值在其域中，可以用polylinear比例尺指定两个以上的值。在这种情况下，必须有值的输出范围内的当量数。一个polylinear刻度表示多个分段线性尺度上划分一个连续的定义域和值域。这是用于定义发散的

定量尺度是特别有用的。例如，把白色和红色作为负值，白色和绿色作为正值：

```
var color = d3.scale.linear()
    .domain([-1, 0, 1])
    .range(["red", "white", "green"]);
```

所得到的值的颜色(- 0.5)是RGB(255，128，128)和色彩(0.5)的值是RGB(128，192，128)。在内部，polylinear比例尺通过二分查找对应于给定域值输出插值。通过重复在这两个领域和范围值，你也可以强制输入域的块映射到一个固定的输出范围。

linear.**range**([*values*])

如果值被指定，设置刻度的输出范围值的指定数组。数组必须包含两个或多个值，以匹配输入域的基数，否则第二个的长被截断以匹配另一个。给定的数组中的元素不必是数字;所支持的底层的内插器(interpolator)的任何值都可以。然而，数值范围是必需的倒置运算符。如果未指定*values*的值，则返回当前比例次的输出范围。

linear.**rangeRound**(*values*)

用指定数组设置比例尺的范围值，而比例尺的插补器也设置为d3.interpolateRound。这是用于当输出由刻度值应该是准确的整数，例如，以避免抗混叠伪像一个方便例程。另外，也可以在比例被施加之后手动四舍五入。

linear.**interpolate**([*factory*])

如果指定了工厂*factory*，设置使用指定的工厂规模的输出插值。内插器工厂默认为d3.interpolate，并用于在[0,1]范围内的归一化域参数t在输出范围对应的值。内插器工厂将用于从输出范围构造内插器为每一对相邻的值。如果未指定工厂，返回规模的插补工厂。

linear.**clamp**([*boolean*])

如果布尔变量*boolean*被指定，用于启用或禁用相应地夹集。默认情况下，夹集被禁用，例如，如果输入域之外的值传递给规模，规模可能外面的输出范围，通过线性外推法返回一个值。例如，使用缺省域并[0,1]范围内，与2的输入值将返回2的输出值，如果夹集被启用，归一化的域参数*t*被钳位到范围[0,1]这样规模的返回值总是规模的输出范围之内。如果未指定布尔值，则返回的规模当前夹值输出范围之内。

linear.**nice**([*count*])

它开始和结束圆标记值的扩展域。这种方法通常会修改规模域，并且只能在边界延伸到最近的整数值。根据下列公式计算圆值的精度依赖于域dx的程度：exp(round(log(*dx*)) - 1)。如果该域是从数据计算并可以是不规则，则nice是有用的。例如，对于[0.20147987687960267, 0.996679553296417]一个领域，漂亮的论域为[0.2, 1]。如果域有两个以上的值，nice这个域只影响第一个和最后一个值。

可选的计数参数*count*允许更大的控制权用于扩展边界的步长，保证返回的ticks可完全覆盖域。

linear.**ticks**([*count*])

从比例尺上的输入域的代表值返回约数。如果未指定*count*，则默认为10。返回的刻度值是均匀间隔的，具有人类可读的值(如10的幂的倍数)，并且保证是输入域的范围之内。刻度尺通常用于显示参考线或刻度线，在与可视化数据一起使用。指定的计数*count*是只是一个提示;规模可能取决于输入域返回更多或更少的值。

linear.**tickFormat**(*count*, [*format*])

返回格式化的数字number format用于显示刻度值的值。指定的计数应具有相同的值被用于产生刻度的计数。您不必使用刻度尺的内置刻度格式，但它会自动计算的基础上刻度值之间的固定间隔适当的精度。

可选的格式参数*format*允许指定的格式说明符format specifier，其中格式的精度自动被取代的规模是适当的刻度间隔。例如，要格式化百分比变化，你可能会说：

```
var x = d3.scale.linear().domain([-1, 1]);
console.log(x.ticks(5).map(x.tickFormat(5, "+%"))); // ["-100%", "-50%", "+0%",
"+50%", "+100%"]
```

