**Time Scales (Time比例尺)**

D3的Time scale是d3.scale.linear比例尺的扩展，使用Javascript的Date对象作为其输入域（domain）。因此，不同于普通的线性比例尺（linear scale）,domain的值会被强制转为时间类型而非数字类型；同样的，invert函数返回的是一个时间类型。最方便的是，time scale同样提供了基于time intervals的ticks，免了为任何基于时间域的轴。

d3.time.scale返回的是一个scale对象，同时也是一个函数。你可以像任何函数一样引用它，同时它有额外的方法可以改变其呈现。如D3的其它类一样，scale对象也可以链式调用方法，setter方法返回scale对象本身，可以用简洁的声明中引用多重的setters。

[#](https://github.com/mbostock/d3/wiki/Time-Scales" \l "wiki-scale) d3.time.**scale**()

使用默认的domain和range来构建一个新的time scale；默认以本地时间配置ticks和tick format。

[#](https://github.com/mbostock/d3/wiki/Time-Scales" \l "wiki-utc) **d3.time.scale.utc()**

使用默认的domain和range来构建一个新的time scale；默认以UTC时间配置ticks和tick format。

[#](https://github.com/mbostock/d3/wiki/Time-Scales" \l "wiki-_scale) **scale(x)**

输入输入域domain内的时间对象x，返回相应输出域range的值;这里的scale就是d3.time.scale()返回的对象或函数。

[#](https://github.com/mbostock/d3/wiki/Time-Scales" \l "wiki-invert)**scale.invert(y)**

输入输入域range内的y值，返回相应的输入域domain内的时间对象x。这个用于从rang到domain的反向映射。对于输出域range内的y值，scale(scale.invert(y))==y；同样，对于输入域domain内的时间对象x，scale.invert(scale(x))==x。这种反向操作符在交互时非常有用，例如，鼠标移动时，计算出鼠标当前位置相对应的输入域的时间。

[#](https://github.com/mbostock/d3/wiki/Time-Scales" \l "wiki-domain) **scale.domain([dates])**

如果设置了参数dates ，则把scale的输入域domain设置为dates数组，该数组必须包含两个及以上的时间对象。如果数组里的元素不是时间对象，将被强制转换为时间对象；这个强制转换发生scale被引用时。

如果没有设置参数dates，则返回scale的当前的输入域domain。通常来说，time scales虽然在输入域domain里只有两个时间对象，但你可以设置多于两个的时间对象用于polylinear scale，这样的话，必须在输出域range中设置相同数量的数值。

**[#](https://github.com/mbostock/d3/wiki/Time-Scales" \l "wiki-nice) scale.nice([interval[, step]])**

[**#**](https://github.com/mbostock/d3/wiki/Time-Scales#wiki-nice)**scale.nice([count])**

工

**[#](https://github.com/mbostock/d3/wiki/Time-Scales" \l "wiki-range) scale.range([values])**

如果设置了参数values，则把scale的输出域range设置为values数组，该数组必须包含两及以上的数值，以匹配输入域domain。values数组里的元素不需要一定是数字类型，任何支持下面的interpolator的数据类型都可以。但是，invert操作符需要数字类型的输出域。如果没有设置参数values，则返回scale的当前输出域。

**[#](https://github.com/mbostock/d3/wiki/Time-Scales" \l "wiki-rangeRound) scale.rangeRound([values])**

设置scale的输出域range为指定的values数组，同时设置了scale的interpolator为d3.interpolateRound。如果scale输出的值应该是整数时，这是一个便捷的方法，可以有效地避免抗锯齿。也可以在引用scale之后手动四舍五入输出的值。

**[#](https://github.com/mbostock/d3/wiki/Time-Scales" \l "wiki-interpolate) scale.interpolate([factory])**

如果设置了参数factory，则设置scale的输出域的interpolate使用参数factory。

[#](https://github.com/mbostock/d3/wiki/Time-Scales#wiki-interpolate) scale.**interpolate**([*factory*])

If factory is specified, sets the scale's output interpolator using the specified factory. The interpolator factory defaults to [d3.interpolate](https://github.com/mbostock/d3/wiki/Transitions#wiki-d3_interpolate), and is used to map the normalized domain parameter t in [0,1] to the corresponding value in the output range. The interpolator factory will be used to construct interpolators for each adjacent pair of values from the output range. Iffactory is not specified, returns the scale's interpolator factory.

