# Quantitative Scales

[Wiki](https://github.com/mbostock/d3/wiki) ▸ [API Reference](https://github.com/mbostock/d3/wiki/API-Reference) ▸ [Scales](https://github.com/mbostock/d3/wiki/Scales) ▸ **Quantitative Scales**

 [d3.scale.sqrt](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-sqrt) - 创建一个求平方根的定量转换。

 [d3.scale.pow](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-pow) - 创建一个指数变换。（可参考linear对应函数的注释）

 [pow](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-_pow) - 输入一个定义域的值，返回一个值域的值。

 [pow.invert](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-pow_invert) - 反变换，输入值域值返回定义域值。

 [pow.domain](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-pow_domain) - get或set定义域。

 [pow.range](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-pow_range) - get或set值域。

 [pow.rangeRound](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-pow_rangeRound) - 设置值域，并对结果取整。

 [pow.interpolate](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-pow_interpolate) - get或set变换的插值函数。

 [pow.clamp](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-pow_clamp) - 设置值域是否闭合，默认不闭合。

 [pow.nice](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-pow_nice) - 扩展定义域范围使定义域更规整。

 [pow.ticks](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-pow_ticks) - 从定义域中取出有代表性的值。通常用于坐标轴刻度的选取。

 [pow.tickFormat](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-pow_tickFormat) - 获取格式转化函数，通常用于坐标轴刻度的格式转化。

 [pow.exponent](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-pow_exponent) - get或set指数的幂次。默认为1次幂。

 [pow.copy](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-pow_copy) - 从已有的变换中复制出一个变换。

## Power Scales(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。

#### [#](https://github.com/mbostock/d3/wiki/Quantitative-Scales" \l "wiki-sqrt) 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](https://developer.mozilla.org/en/JavaScript/Reference/Global_Objects/Math/sqrt)的功能；例如: sqrt(0.25)的结果将返回0.5。

#### [#](https://github.com/mbostock/d3/wiki/Quantitative-Scales" \l "wiki-pow) 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。

#### [#](https://github.com/mbostock/d3/wiki/Quantitative-Scales" \l "wiki-_pow) 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](https://github.com/mbostock/d3/wiki/Quantitative-Scales" \l "wiki-pow_interpolate)的重用将返回值。例如,如果值域是任意对象,然后[d3.interpolateObject](https://github.com/mbostock/d3/wiki/Transitions" \l "wiki-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](https://github.com/mbostock/d3/wiki/Selections#wiki-attr))或样式([style](https://github.com/mbostock/d3/wiki/Selections#wiki-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。

#### [#](https://github.com/mbostock/d3/wiki/Quantitative-Scales" \l "wiki-pow_invert) 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。（The invert operator is particularly useful for interaction, say to determine the value in the input domain that corresponds to the pixel location under the mouse.）

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](https://github.com/mbostock/d3/wiki/Transitions#wiki-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。

转化运算符仅支持如果输出范围是数字!D3允许任何类型的输出范围;在引擎盖下,d3.interpolate或您选择的自定义插入器是用来映射值的归一化参数t的输出范围。因此,输出范围可能是颜色,字符串,甚至任意对象。所以对于“非差值”的任意类型是没有任何措施的，逆操作目前仅仅支持数字范围。（As there is no facility to "uninterpolate" arbitrary types, the invert operator is currently supported only on numeric ranges.）

#### [#](https://github.com/mbostock/d3/wiki/Quantitative-Scales" \l "wiki-pow_domain) pow.domain([*numbers*])

如果参数numbers被指定，便将指定包含numbers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](https://github.com/mbostock/d3/wiki/Quantitative-Scales" \l "wiki-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。（As with linear scales (see [linear.domain](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-linear_domain)), power scales can also accept more than two values for the domain and range, thus resulting in polypower scale.）

#### [#](https://github.com/mbostock/d3/wiki/Quantitative-Scales" \l "wiki-pow_range) 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;支持基本插值器的任何值都可以使用（any value that is supported by the underlying [interpolator](https://github.com/mbostock/d3/wiki/Quantitative-Scales#wiki-pow_interpolate) will work.）。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。

#### [#](https://github.com/mbostock/d3/wiki/Quantitative-Scales" \l "wiki-pow_rangeRound) 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](https://github.com/mbostock/d3/wiki/Transitions#wiki-d3_interpolateRound)(while also setting the scale's interpolator to [d3.interpolateRound](https://github.com/mbostock/d3/wiki/Transitions#wiki-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（to avoid antialiasing artifacts）。也可以在比例尺应用后通过手动将输出值处理为整数。

#### [#](https://github.com/mbostock/d3/wiki/Quantitative-Scales" \l "wiki-pow_exponent) 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。

#### [#](https://github.com/mbostock/d3/wiki/Quantitative-Scales" \l "wiki-pow_interpolate) pow.interpolate([*interpolator*])

如果工厂（interpolator）被指定,便将此指定的工厂设置为比例尺的输出插值器。插值器工厂默认用[d3.interpolate](https://github.com/mbostock/d3/wiki/Transitions#wiki-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,将返回比例尺的插入器工厂。

#### [#](https://github.com/mbostock/d3/wiki/Quantitative-Scales" \l "wiki-pow_clamp) 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）。

#### [#](https://github.com/mbostock/d3/wiki/Quantitative-Scales" \l "wiki-pow_nice) 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（Extends the domain so that it starts and ends on nice round values）。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（This method typically modifies the scale's domain, and may only extend the bounds to the nearest round value）。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（The optional *m* argument allows a tick count to be specified to control the step size used prior to extending the bounds）。

#### [#](https://github.com/mbostock/d3/wiki/Quantitative-Scales" \l "wiki-pow_ticks) pow.ticks([*count*])

近似地返回几个（count：个数）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(such as multiples of powers of 10)),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。指定的数量只是一个暗示；比例尺可能会返回值或多或少，这取决于输入值域。

#### [#](https://github.com/mbostock/d3/wiki/Quantitative-Scales" \l "wiki-pow_tickFormat) pow.tickFormat([*count*, [*format*]])

返回一个数字格式化（[number format](https://github.com/mbostock/d3/wiki/Formatting#wiki-d3_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](https://github.com/mbostock/d3/wiki/Formatting#wiki-d3_format)）。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。

#### [#](https://github.com/mbostock/d3/wiki/Quantitative-Scales" \l "wiki-pow_copy) 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。

Harry

2014年4月12日 星期六