如果格式已经指定了精度，这种方法相当于d3.format。

同样地，如果*format*使用了格式类型 `s` ，比例尺将基于定义域中的最大值计算SI-前缀，使用SI-前缀的案例：all tick values。如果*format*已经指定了精度，则等价于d3.format。

注意：当结合数轴使用对数比例尺时，通常要用axis.ticks而不是axis.tickFormat，案例：bl.ocks.org/5537697.

linear.**copy**() 返回该线性标尺的精确副本。改变这一比例尺不会影响返回的比例尺，反之亦然。

恒等比例尺

恒等尺是线性尺的一个特例其定义域和值域相同；尺以及它的反转方法都是恒等的。与像素坐标工作时，这些尺度偶尔有用，例如在与轴axis和刷brush组件一起使用。

d3.scale.**identity**()

构造一个新恒等尺，默认定义域[0，1]，默认值域是[0，1]。恒等尺总是等同于恒等函数。

identity(*x*)
identity.**invert**(*x*)

返回给定值*x*。

identity.**domain**([*numbers*])
identity.**range**([*numbers*])

如果指定了数字*numbers*，设置尺度的输入域和输出范围为指定的数字数组。这个数组必须包含两个或两个以上的数字。如果给定的数组中的元素不是数字，他们将被强制转换为数字; 这种强制转换在尺scale被调用时同样发生。 如果未指定数字，则返回尺的当前输入定义域(或等价地，输出范围)。

identity.**ticks**([*count*])

大致返回刻度的输入域(或等价地，输出范围)*count* 代表性的值。如果未指定*count*，则默认为10。返回的刻度值是均匀间隔的，具有人类可读的值(如10的幂的倍数)，并且保证是输入域的范围之内。刻度线通常用于显示在与可视化数据一起使用的参考线或刻度线。指定的计数*count*是只是一个提示;刻度取决于输入域可能返回更多或更少的值。

identity.**tickFormat**(*count*, [*format*])

返回适合用于显示刻度值的数字格式number format函数。指定的计数*count* 应和用于产生刻度值的计数的值相等。您不必使用尺度的内置刻度格式，但它会基于刻度值之间的固定间隔自动计算适当的精度。

可选参数*format*允许指定格式说明符format specifier。如果格式说明符没有一个定义的精度，精度将由尺自动设置，并返回相应的格式。这提供了方便，可用声明的方式指定一个格式其精度将由尺度自动设置。

identity.**copy**()

返回此尺度的精确副本。改变这一尺度不会影响返回的尺度，反之亦然。

乘方比例尺

除了有一个在输出范围值被计算之前应用于输入值域的指数变换之外，乘方比例尺类似于线性比例尺。映射到输出范围值*y*可以被表示为一个输入值域*x*的函数：*y* = *mx*^*k* + *b*,其中*k*是指数的值。乘方比例尺也支持负值，在这种情况下，

输入值权力天平也支持负值,在这种情况下,输入值以指数-1进行乘方计算,由此产生的输出值也将以-1来乘方计算。

d3.scale.**sqrt**()

构造一个新的乘方比例尺，输入值域默认为[0,1],输出范围默认为[0,1],指数默认为0.5。这种方法可以简记为:

```
d3.scale.pow().exponent(.5)
```

返回的比例尺是一个函数,它接受一个代表输入值域中某一个值的参数x,返回值是对应于输出范围的一个值。因此,这个比例尺对于数字来说，相当于函数sqrt的功能；例如:sqrt(0.25)的结果将返回0.5。

d3.scale.**pow**()

构造一个新的乘方比例尺，默认输入值域为[0,1],默认输出范围[0,1],默认指数为1。因此,对于数字来说，默认的乘方比例尺相当于数字正比例函数(the identity function);例如pow(0.5)的结果将返回0.5。

pow(*x*)

