

If we try to insert Pangea as well then our scale doesn't have a free value that can use that range.

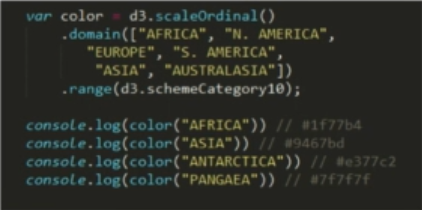
So it looks back and assigns that the value index 0 which is red.

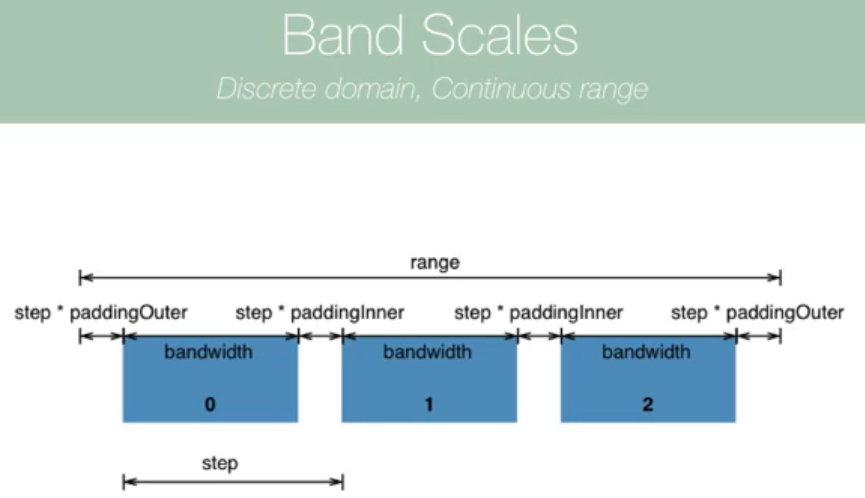
Usage:

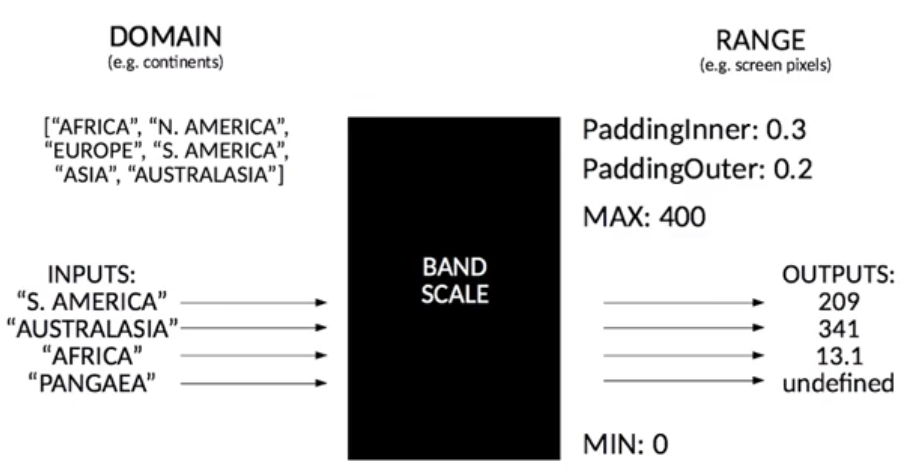


Use predefined color group:

<https://github.com/d3/d3-scale-chromatic/blob/v1.5.0/README.md#schemeCategory10>









const arr = [];

        this.building.forEach((elem) => {

            arr.push(elem.name);

        });

        let width = $D

            .**scaleBand**()

            .domain(arr)

            .range([0, 400])

            .paddingOuter(0.2)

            .paddingInner(0.2);

        let bars = svg.selectAll("rect").data(this.building);

        bars

            .enter()

            .append("rect")

            .attr("x", (d, i) => {

**// get x position by passing a key**

                return **width**(d.name);

            })

            .attr("y", 50)