如果设置了参数factory，则将scale的输出interpolator设置为factory。默认的interpolator factory为[d3.interpolate](https://github.com/mbostock/d3/wiki/Transitions#wiki-d3_interpolate)，将标准化的输入域参数[0,1]范围内的t，映射到相应的输出域range内的数值。interpolator factory将为输出域内的相邻的每对数值构造interpolators。

[#](https://github.com/mbostock/d3/wiki/Time-Scales" \l "wiki-clamp) scale.**clamp**([*boolean*])

If boolean is specified, enables or disables clamping accordingly. By default, clamping is disabled, such that if a value outside the input domain is passed to the scale, the scale may return a value outside the output range through linear extrapolation. For example, with the default domain and range of [0,1], an input value of 2 will return an output value of 2. If clamping is enabled, the normalized domain parameter t is clamped to the range [0,1], such that the return value of the scale is always within the scale's output range. If boolean is not specified, returns whether or not the scale currently clamps values to within the output range.

如果设置了参数boolean（布尔值），则相应地开启或者关闭clamping。默认情况下，clamping是关闭的，如果一个超出了输入域domain的值被传递给了scale，则scale将通过线性推断，可能会返回一个输出域range之外的数值。以默认的[0，1]输入域和输出域为例，输入数值2将返回输出数值2。如果clamping开启，标准化的输入域参数t会被限定在输出域[0，1]内，这样scale返回的值永远都在输出域内。如果boolean没有设置，返回的是scale是否限定当前数值到输出域range内的布尔值。

[#](https://github.com/mbostock/d3/wiki/Time-Scales" \l "wiki-ticks) scale.**ticks**([*interval*[, *step*]])   
[#](https://github.com/mbostock/d3/wiki/Time-Scales#wiki-ticks) scale.**ticks**([*count*])

Returns representative dates from the scale's input domain. The returned tick dates are uniformly spaced (modulo irregular time intervals, such as months and leap years), have human-readable values (such as midnights), and are guaranteed to be within the extent of the input domain. Ticks are often used to display reference lines, or tick marks, in conjunction with the visualized data.

返回scale输入域的代表性时间。返回的tick时间，它们有相同间距（模式不标准的时间间隔，比如月份和闰年）、包含可读性的数值（比如午夜），以及确保在输入域domain范围之内。**Ticks常被用在显示参考线、tick标记，以连接视觉化的数据。**

If count is a number, then approximately count ticks will be returned. If count is not specified, it defaults to 10. The specified count is only a hint; the scale may return more or fewer values depending on the input domain. If a [time interval](https://github.com/mbostock/d3/wiki/Time-Intervals) is specified, then the time interval’s [range function](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-interval_range) will be used to generate ticks, being passed the optional step argument, if any. For example, to create ten default ticks, say:

如果参数count是一个数字，将返回大约count数量的ticks。如果未设置参数count，默认设置为10。Count只是一个提示，根据输入域domain，scale可能会返回更多或更少的数值。

如果设置了参数[time interval](https://github.com/mbostock/d3/wiki/Time-Intervals) ，那么将会引用time interval的[range function](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-interval_range) 函数，同时传递可选参数step，以生成ticks。

比如，生成默认的10个ticks:

scale.ticks(10);

While to create ticks at 15-minute intervals, say:

或者，以15分钟为间隔生成ticks：

scale.ticks(d3.time.minute,15);

Note: for UTC scales, be sure to use the appropriate UTC range method (such as d3.time.minute.utc).

注意：对于UTC scales，使用相应的UTC输出域range方法（比如，d3.time.minute.utc）。

The following time intervals are considered for automatic ticks:

以下时间间隔被视为自动ticks：

* 1-, 5-, 15- and 30-[second](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-second).
* 1-, 5-, 15- and 30-[minute](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-minute).
* 1-, 3-, 6- and 12-[hour](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-hour).
* 1- and 2-[day](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-day).
* 1-[week](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-week).
* 1- and 3-[month](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-month).
* 1-[year](https://github.com/mbostock/d3/wiki/Time-Intervals#wiki-year). 比如：

This set of time intervals is somewhat arbitrary and additional values may be added in the future.

这一时间间隔系列是一些随意和附加的数值，未来可能会增加。

注：Scale.ticks()的用法是为了计算输入域domain的区间结果

以下面的time scale为例：

var x = d3.time.scale()

.domain([new Date(2013,0,1),new Date(2014,0,1)])

.range([0,500]);

这样的x scale 就正常解析，如果后面还增加.ticks()，将不能正常解析，如下

var x = d3.time.scale()

.domain([new Date(2013,0,1),new Date(2014,0,1)])

.range([0,width])

.ticks(d3.time.month,6);