给定一个输入域值*x*,返回输出范围中相应的值。

注意:一些interpolators*重用返回值*。例如,如果值域是任意对象,然后d3.interpolateObject自动被应用并且比例尺重用这个返回的对象。通常,一个比例尺的返回值会立刻被应用于设置一个属性(attribute)或样式(style),你不必担心；不过,如果你需要存储比例尺的返回值,可以适当地使用字符串强转换或复制值。

pow.**invert**(*y*)

返回对应于输出域值*y*值的输入值域值*x*。这意味着从输出范围到输入值域映射关系的逆向操作。对于一个有效的输出范围值*y*，`pow(pow.invert(y))` 的结果等于*y*，对于一个有效的输入域值*x*，`pow.invert(pow(x))` 的结果等于*x*。同样，你可以通过交换输出与输入值域的新比例尺的建立去创建其逆操作。对于确定输入值域的值(此值对应于鼠标下的像素位置)而言，这个逆操作对于交互尤为重要。

注意:逆操作仅仅支持输出范围是数字的情况！D3是允许输出范围属于任意类型；在其内部结构中，d3.interpolate或者一个自定义的插值器是用来映射这个标准化参数*t*到一个输出范围的值。因此，输出范围可能是颜色值、字符串、甚至任意的对象。

pow.**domain**([*numbers*])

如果参数*numbers*被指定，便将指定的数组设置为比例尺的输入值域。这个数组必须包含2个或更多的数据值。如果给定的数组中的元素不是数字，它们便被强转为数字；当比例尺被调用时强转同样会发生。因此，一个乘方比例尺被用来编码可以被转换为数字的任意类型。如果*numbers*没有被指定，将返回比例尺当前的输入值域。

与线性比例尺(详见linear.domain)一样,对于输入和输出值域乘方比例尺也可以接受超过两个以上的值,因而产生结果的polypower比例尺。

pow.**range**([*values*])

如果*values*被指定,便将指定包含*values*的数组设置为比例尺的输出范围。这个数组必须包含两个或两个以上的值,来对应匹配输入域的基数,否则多于两个以上的将被截断从而去对应其它值。给定数组中的元素不要求必须为数字;支持基本插值器(interpolator)的任何值都可以使用。然而,数字范围需要转化操作符去转化。如果*values*没有被指定,将返回当前的输出范围。

pow.**rangeRound**(*values*)

将指定的values组成的数组设置为输出范围，同时设置比例尺的插值器到d3.interpolateRound。当比例尺输出的值需要为整数时，这便是一个非常方便使用套路，以此避免不确定性的小数产生。也可以在比例尺应用后通过手动将输出值处理为整数。

pow.**exponent**([*k*])

如果*k*值被指定值,当前指数将被设置为*k*值。如果*k*值没有被指定,则返回当前指数。默认值是1。

pow.**interpolate**([*factory*])

如果工厂*factory*被指定,便将此指定的工厂*factory*设置为比例尺的输出插值器。插值器工厂默认用d3.interpolate,并且它被用来将标准的[0,1]的值域参数t映射到相关的输出值域中对应的值。这个插值器工厂将针对输出值域中每两个相邻的值来用于构建插值器。如果没有指定工厂*factory*,将返回比例尺的插入器工厂。

pow.**clamp**([*boolean*])

如果布尔值*boolean*被指定,启用或禁用相应的闭合。默认情况下,是启用闭合,这样的情况下,如果一个超出输入值域的值在传递给比例尺时，比例尺将通过线性外推法返回一个输出范围外的一个值。例如,使用输入和输出值域采用默认值域[0,1],那么对于输入值为2的就会返回一个输出值2。如果启用了闭合,标准的值域参数*t*将被闭合在值域区间[0,1]中，这样的情况下，比例尺的返回值总是在比例尺的输出范围内。如果布尔值*boolean*不被指定,返回值都将在输出范围内(returns whether or not the scale currently clamps values to within the output range)。

pow.**nice**([*m*])

