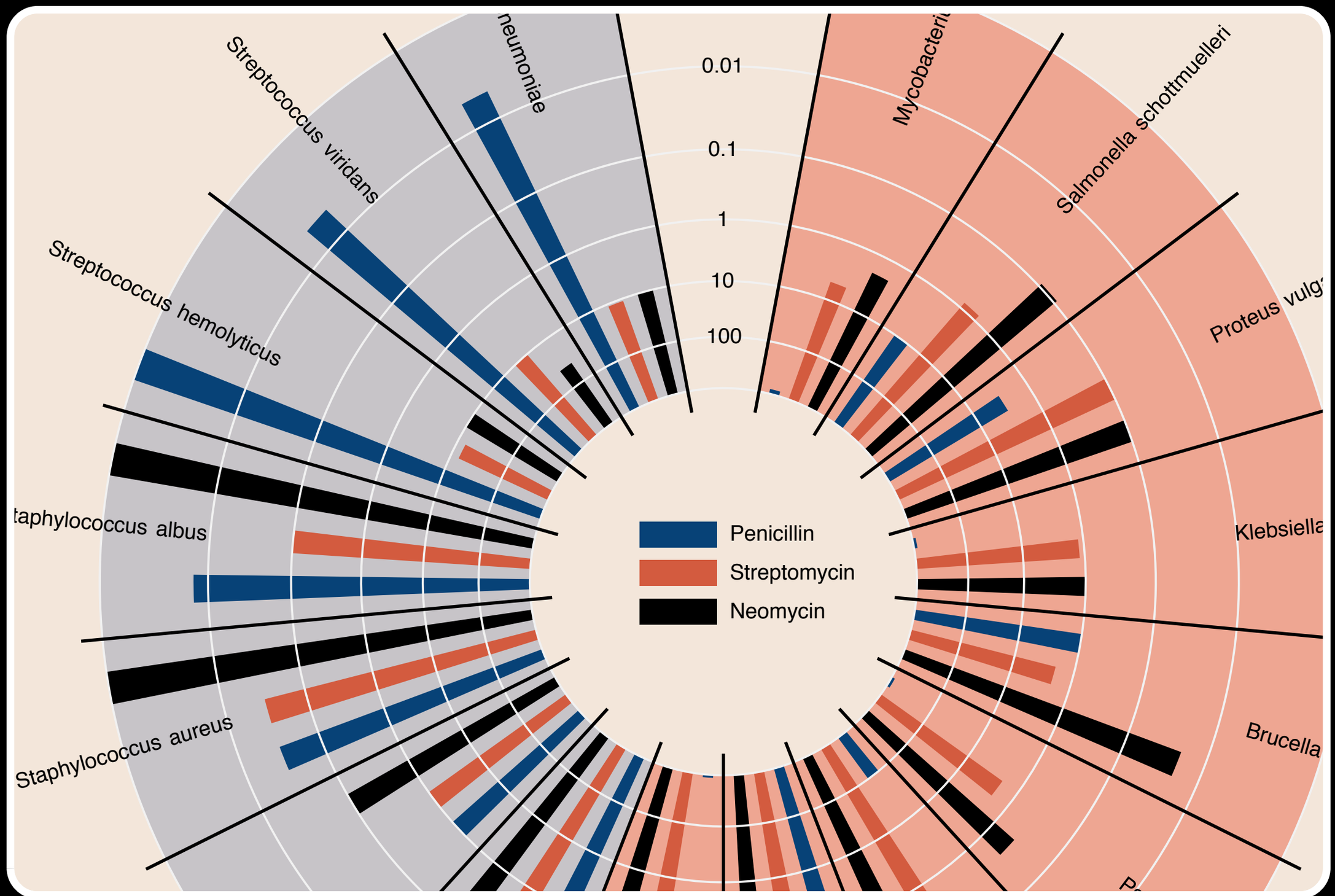


Burtin's Antibiotics, 1951

Protovis Tutorial

CS448b / October 9, 2009 / Mike Bostock

1. Fundamentals

Language design and features

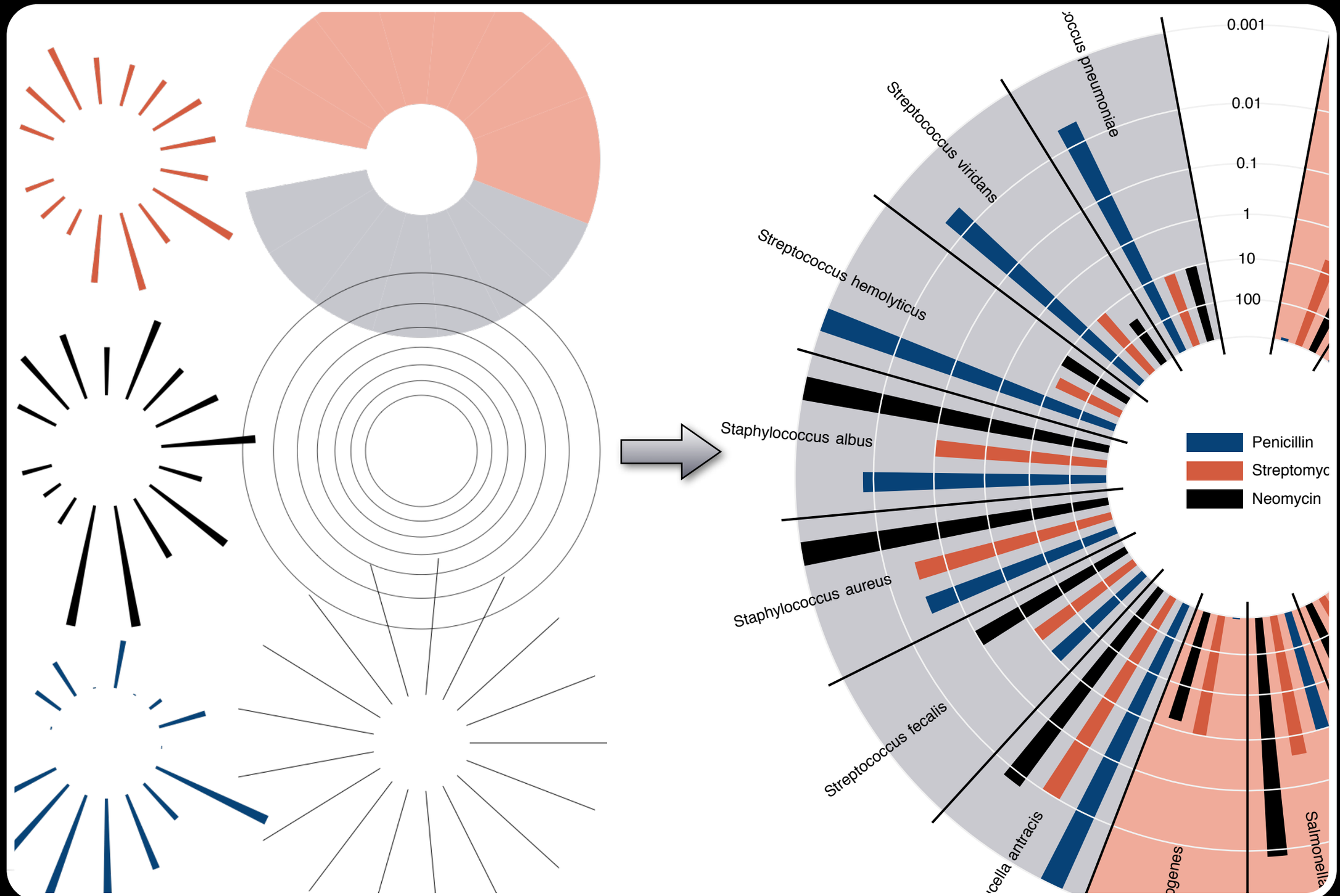
2. Demo

Examples of what you can do with Protovis

3. Sandbox

Try it out!

Outline



Mark decomposition

A graphical approach to visualization

mark

data	λ
visible	λ
left	λ
bottom	λ
height	λ
width	λ
fillStyle	λ
strokeStyle	λ
lineWidth	λ
...	...

