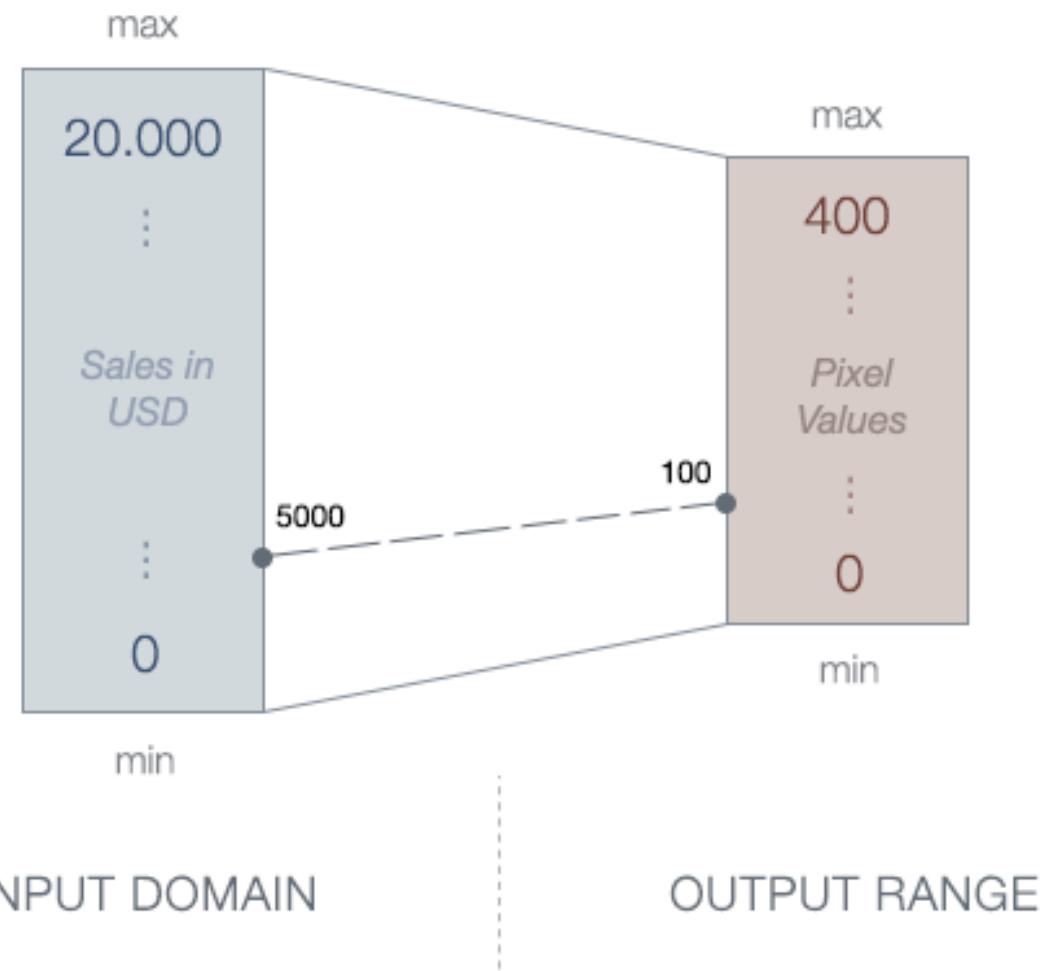


Lab 4 - Scales

Until now, when creating a D3 visualization, we used only x and y values that corresponded directly to pixel measurements on the screen, within a pre-defined SVG drawing area. That is not very flexible and only feasible for static data. What if our data attributes are suddenly doubled? We can not increase the size of the chart every time a value increases. At some point, the user might have to scroll through a simple bar chart to get all the information.



Linear Scale

```
// Creating a scale function
var iceCreamScale = d3.scale.linear()
  .domain([0, 20000])
  .range([0, 400]);

// Call the function and pass an input value
iceCreamScale(5000);    // Returns: 100
```

```
// Create an ordinal scale function
var xScale = d3.scale.ordinal()
  .domain(["May", "June", "July", "August"])
  .rangeBands([0, 400]); // D3 fits n (=4) bands within this interval
```

Min, Max, and Extent

```
var quarterlyReport = [
  { month: "May", sales: 6900 },
  { month: "June", sales: 14240 },
  { month: "July", sales: 25000 },
  { month: "August", sales: 17500 }
];

// Returns the maximum value in a given array (= 25000)
var max = d3.max(quarterlyReport, function(d) {
  return d.sales;
});

// Returns the minimum value in a given array (= 6900)
var min = d3.min(quarterlyReport, function(d) {
  return d.sales;
});

// Returns the min. and max. value in a given array (= [6900,25000])
var extent = d3.extent(quarterlyReport, function(d) {
  return d.sales;
});
```

Colors

- `d3.scale.category10()` - ordinal scale with a range of 10 categorical colors
- `d3.scale.category20()` - ordinal scale with a range of 20 categorical colors