扩展值域,即它的开始和结束值都处理为容易识别的整数值。这种方法通常来修改比例尺的值域,可能只是将临界的值处理为最近的整数值。这个整数值的精度依赖于值域*dx*的范围，根据下列的公式：exp(round(log(*dx*)) - 1)。如果值域是通过不规则的值计算而来的，那么这样处理(使用本函数)是很有用的。例如,对于一个值域为[0.20147987687960267,0.996679553296417],那么好的值域(可以通过本函数处理)便是(0.2,1)。如果值域拥有超过两个的值,那么这样处理后只能影响到第一个和最后一个值。

可选参数*m*允许指定一个滴答计数用来被指定用来在扩展边界值之前去控制步长使用。

pow.**ticks**([*count*])

近似地返回个数*count*来自于比例尺输入值域的代表数。如果*count*没有被指定,默认为10。返回的值都有均匀一致的间隔,拥有人类可读的值(如乘方指数为10的倍数),并保证是在输入值域的范围内。刻度值(Ticks)通常用于结合可视化数据去显示参考线,或刻度线。指定的*count*只是一个暗示；比例尺可能会返回值或多或少，这取决于输入值域。

pow.**tickFormat**([*count*, [*format*]])

返回一个数字格式化(number format)函数，适用于显示一个点值。被指定的count数量值与被用来生刻度值的数量相同。你不必使用比例尺的内置的刻度值格式化,但它能够基于固定时间间隔值去自动计算出适当的精度。 可选参数format允许为被指定的格式说明符(format specifier)。如果格式说明符没有明确定义的精度,那么精度将通过比例尺被自动设置，返回适当的格式。这提供了一个便捷的、声明的方式去指定一个格式，此格式的精度将通过比例尺被自动设定。

pow.**copy**()

返回一个比例尺的精确复制体。这个比例尺的变化不会影响到返回的复制体比例尺,反之亦然。

对数比例尺

对数比例尺类似于线性比例尺，除了有一个对数变换被应用于在输出范围值计算的输入值域的值。映射到输出范围的*y*值可以表示为一个输入值域*x*的函数: $y = m\log(x) + b$。

log(0)是负无穷大，一个对数比例尺必须有一个专门的正域或负域；这个范围不能包含或过零。一个正域对数比例尺有一个定义明确的正值的行为状态，一个负域对数比例尺有一个定义明确的负值的行为状态(输入值乘以-1，所得到的输出值也乘以-1)。如果你传递一个负值给对数比例尺是正域，反之比例尺的行为是未定义的。

d3.scale.**log**()

构建一个默认域[1,10]新的对数比例尺，默认范围[0,1]，底数是10。

log(*x*)

给一个在输入定义域中的*x*值，返回一个输出值域的相应值。

注：一些内插器(interpolators)**重用返回值**。例如，如果域的值是任意对象，那么d3.interpolateObject是自动应用和按比例返回的对象。通常情况下，一个比例尺的返回值被立即用于设置属性(attribute)或样式(style)，你不必担心这一点;但是，如果你需要存储比列尺的返回值，使用字符串控制或创建一个合适的副本。

log.**invert**(*y*)

返回值的输入值域*X*在输出范围*y*中的相应值。这代表了逆映射的范围域。在输出范围有效的*y*值，log(log.invert(*y*))等于*y*;同样，在输入值域中的有效值*x*，log.invert(log(*x*)) 等于*x*。相同地，你可以通过建立一个新的比例尺，对变换输入值域和输出范围进行反转操作。反转操作是交互特别有用的，例如，以确定在对应于该像素位置的鼠标下的输入值域的值。

注：反转操作仅支持输出范围是数字！ D3允许的输出范围是任何类型;在后台，d3.interpolate或您选择的自定义插值是用来归一化参数*t*映射到输出范围内的值。因此，输出范围可以是颜色，字符串，或者甚至任意对象。由于没有便利以"篡改"任意类型，反转操作目前仅在数值范围的支持。

log.**domain**([*numbers*])

如果指定了数字*numbers*，设置比例尺的输入值域给数字的指定数组。数组必须包含两个或两个以上的数字。如果给定数组中的元素不是数字，他们将被强制转换为数字;这种强制同样发生在当比例尺被调用。因此，一个对数比例尺可用于能被转换成数字的任何类型。如果数字未指定，则返回比例的当前输入值域。