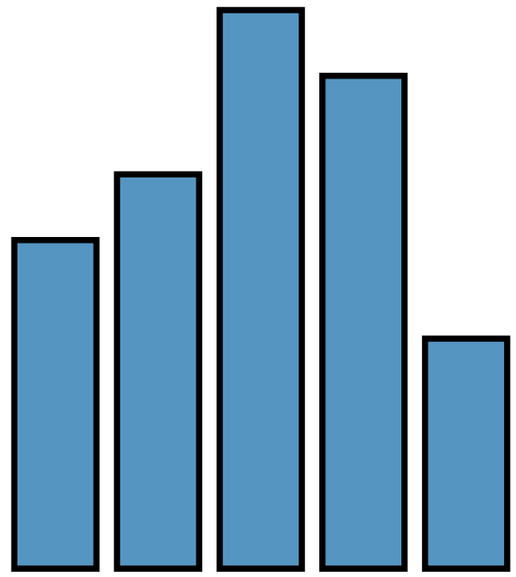
$$f : \mathbb{N} \rightarrow \mathbb{R}$$

Mark specification

Data and a set of visual encodings (properties)

bar

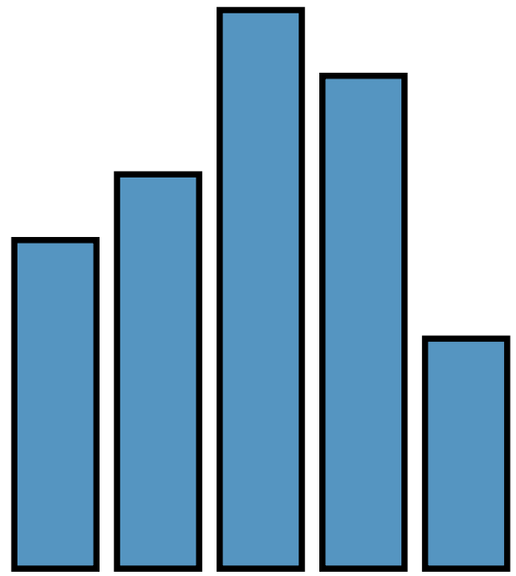
data	[1, 1.2, 1.7, 1.5, .7]
visible	true
left	λ: i * 25
bottom	0
height	λ: d * 80
width	20
fillStyle	blue
strokeStyle	black
lineWidth	1.5
...	...



Mark specification

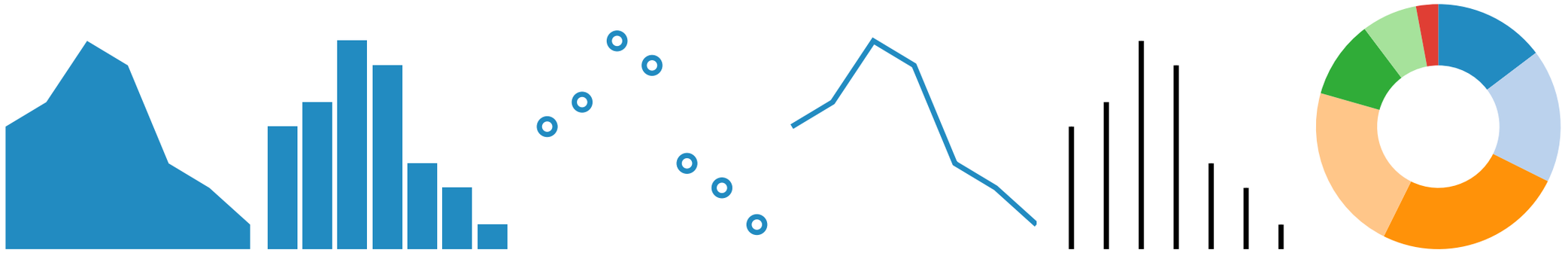
Data and a set of visual encodings (properties)

```
vis.add(pv.Bar)
  .data([1, 1.2, 1.7, 1.5, .7])
  .visible(true)
  .left(function() this.index * 25)
  .bottom(0)
  .height(function(d) d * 80)
  .width(20)
  .fillStyle(“blue”)
  .strokeStyle(“black”)
  .lineWidth(1.5);
```