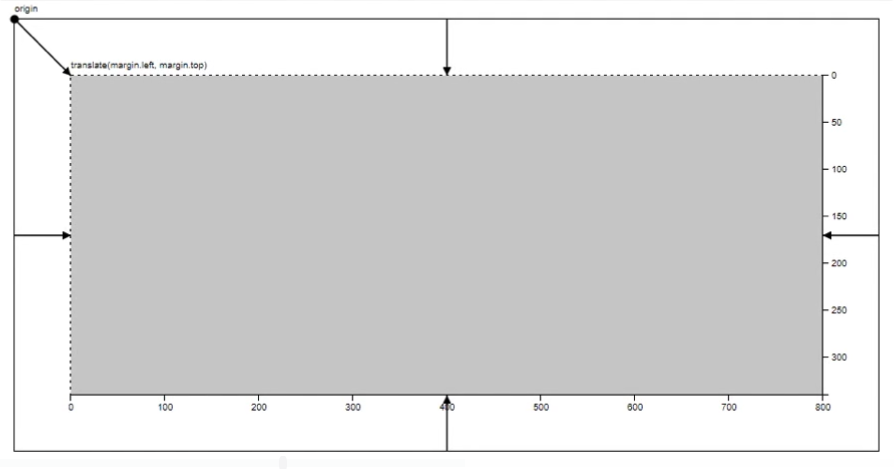
            .attr("fill", "blue")

**// scaleBand allocate width of each bar equally**

            .attr("width", width.**bandwidth**)

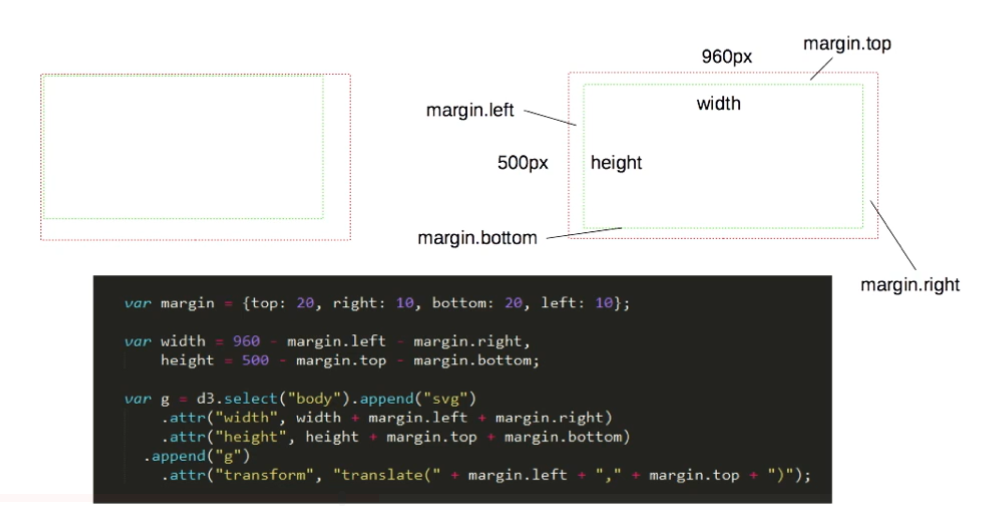


Margins and groups:

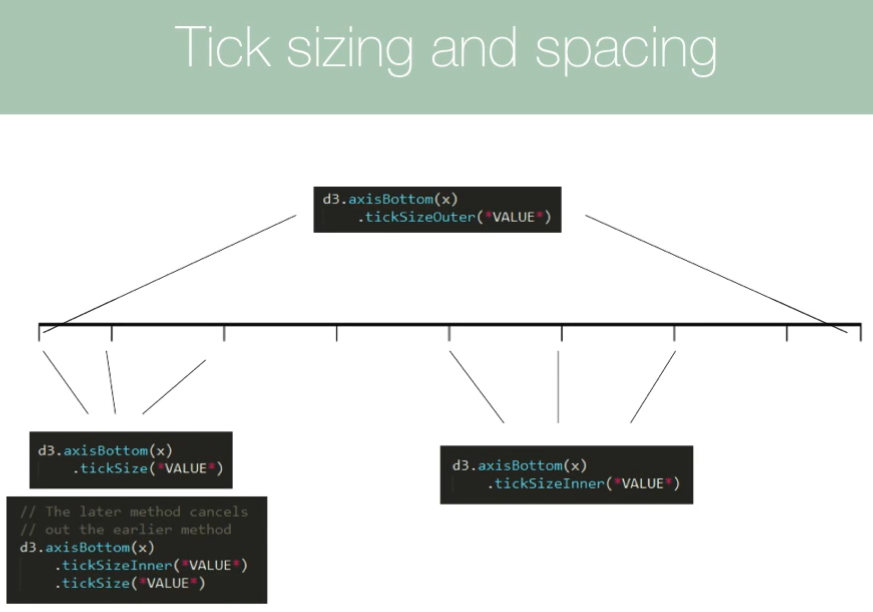
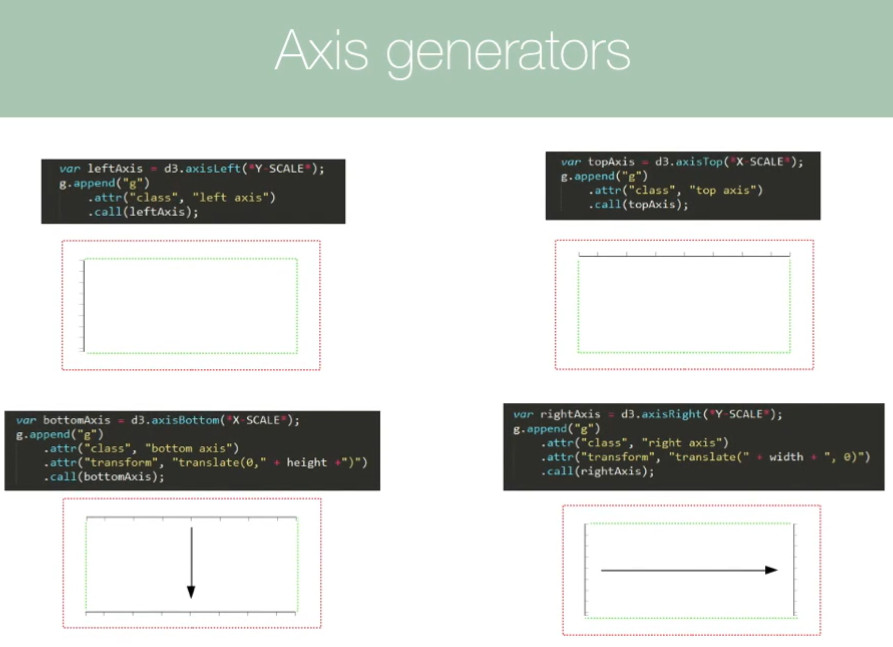