以上方式将报错。

所以，x.ticks()---->返回的是一个13长度的时间数组

[Tue Jan 01 2013 00:00:00 GMT+0800 (CST),

Fri Feb 01 2013 00:00:00 GMT+0800 (CST),

Fri Mar 01 2013 00:00:00 GMT+0800 (CST),

Mon Apr 01 2013 00:00:00 GMT+0800 (CST),

Wed May 01 2013 00:00:00 GMT+0800 (CST),

Sat Jun 01 2013 00:00:00 GMT+0800 (CST),

Mon Jul 01 2013 00:00:00 GMT+0800 (CST),

Thu Aug 01 2013 00:00:00 GMT+0800 (CST),

Sun Sep 01 2013 00:00:00 GMT+0800 (CST),

Tue Oct 01 2013 00:00:00 GMT+0800 (CST),

Fri Nov 01 2013 00:00:00 GMT+0800 (CST),

Sun Dec 01 2013 00:00:00 GMT+0800 (CST),

Wed Jan 01 2014 00:00:00 GMT+0800 (CST)]

x.ticks(d3.time.month,6),返回的是以6个月为间隔或步长均分输入域domain的时间数组：

[Tue Jan 01 2013 00:00:00 GMT+0800 (CST),

Mon Jul 01 2013 00:00:00 GMT+0800 (CST),

Wed Jan 01 2014 00:00:00 GMT+0800 (CST)]

通过测试，ticks()方法返回的输入域内的时间数组，这样就可以把这时间数组当成数据源来引用。你可自定义的步长，如：

x(d3.time.hour,4)，就是以4小时为步长，均分输入域domain内的时间域

x(d3.time.minute,30)，就是以30分钟为步长，均分输入域domain内的时间域

注意的是，scale.ticks()和axis.ticks()的用法是有区别的

scale.ticks()的用法上面已经提过了，主要是用于计算均分输入域domain，然后返回一个数组；

axis.ticks()的用法是坐标轴的显示，但这里设置的axis.ticks()不会影响scale.ticks()的计算结果；

还是以上面定义的x time scale为例：

varxAxis = d3.svg.axis().scale(x).orient('bottom')，未设置坐标的ticks()。

则默认的输出坐标图是：Macintosh HD:Users:JiGuoliang:Desktop:屏幕快照 2014-03-31 下午2.38.05.png

x.ticks().length==13;

varxAxis = d3.svg.axis().scale(x).orient('bottom').ticks(d3.time.month,4) //以4个月为步长

Macintosh HD:Users:JiGuoliang:Desktop:屏幕快照 2014-03-31 下午2.36.58.png

x.ticks().length==13,所以即使设置了坐标轴的ticks()，也不影响比例尺的ticks()的计算结果

[#](https://github.com/mbostock/d3/wiki/Time-Scales#wiki-tickFormat) scale.**tickFormat**(*count*)

Returns a [time format](https://github.com/mbostock/d3/wiki/Time-Formatting) function suitable for displaying a tick value. The specified count should have the same value as the count that is used to generate the tick values. You don't have to use the scale's built-in tick format, but it automatically computes the appropriate display based on the input date.

返回一个可以显示tick值的[time format](https://github.com/mbostock/d3/wiki/Time-Formatting) 的函数。参数count应该和用于生成tick数值的count一致。你不用必须使用scale内建的tick format，但它会根据输入域的时间自动计算相应的结果。

The following time formats are considered:

比如以下时间格式：

* %Y - for year boundaries, such as "2011".
* %B - for month boundaries, such as "February".
* %b %d - for week boundaries, such as "Feb 06".
* %a %d - for day boundaries, such as "Mon 07".
* %I %p - for hour boundaries, such as "01 AM".
* %I:%M - for minute boundaries, such as "01:23".
* :%S - for second boundaries, such as ":45".
* .%L - milliseconds for all other times, such as ".012".

By using multi-scale time formats, the default tick format provides both local and global context for each time interval. For example, by showing the sequence [11 PM, Mon 07, 01 AM], the tick formatter reveals information about hours, dates, and day simultaneously—rather than just the hours. If you'd prefer single-scale time formatting, you can always use your own [d3.time.format](https://github.com/mbostock/d3/wiki/Time-Formatting). You can also roll your own [custom multi-scale time format](http://bl.ocks.org/mbostock/4149176).

使用多重time format，默认的tick format为每个time interval提供局部和全局环境。例如，为显示[11 PM, Mon 07, 01 AM]，tick格式器可以同时展现小时和日期信息--而非只有小时。如果你喜欢用单一的time 格式器，你可以一直使用你自己定义的 [d3.time.format](https://github.com/mbostock/d3/wiki/Time-Formatting)。

[#](https://github.com/mbostock/d3/wiki/Time-Scales" \l "wiki-copy) scale.**copy**()

Returns an exact copy of this time scale. Changes to this scale will not affect the returned scale, and vice versa.

返回当前time scale的完整拷贝。所有对当前time scale的改变都不会改变返回的拷贝，反之亦然。

By 季国亮