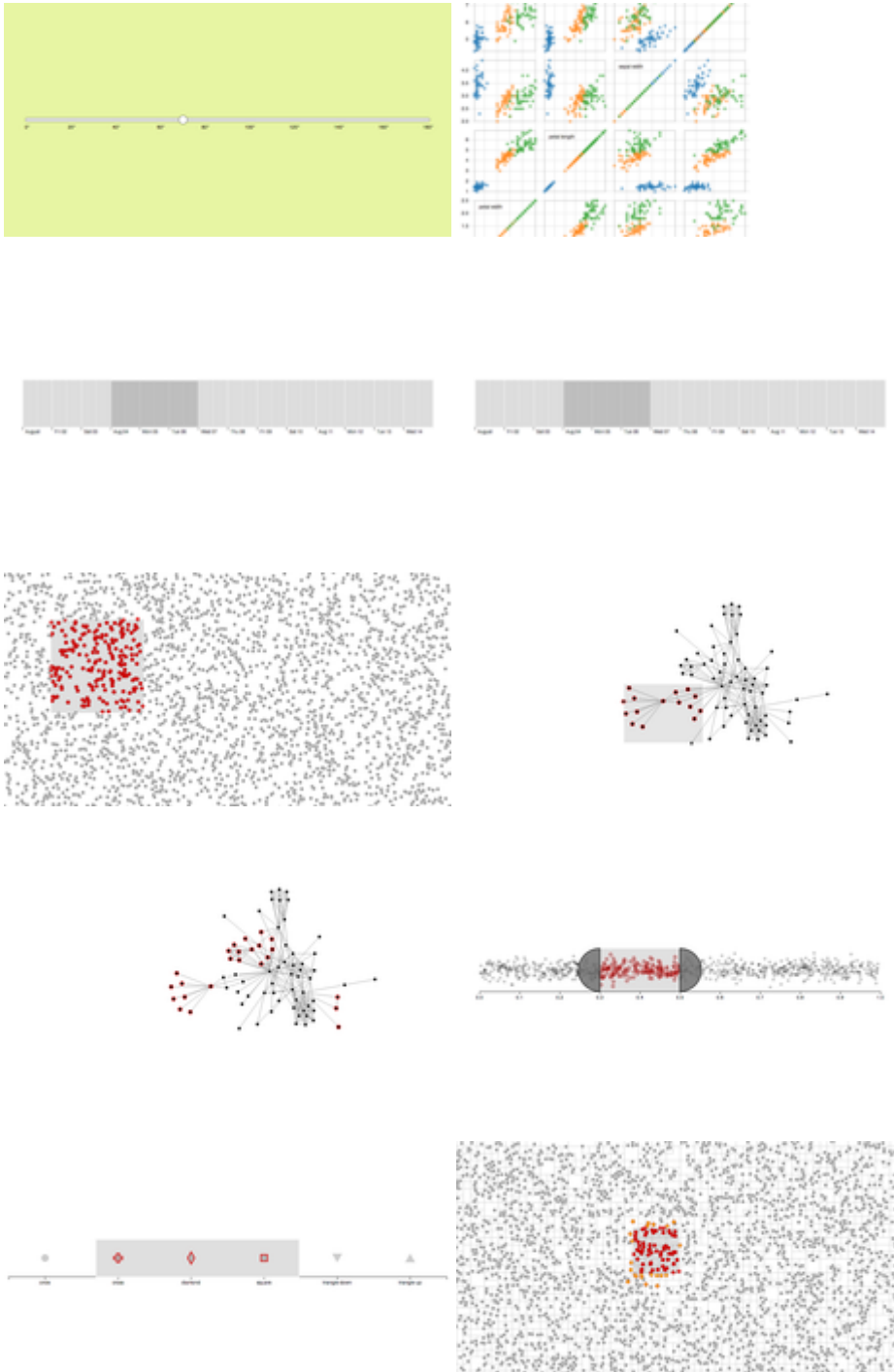


- 本文档是D3官方文档中文翻译，并保持与[最新版](#)同步。
- 如发现翻译不当或有其他问题可以通过以下方式联系译者:
- 邮箱: zhang_tianxu@sina.com
- QQ群: [D3数据可视化](#)205076374, [大数据可视化](#)436442115

刷子



[# d3.svg.brush\(\)](#)

构造一个新的刷子，使用默认的 x 和 y 比例尺，和空的范围(extent)。

[# brush\(selection\)](#)

绘制或重绘当前brush拖选到指定的选择元素`selection`；brush可以同时绘制多个元素，值得注意的是，这些选择刷会共用相同的背景范围；通常一个选择刷一个时间只能绘制一个元素；`selection` 参数可以是一个过渡变换([transition](#))，在这种情况下，选择刷将执行自动变换([automatic transition](#))；可以使用 [brush.event](#) 来触发选择刷事件在动画刷的变换过程中。

