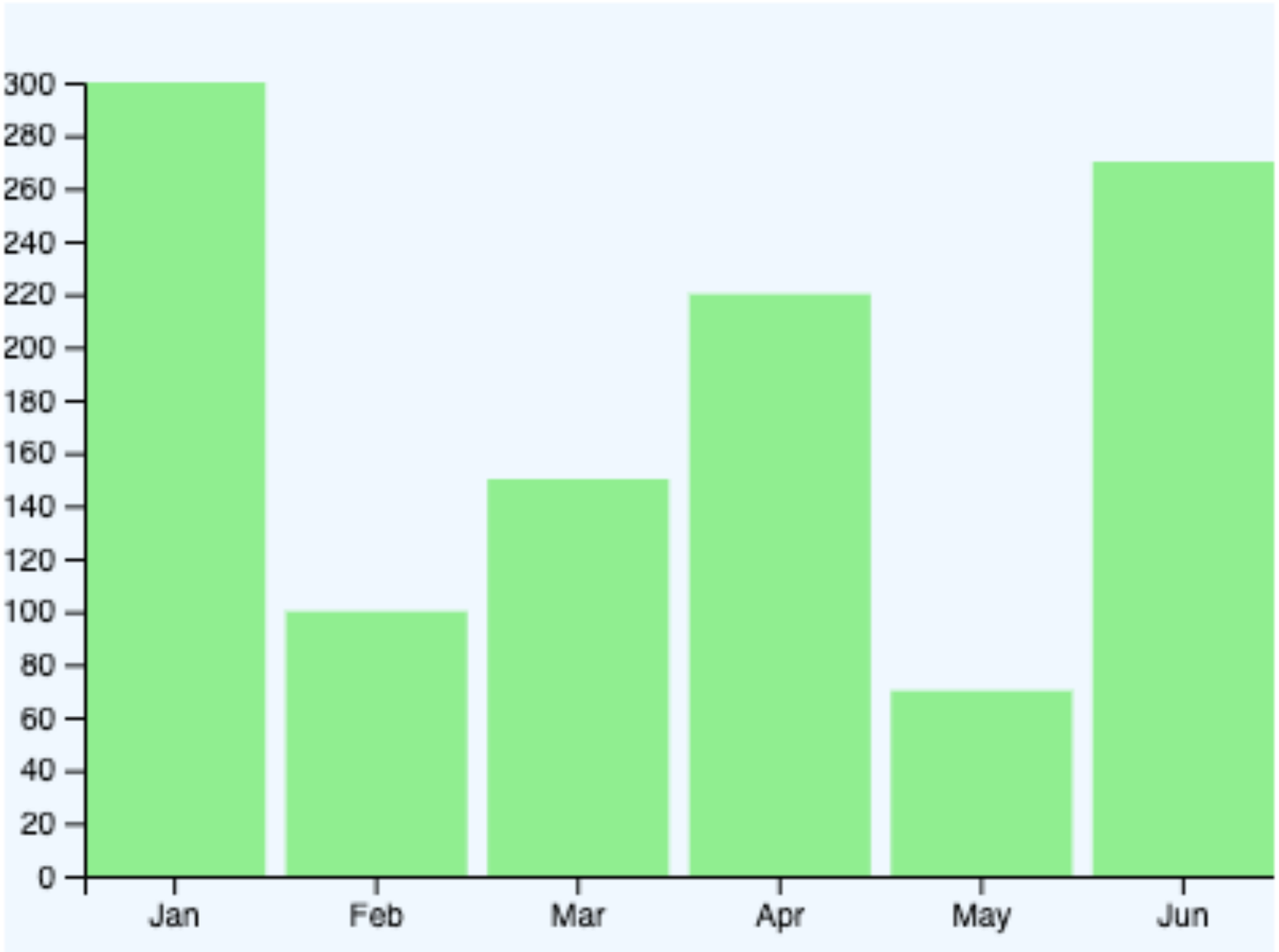
Axes

Why do we need axes?



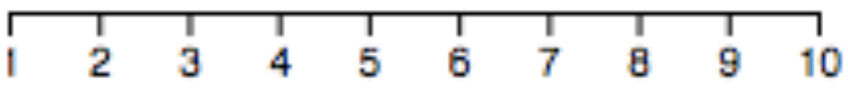
Start with:

scale

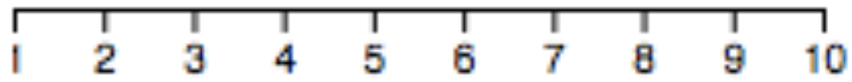
```
const xScale = d3.scaleLinear()
    .domain([1,10])
    .range([0,200]);
```

axis component

d3.axisBottom()