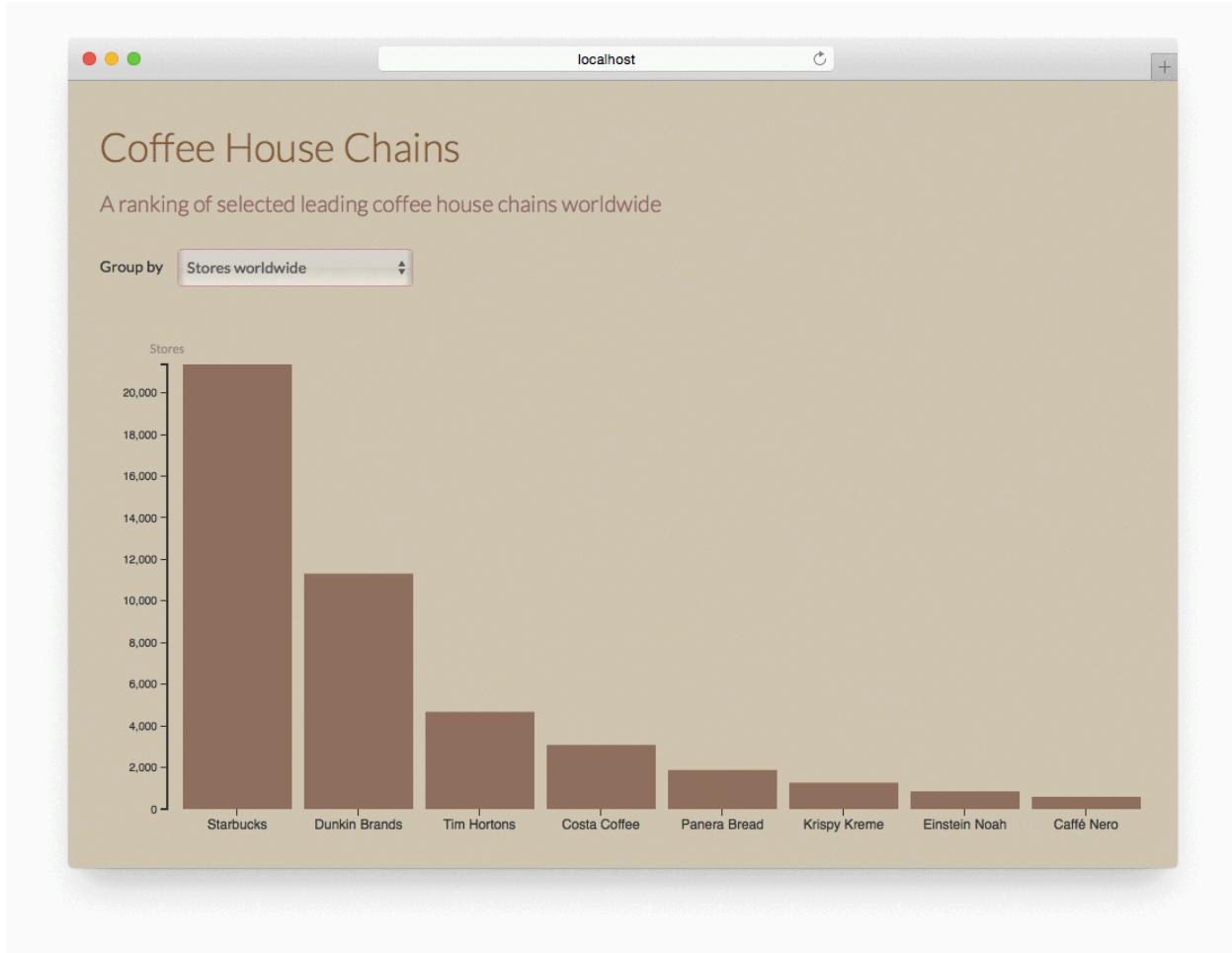
```
// Construct a new ordinal scale with a range of ten categorical colors
var colorPalette = d3.scale.category10();

// Print color range
console.log(colorPalette.range());
// ["#1f77b4", "#ff7f0e", "#2ca02c", "#d62728", "#9467bd", "#8c564b", "#e377c2", "#7f7f7f", "#bcbd22", "#1f77b4"]

// Specify domain (optional)
colorPalette.domain(["Vanilla", "Cookies", "Chocolate", "Pistachio"]);

// Use color palette
colorPalette("Chocolate") // Returns: #2ca02c
```

Example:



// 1a START

```
var y = d3.scale.linear().range([height, 0]);
```

// 1a END

//1b START

```
var rankingType = selectRankingType.property("value");
```

//1b END

```
//1c: START  
y.domain([0, d3.max(data, function(d) { return d[rankingType]; })]);
```

```
//1c END
```

```
//1d START  
var bars = svg.selectAll(".bar").data(data);
```

```
// Enter  
bars.enter().append("rect").attr("class", "bar");  
  
// Update  
bars  
.attr("x", function(d) { return x(d.company); })  
.attr("y", function(d) { return y(d[rankingType]); })  
.attr("width", x.rangeBand())  
.attr("height", function(d) { return height - y(d[rankingType]); })
```

```
// Exit  
bars.exit().remove();
```

```
//1d END
```

```
bars  
.transition()  
.duration(1000)
```