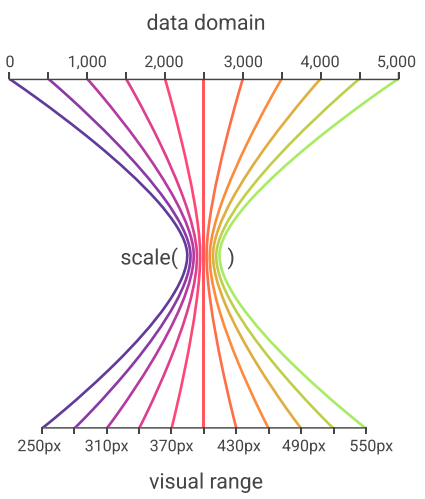
Scales

Scales project your data from an abstract data domain to a visual range

Observable Plot version 0.4.0

Working with scales

How scales map values:



Configure the scale for each channel:

```
Plot.plot({
  // Configure the scale for the x channel
  x: {
    type: "log",        // scale type
    ticks: 5,           // # of ticks
    tickFormat: ".2s",  // tick format
    grid: true,      // show grid lines
    axis: "top"      // show above chart
  }
})
```

Scale options:

```
axis: "top"
domain: [0, .5]
grid: true
inset: 20
line: true
percent: true
reverse: true
```

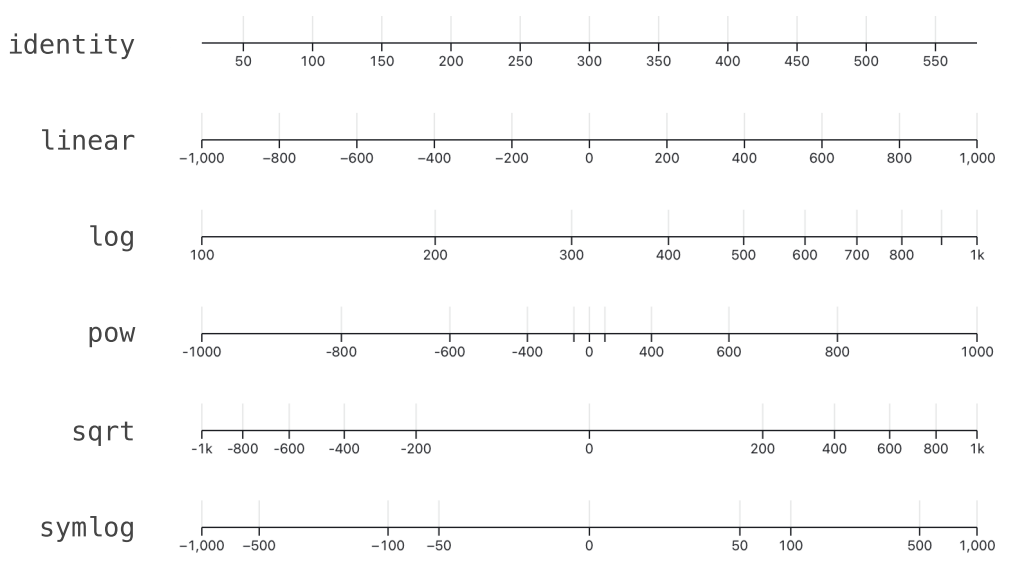
Label and tick options:

```
label: "My label"
labelAnchor: "left"
labelOffset: 10
nice: true
tickPadding: 20
tickRotate: -90
tickSize: 23
ticks: 1
```

Quantitative

Display continuous data by setting one of these types:

```
Plot.plot({ x: { type: "identity" } })
```