Create an axis generator

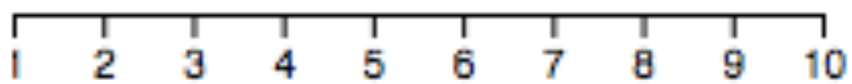



```
const xAxis = d3.axisBottom()
    .scale(xScale);
```

think:

```
const xAxis = d3.axisBottom(xScale);
```

Call it to create SVG elements



When called on a selection, the axis generator creates axis SVG elements

```
d3.select("svg").append("g")
    .call(xAxis);
```

think:

```
xAxis(d3.select("svg").append("g"));
```

Generated SVG axis elements

```
<g fill="none" font-size="10" font-family="sans-serif" text-anchor="middle">

  <path class="domain" stroke="#000" d="M0.5,6V0.5H200.5V6"></path>

  <g class="tick" opacity="1" transform="translate(0.5,0)">
    <line stroke="#000" y2="6"></line>
    <text fill="#000" y="9" dy="0.71em">1</text>
  </g>

  <g class="tick" opacity="1" transform="translate(22.72222222222222,0)">
    <line stroke="#000" y2="6"></line>
    <text fill="#000" y="9" dy="0.71em">2</text>
  </g>

    (8 more tick mark / tick label groups)

</g>
```

Generated SVG axis elements

axis component

axis generator

d3.axisBottom(*scale*) --> xAxis

xAxis([*selection*]) --> SVG elements

call on a selection

Possible, but not advisable:

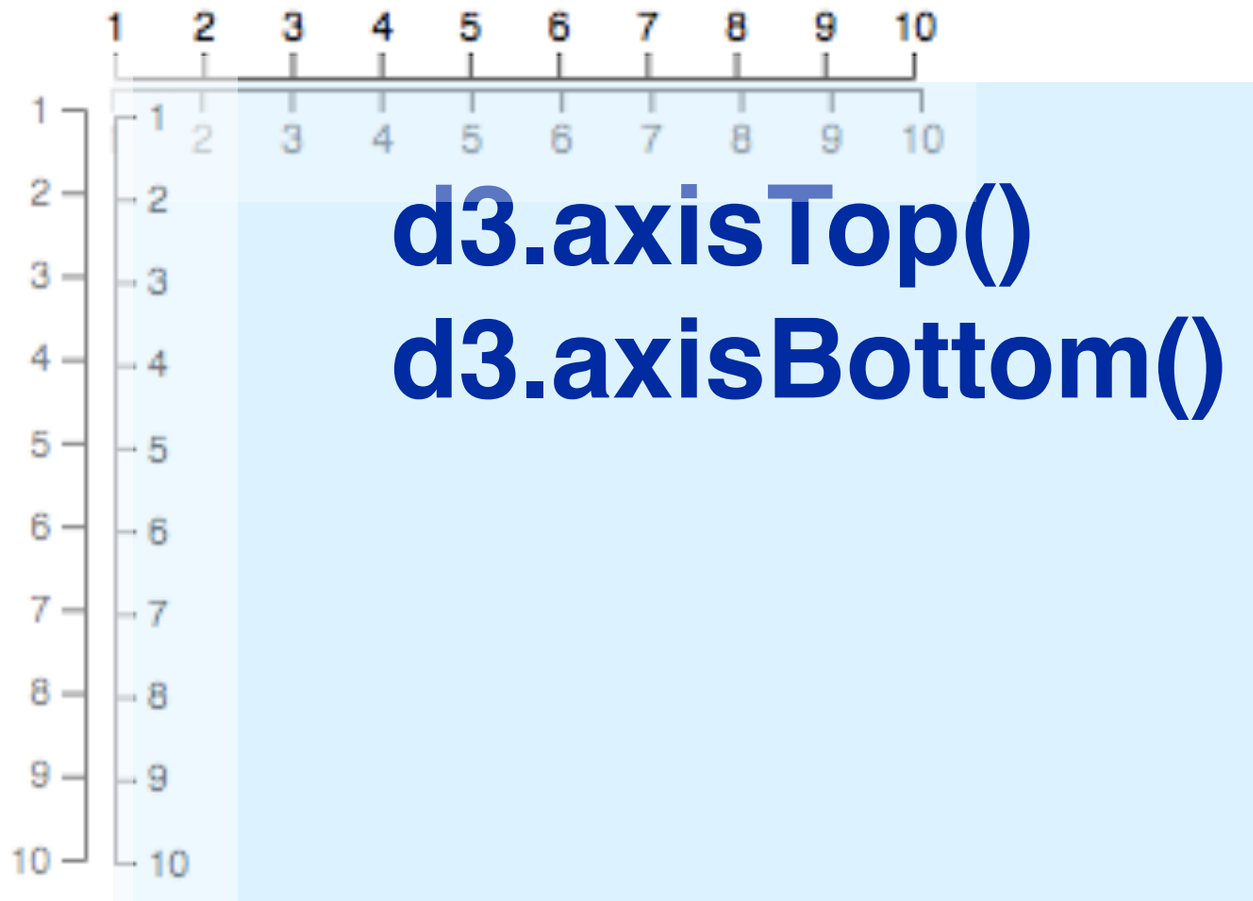
```
d3.axisBottom(d3.scaleLinear()
    .domain([1,10])
        .range([0,200]))(d3.select("svg")
                .append("g"));
```

Axis Components

control *orientation* not *location* on the svg

all axes are rendered at the origin

d3.axisLeft()
d3.axisRight()



d3.axisTop()
d3.axisBottom()

Translate axes to position them

```
const yAxis = d3.axisLeft()
    .scale(yScale);

svg.append("g")
  .attr("class", "yAxis")
  .attr("transform",
     `translate(${margin.left},
                ${margin.top})`)
  .call(yAxis);
```