Mark specification

Data and a set of visual encodings (properties)



area

data	λ
visible	λ
left	λ
top	λ
bottom	λ
right	λ
width	λ
height	λ
fillStyle	λ
strokeStyle	λ
lineWidth	λ
...	...

bar

data	λ
visible	λ
left	λ
top	λ
bottom	λ
right	λ
width	λ
height	λ
fillStyle	λ
strokeStyle	λ
lineWidth	λ
...	...

dot

data	λ
visible	λ
left	λ
top	λ
bottom	λ
right	λ
size	λ
shape	λ
fillStyle	λ
strokeStyle	λ
lineWidth	λ
...	...

line

data	λ
visible	λ
left	λ
top	λ
bottom	λ
right	λ
fillStyle	λ
strokeStyle	λ
lineWidth	λ
...	...

rule

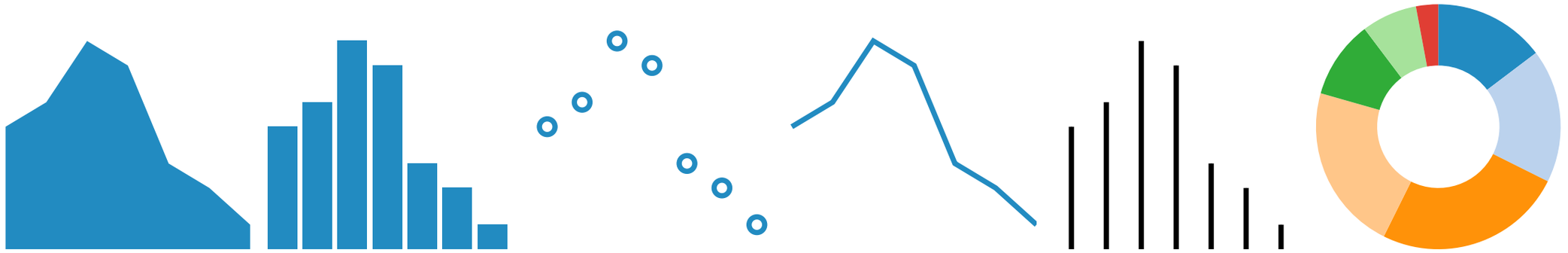
data	λ
visible	λ
left	λ
top	λ
bottom	λ
right	λ
width	λ
height	λ
strokeStyle	λ
lineWidth	λ
...	...

wedge

data	λ
visible	λ
left	λ
top	λ
bottom	λ
right	λ
innerRadius	λ
outerRadius	λ
startAngle	λ
endAngle	λ
fillStyle	λ
strokeStyle	λ
lineWidth	λ
...	...

Mark types

Not pictured: image, label, panel, anchor



area

data	λ
visible	λ
left	λ
top	λ
bottom	λ
right	λ
width	λ
height	λ
fillStyle	λ
strokeStyle	λ
lineWidth	λ
...	...

bar

data	λ
visible	λ
left	λ
top	λ
bottom	λ
right	λ
width	λ
height	λ
fillStyle	λ
strokeStyle	λ
lineWidth	λ
...	...

dot

data	λ
visible	λ
left	λ
top	λ
bottom	λ
right	λ
size	λ
shape	λ
fillStyle	λ
strokeStyle	λ
lineWidth	λ
...	...

line

data	λ
visible	λ
left	λ
top	λ
bottom	λ
right	λ
fillStyle	λ
strokeStyle	λ
lineWidth	λ
...	...

rule

data	λ
visible	λ
left	λ
top	λ
bottom	λ
right	λ
width	λ
height	λ
strokeStyle	λ
lineWidth	λ
...	...