如线性比例尺(见linear.domain)，对数比例尺也可以接受两个以上值的值域和范围，从而得到聚对数函数。

log.**range**([*values*])

如果值*values*是指定的，要通过指定数组的值设置比例尺的输出范围。数组必须包含两个或多个值，以匹配输入值域的底数，否则两个中长的会被截断以匹配另一个。给定的数组中的元素不必是数字;所支持的底层的内插器(interpolator)的任何值都可以。然而，数值范围是必需的倒置运算符。如果未指定*values*值，则返回比例尺的当前输出范围。

log.**rangeRound**(*values*)

设置比例尺的输出范围要通过指定数组的值，而比例尺的内插器也设置为d3.interpolateRound。这是用于输出的比例尺的值应该是准确的整数，以避免抗混叠伪像的一个方便例程。另外，当输出值的比例尺被应用之后也可以手动设置四舍五入求值。

log.**base**([*base*])

如果指定了底数*base*，设置这个对数比例尺的底数。如果未指定底数，返回当前的底数，默认为10。

log.**interpolate**([*factory*])

如果工厂模式*factory*是指定的，设置比例尺的输出内插器使用指定的工厂模式，内插器的工厂模式默认为d3.interpolate，并用于在[0,1]的归一化域参数*t*映射到输出范围对应的值。内插器工厂模式将用于从输出范围创建内插器为每一对相邻的值，如果没有指定的工厂模式时，返回比例尺的内插器的工厂模式。

log.**clamp**([*boolean*])

如果布尔值*boolean*是指定的，启用或禁用相应的锁定。默认情况下，锁定被禁用，例如，如果输入值域之外的值传递给比例尺，比例尺可能返回外面的输出范围，通过线性外推法返回一个值。例如，使用默认值域并在[0,1]范围内，一个2的输入值将返回2的输出值，如果锁定被启用，归一化的域参数*t*被锁定到范围[0,1]，这样比例尺的返回值始终在比例尺的输出范围之内。如果未指定布尔值，则返回比例尺当前锁定值是否在输出范围内。

log.**nice**()

扩展值域以便它开始和结束美化四舍五入的值。这种方法通常会修改比例尺的值域，并可能只有边界延伸至最近的四舍五入的值。最近的整数值是基于比例尺的底数(base)的整数幂，默认为10。美化是有用的，如果该值域是从数据计算的并可以是不规则的。例如，对于值域[0.20147987687960267，0.996679553296417]，美化的值域为[0.1，1]。如果值域有两个以上的值，美化值域只影响第一个和最后一个值。

log.**ticks**()

从比例尺上的输入值域将返回代表值。返回的刻度值是均匀间隔在10的每个幂的范围内，并保证是输入值域的范围之内。刻度通常用于显示参考线或刻度线，与可视化数据一起使用。请注意，刻度的数量不能自定义(由于对数比例尺的性质);但是，如果你想减少刻度的数量，你可以筛选返回的数组中的值。

log.**tickFormat**([*count*, [*format*]])

返回一个数字格式(number format)的函数适合用于显示一个刻度值。这个返回刻度格式是为实现 `d.toPrecision(1)` 。如果一个计数*count*被指定，那么一些刻度标记可能无法显示; 如果没有足够的空间来容纳所有的刻度标记，这是非常有用的。但是，请注意刻度线仍然会显示出来(这样对数比例尺失真仍然可见)。当指定一个计数，你也可能会覆盖这个格式函数*format*; 您也可以指定一个格式说明符的字符串，它会自动被包裹在d3.format。例如，要获得一个刻度格式，将显示20个刻度的货币：

```
scale.tickFormat(20, "$,.2f");
```

如果格式说明符没有一个定义的精度，精度将通过比例尺自动设置，返回适当的格式。这提供了一个便利，既指定其精度格式的声明方式将由比例尺自动设定。

log.**copy**()