Specify a `tickFormat: "[symbol][comma][precision][type]"`

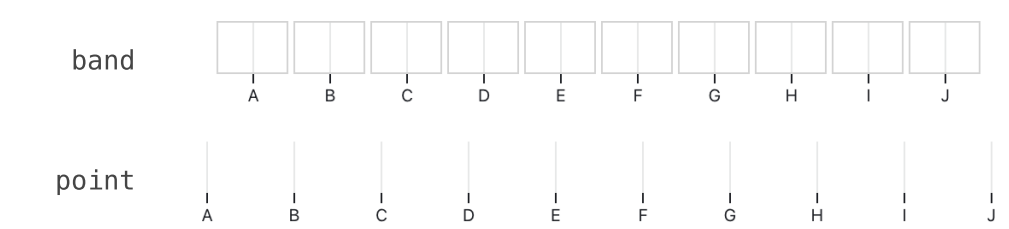
```
Plot.plot({ x: { tickFormat: ".2s" } })
```

Syntax	Description	format(0.00013)	format(543005)
$	Currency symbol	$0.00013	$543005
,	Comma separated	0.00013	543,005
.2	Precision of 2 digits	0.00013	5.4e+5
.5	Precision of 5 digits	0.00013	5.4301e+5
s	International System of Units (SI).	130.000µ	543.005k
e	Exponent notation	1.300000e-4	5.430050e+5
f	Fixed point notation	0.000130	543005.000000
p	Percentage notation	0.0130000%	54300500%
.2s	Two significant digits, shown in SI.	130µ	540k
,.1f	Comma separated, one fixed value after the decimal place	0.0	543,005.0
,.1p	Comma separated, one digit, percentage type	0.01%	50,000,000%
$,.1	Currency syntax, Comma separated, one digit, percentage type	$0.0001	$5e+5

Categorical

Display categorical data by setting one of these types:

```
Plot.plot({ x: { type: "band" } })
```



Customize your ticks using a function:

```
Plot.plot({ x: { tickFormat: (d) => `Group ${d}` } })
```

```
tickFormat: d => `Group ${d}`
```


```
tickFormat: d => d.toLowerCase()
```

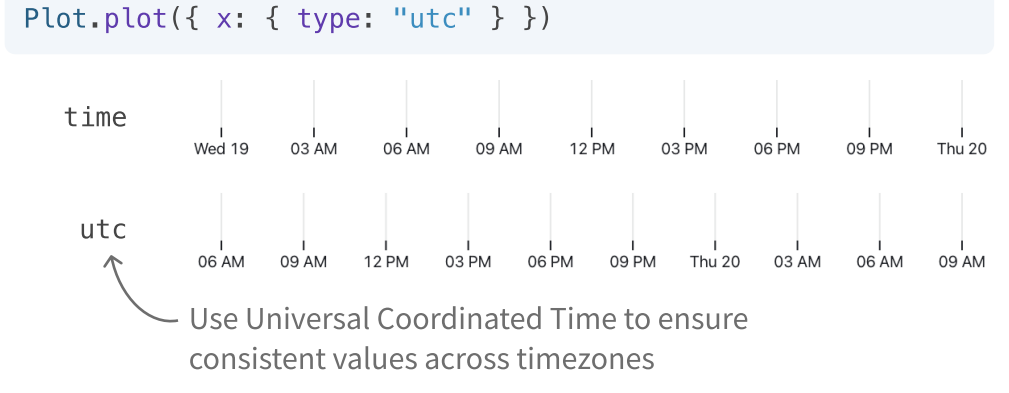


```
tickFormat: (d, i) => `${d}: ${i}`
```


Date

Display temporal data by setting one of these types:

```
Plot.plot({ x: { type: "utc" } })
```



e.g. Saturday January 01, 2022

Compose a time formatter using this syntax:

```
Plot.plot({ x: { tickFormat: d3.utcFormat("%A %B %d, %Y") } })
```

Year	Month	Day	Hour	Minute	Second	Misc
%Y 2022	%B January	%A Saturday	%I 04	%M 00	%S 00	%p AM
%y 22	%b Jan	%a Sat	%H 16			
	%m 01	%d 01				
		%e 1				