wedge

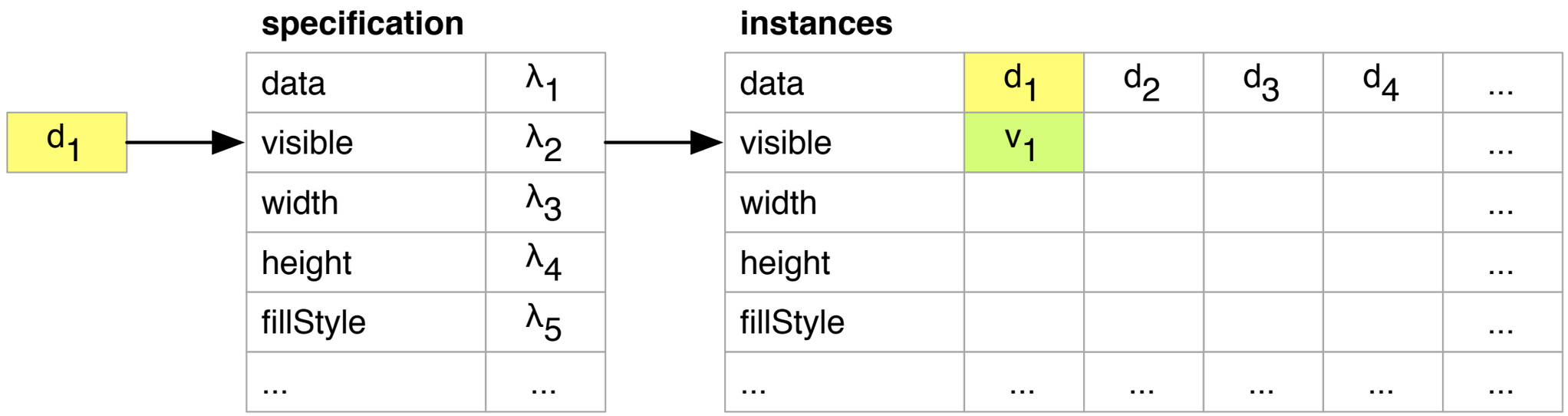
data	λ
visible	λ
left	λ
top	λ
bottom	λ
right	λ
innerRadius	λ
outerRadius	λ
startAngle	λ
endAngle	λ
fillStyle	λ
strokeStyle	λ
lineWidth	λ
...	...

Mark types

Almost all properties are the same across types

Property evaluation

Computed property values stored in scene graph



Property evaluation

Computed property values stored in scene graph

specification

data	λ_1
visible	λ_2
width	λ_3
height	λ_4
fillStyle	λ_5
...	...