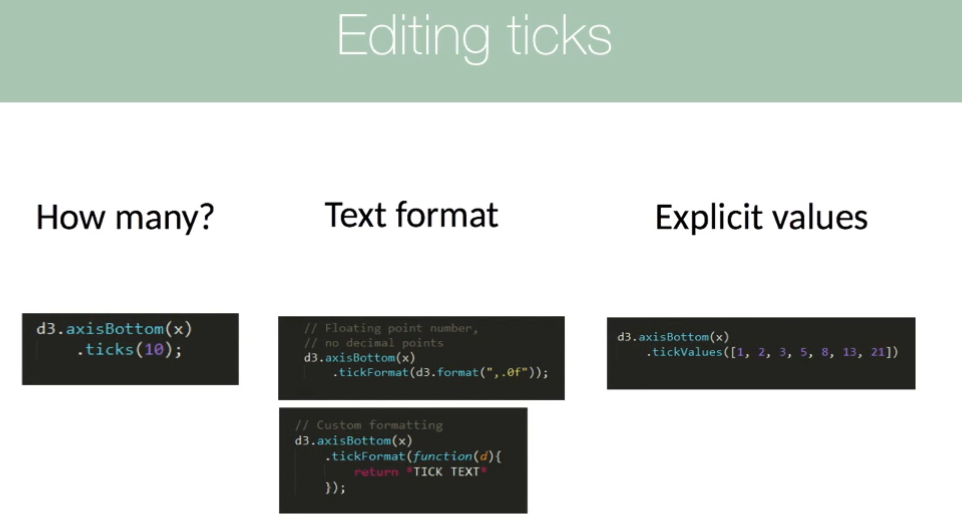


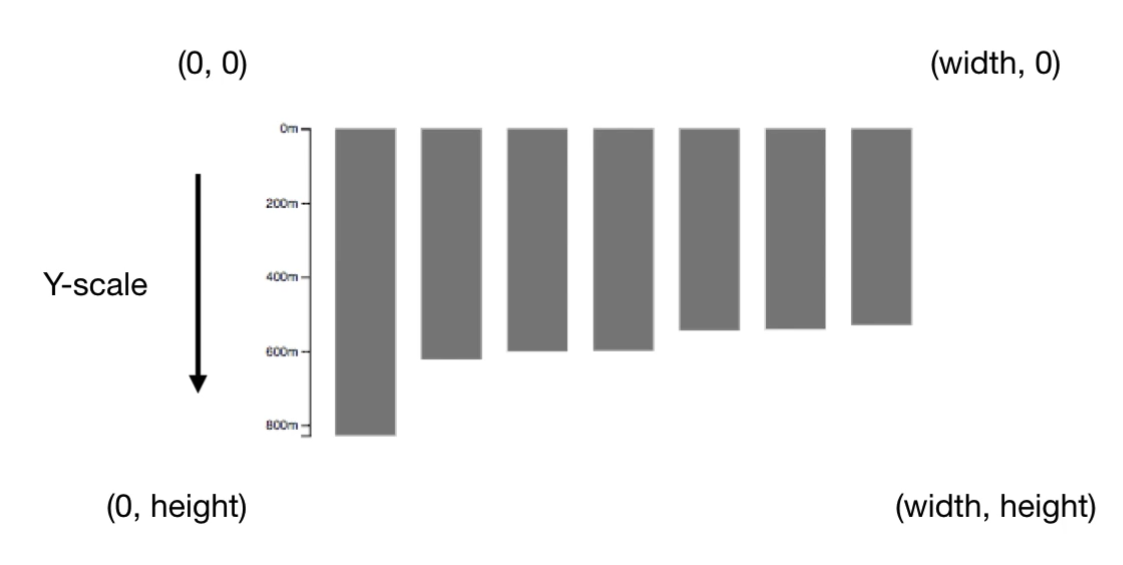




Axes and Labels:







let height = $D

            .scaleLinear()

            .domain([

                0,

                $D.max(this.building, (bd) => {

                    return bd.height;

                })

            ])

**.range([0, 800]);**

bars

            .enter()

            .append("rect")

            .attr("x", (d, i) => {

                return width(d.name);

            })

**.attr("y", 0)**

            .attr("fill", "blue")

            .attr("width", width.bandwidth)