返回此比例尺的精确副本。改变这一比例尺不会影响返回的比例尺，反之亦然。

量化比例尺

量化比例尺是线性比例尺的变体，定义域是连续的，值域是离散的。定义域基于值域的份数将被划分为同样的份数。这种线性关系可以表示为： $y = mx + b$。定义域是用来可视化的数据维度，例如，人口样本中的一组学生的高度（以米为单位）.值域是输出的可视化维度，例如条形图中的条高（以像素为单位）。

d3.scale.**quantize**()

使用默认的定义域[0,1]，定义一个量化变换，默认的范围是[0,1]；因此，默认的量化变换等同于数值的四舍五入round方法；例如：quantize(0.49)返回0，quantize(0.51)返回1，详见下例：

```
var q = d3.scale.quantize().domain([0, 1]).range(['a', 'b', 'c']);
//q(0.3) === 'a', q(0.4) === 'b', q(0.6) === 'b', q(0.7) ==='c';
//q.invertExtent('a') returns [0, 0.3333333333333333]
```

quantize(*x*)

指定一个值*x*作为输入域值，返回对应该域值的范围值。 理解：quantize(0.4)则会使用默认的值域0和1，即：该quantize(0.4)返回值为0，因为四舍五入0.4为0，以此类推。

quantize.**invertExtent**(*y*)

根据输出的范围值*y*返回对应于定义域中的取值范围，即为数组[*x0, x1*]，该功能即：反向找到值*y*对应的定义域；这是非常有用的，例如：为了确定对应于该像素的鼠标的可取输入域；

quantize.**domain**([*numbers*])

如果*numbers*指定，则设置变换的定义域/输入域为*numbers*，该numbers是两个元素的数组，即代表定义域的开始和结束；如果该数组的元素多于2个数值，则只会取第一个元素为开始，最后一个元素为结束；如果该数组中元素不是数值，则会强制转换为数值；强制转换会在调用该scale 转换时发生；因此，定量变换可被用于任何可强转成数值的编码/参数类型；如果numbers未指定，则返回当前的定义域/输入域；

quantize.**range**([*values*])

如果*values* 指定，则设置变换的值域/输出域为*values* ，该*values* 可以是任何类型的数组元素，并且元素的数量不限；如果values 未指定，则返回当前的值域/输出域。

quantize.**copy**()

返回当前变换的一个副本，无论是改变原本或副本，都不会影响彼此。

分位数比例尺

分位数比例尺映射输入域到离散的的范围。即使输入域是连续的比例尺接受任意合理的输入值，输入域指定为一组离散的值。输出范围的值的数量(基数)决定输入域中将被计算的分位数数量。为了计算分位数，输入域是排好序的，并且被当成离散值的群体(population of discrete values)。输入域通常是你想要可视化的数据维度，例如股票市场每天的变化。输出范围通常是需要输出的可视化，例如一个发散的颜色尺。

d3.scale.**quantile**()

构造一个新的分位数比例尺，使用空的输入域和输出范围。分位数比例尺在输入域和输出范围没有同时指定时无效。

quantile(*x*)

给定输入域中的值*x*，返回输出范围中对应的值。

quantile.**invertExtent**(*y*)

为对应输出范围中的值*y*，返回输入域[*x0*, *x1*]的范围，代表从范围到域逆向映射。这个方法在交互时是很有用的。用来决定输入域中的值对应鼠标下的像素位置。

quantile.**domain**([*numbers*])

如果指定了参数*numbers*，设置分位数比例尺的输入域为指定的离散数值集。数组必须是非空的，并且必须包含至少一个数值；`NaN`，`null`和`undefined`是被忽略的，不被视为样本群体的一部分。如果给定数组中的元素不是数字，将被强制转换为数字；当比例尺被调用的时候强制转换。这样，分位数比例尺可以用来强制转换所有可以转换为数字的类型。如果*numbers*参数没有指定，就返回比例尺当前的输入。

quantile.**range**([*values*])