[# brush.x\(\[scale\]\)](#)

获取或设置选择刷相关联的 x 比例尺；如果指定 `scale`，则设置 x 比例尺为指定的scale并返回brush，如果未指定 `scale`，则返回当前的 x 比例尺，默认为null；变换通常可指定为[数值比例尺](#)，在这种情况下，范围 [extent](#) 处于比例尺域 [domain](#) 的数据空间；然而，它也可以被定义为[序数比例尺](#)代替，这时，范围extent来自于比例尺的[range extent](#)的像素空间。

[# brush.y\(\[scale\]\)](#)

获取或设置选择刷相关联的 y 比例尺；如果指定`scale`，则设置 y 比例尺为指定的scale并返回brush，如果未指定`scale`，则返回当前的 y 比例尺，默认为 `null`；比例尺通常可指定为[数值比例尺](#)，在这种情况下，[extent](#) 是来自于比例尺域 [domain](#) 的数据空间；然而，它也可以被定义为[序数比例尺](#)代替，这时，范围extent来自于变换的[range extent](#)的像素区间。

[# brush.extent\(\[values\]\)](#)

获取或设置当前选择刷的范围，如果指定`values`，则设置范围为指定的值并返回当前brush；如果未指定`values`，则返回当前的范围；范围的定义依赖于关联的比例尺；如果 x 和 y 比例尺都可用，范围是一个二维的数组：`[[x0, y0], [x1, y1]]`，`x0`和`y0`是范围的最低端，`x1`和`y1`是范围的最顶端；如果只有 x 比例尺可用，范围被定义为一维数组：`[x0, x1]`，同样地，如果只有 y 变换可用，范围被定义为：`[y0, y1]`；如果没有变换可用，范围为 `null`。

当范围被设定为指定的值`values`，所得到的范围会被正确的保存起来；然而，一旦选择刷被用户移动（鼠标按下并拖动），这时，范围必须要调用 [scale.invert](#)来重新计算；注意，在这种情况下，值可能由于像素的精度有限而略有偏差。

注意，这并不会自动重绘选择刷或触发任何的监听事件；想要重绘选择刷，可以在选择器或过渡上调用 [brush](#)，想要触发事件，使用 [brush.event](#)。

[# brush.clamp\(\[clamp\]\)](#)

设置或获取当前的夹选行为，如果指定`clamp`，则设置夹选行为为指定值并返回brush，如果未指定`clamp`，则返回当前的行为；夹选行为的定义依赖于关联的比例尺；如果 x 和 y 比例尺都可用，夹选行为是一个数组`[x,y]`， x 和 y 是布尔类型，用来确定是否二维范围内每个维度应该被夹选到各自的 x 和 y 比例尺；如果只有 x 或 y 比例尺可用，夹选行为是一个布尔类型，用来指定是否一维范围该被夹选到比例尺，如果变换都不可用，则夹选行为是 `null`。

[# brush.clear\(\)](#)

清空范围，使得brush的范围为 [empty](#)。

[# brush.empty\(\)](#)

当且仅当选择刷的范围为空时，返回 `true`；当brush被创建时，被初始化为空；当点击背景而不移动时，或者范围被[清除](#)，选择刷会变为空的；如果选择刷有零宽度或零高度，它将被视为空；当选择刷为空，则它的范围即视为未定义。

[# brush.on\(type\[, listener\]\)](#)

设置或获取指定类型 *type* 的监听器 *listener*；选择刷支持三种类型事件：

- *brushstart* - 鼠标按下时，即 *mousedown*；
- *brush* - 鼠标移动时，如果范围在改变，即 *mousemove*；
- *brushend* - 鼠标弹起/松开时，即 *mouseup*；

需要注意，当鼠标在背景上点击时也会触发“brush”事件，因为选择刷范围会立刻被清除来开始一段新的范围。

<#> `brush.event(selection)`

如果 *selection* 是选择器，立刻触发brush行为到注册的监听器，即三个事件序列：*brushstart*, *brush* 和 *brushend*；这是非常有用的，在设置完 [brush extent](#) 后来触发相应的事件；如果 *selection* 是一个过渡，注册合适的补间动画，这样在过渡的过程中来触发事件：当过渡开始于初始设置范围时触发*brushstart*，过渡进行期间每刻都会触发*brush*，过渡结束时触发*brushend*；需要注意，当用户开始刷时，即使过渡没结束也会被立刻终止([interrupted](#))。

- 魏飞译 2014-07-25 19:25 咕噜校对 2014-11-29 20:06:46