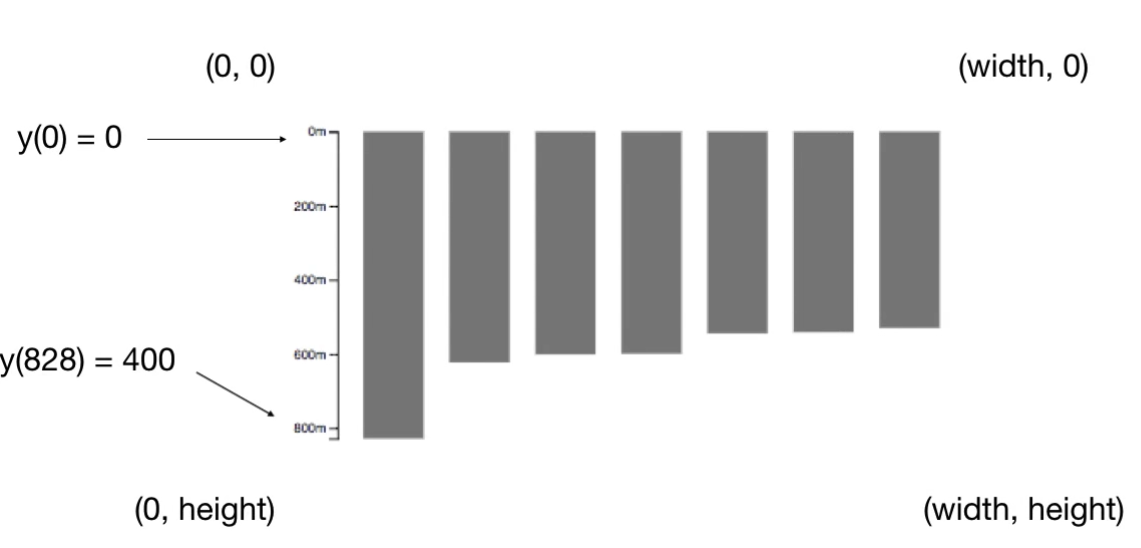
**.attr("height", (d, i) => {**

**// linear map a new height**

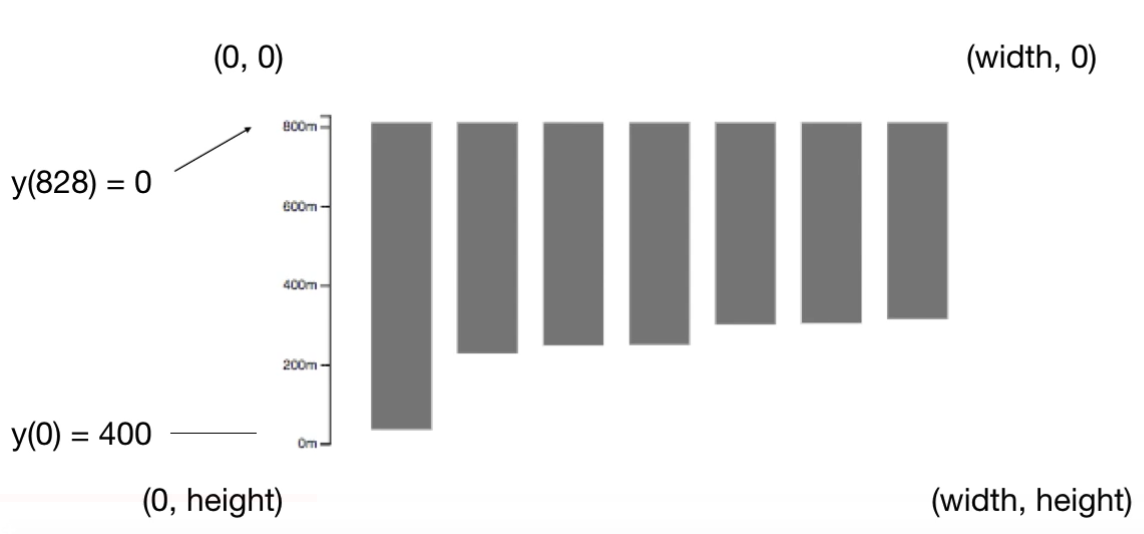
**return height(d.height);**

**})**

Current:



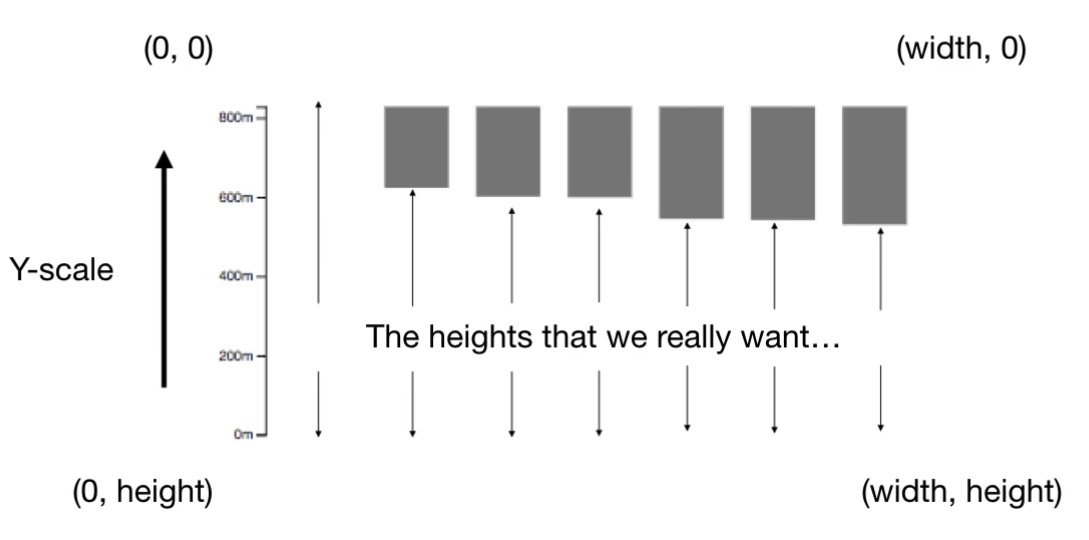
Logically right:



**// reverse mapping**

**.range([800, 0]);**

****

****

**return height - y(height);**

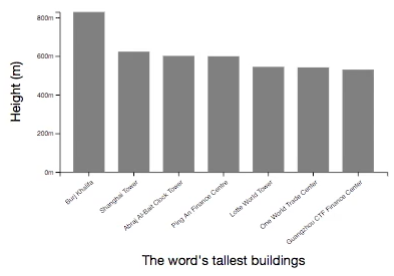
****

**.attr("y", 0)**

**therefore these bars hand on top**

**.attr("y", (d)=>{return y(d.height)})**

**y should be y(d.height)**

****