如果指定了*values*参数，就设置输出范围中的离散值。数组必须是非空的，可能包含任意类型的值。*values*数组中值得数量(基数或长度)决定了将被计算的分位数的数量。例如，为了计算分位数，values必须是一个像[0, 1, 2, 3] 的四元素的数组。如果没有指定*values*参数，返回当前的输出范围。

quantile.**quantiles**()

返回分位数阈值。如果输出范围包含*n*个离散值，返回的阈值数组就包含*n*-1个值。值小于阈值数组的第一个元素，quantiles()[0]视为在第一个分位；更大的值第二个阈值在第二个分位，等等。在内部，阈值数组用于d3.bisect查找给定输入值对应的输出分位。

quantile.**copy**()

返回这个比例尺的一个精确的副本。比例尺的改变不影响返回的比例尺，反之亦然。

临界值比例尺

临界值比例尺很像定量 quantize比例尺，只是他允许你将输入域的任意的子集合映射到离散的输出值上。输入值是连续的，并且基于一组临界值被划到不同的区域。输入值域通常是你想要用来可视化的数据的尺度，例如是一群学生的身高。输出值域通常是所需的输出可视化数据的尺度，。例如是一组颜色(用字符串表示)。

d3.scale.**threshold**()

构造一组新的临界值比例尺，默认的输入范围是[.5]，默认的输出范围是[0,1]，因此，对数字来说，默认quantize比例尺将和round函数相同，例如，输入0.49将输出0，输入0.51将输出1.

```
var t = d3.scale.threshold().domain([0, 1]).range(['a', 'b', 'c']);
t(-1) === 'a';
t(0) === 'b';
t(0.5) === 'b';
t(1) === 'c';
t(1000) === 'c';
t.invertExtent('a'); //returns [undefined, 0]
t.invertExtent('b'); //returns [0, 1]
t.invertExtent('c'); //returns [1, undefined]
```

threshold(*x*)

*x*代表输入域值，函数将返回对应的输出范围的值。

threshold.**invertExtent**(*y*)

返回输出范围*y*中值对应的输入域值的范围，代表输出值到输入值的一个反向映射，这个方法对于交互非常有用，例如可以用来定位鼠标的像素点所在的区域。

threshold.**domain**([*domain*])

如果指定了*domain*参数，那么将这个数组形式的参数作为输入域，参数数组的值必须是升序排列的，否则比例尺将是不确定的，这个参数数组的值通常是数字，但是任何自然排序的值(例如字符串)都可以，因此一个临界值比例尺适用于任何可以排序的类型。如果输出范围是N+1那么输入值范围一定是N，如果小于N的元素在输入值里，那么这个额外的元素将被忽略，如果大于N的元素在输入值里，那么对于一些输入值比例尺将返回未定义，如果输入域未定义，将返回这个比例尺当前的输入值

threshold.**range**([*values*])

如果存在参数*values* ，那么将这个数组形式的参数作为输出域。如果输入域是N，那么输出范围必须是N+1，如果小于N+1的元素在输出值中，那么对于任何输入比例尺将返回undefined，如果大于N+1的元素在输出值中，那么这个额外的值将被忽略，数组中的元素并不一定是数字，任何形式都可以，如果输出域*values*未定义，那么将返回比例尺当前的输出值。

threshold.**copy**()

复制这个比例尺的精确副本。改变这个的比列尺不会影响返回值，反之亦然。

- 补充知识

	人员	组织	时间
翻译	谭树国： Linear	VisualCrew小组	20140412
	大傻： Identity	VisualCrew小组	20140412
	马语者 : Log	VisualCrew小组	20140412
	WeiFei365 : Quantize	VisualCrew小组	20140412
	现明涟漪： Threshold	VisualCrew小组	20140420
	Harry： Power	VisualCrew小组	20140412
	大傻： Quantile Scales	VisualCrew小组	20141124
校对	大傻	VisualCrew小组	20141124
	大傻	VisualCrew小组	2016-4-6 21:13:05
排版	liang42hao	VisualCrew小组	2016-02-18 15:48:07