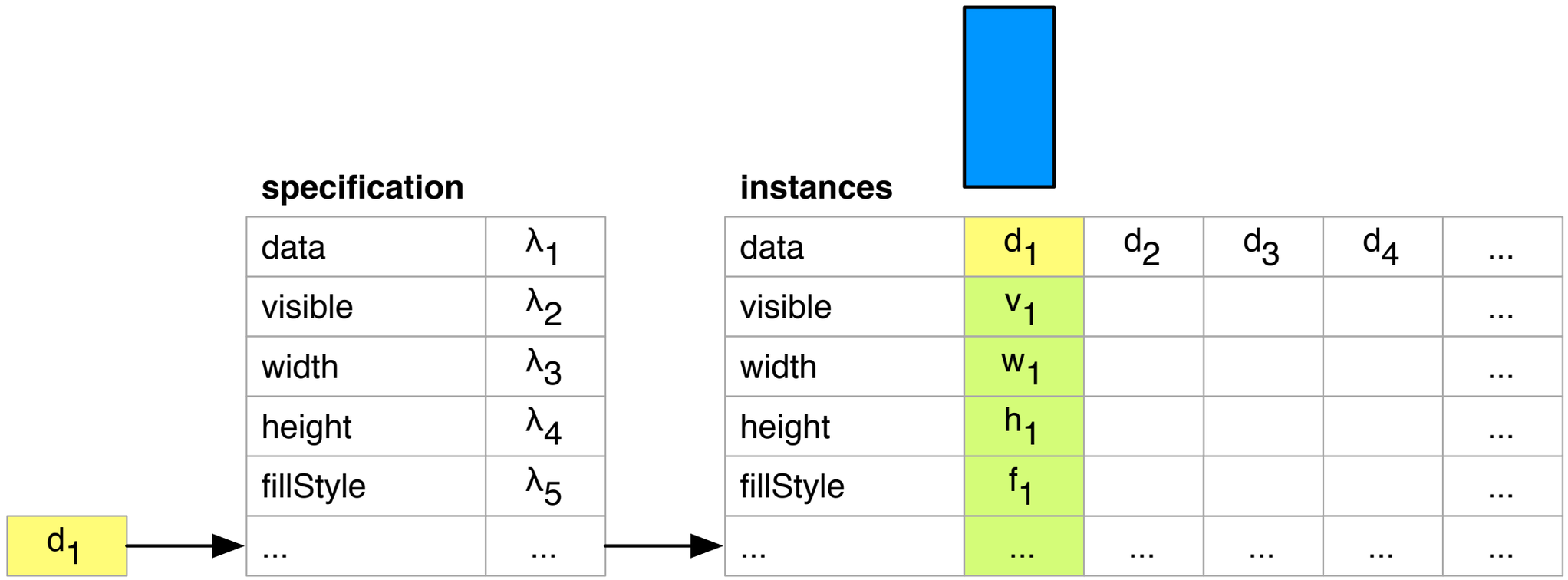
d_1 →

instances

data	d_1	d_2	d_3	d_4	...
visible	v_1				...
width	w_1				...
height					...
fillStyle					...
...	...	...	...	...	...

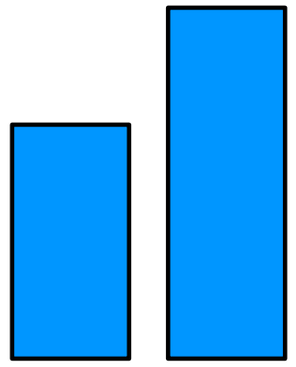
Property evaluation

Computed property values stored in scene graph



Property evaluation

Computed property values stored in scene graph



specification

data	λ_1
visible	λ_2
width	λ_3
height	λ_4
fillStyle	λ_5
...	...

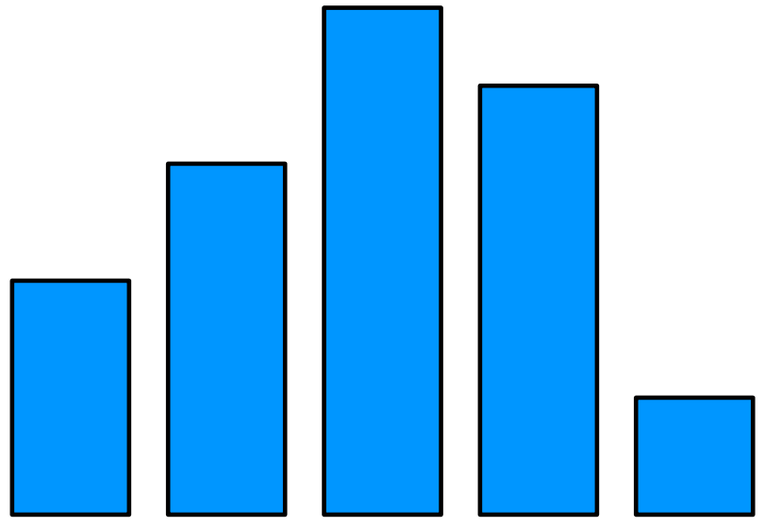
d_2 →

instances

data	d_1	d_2	d_3	d_4	...
visible	v_1	v_2			...
width	w_1	w_2			...
height	h_1	h_2			...
fillStyle	f_1	f_2			...
...	...	...	...	...	...

Property evaluation

Computed property values stored in scene graph



specification

data	λ_1
visible	λ_2
width	λ_3
height	λ_4
fillStyle	λ_5
...	...

d_n →

instances

data	d_1	d_2	d_3	d_4	...
visible	v_1	v_2	v_3	v_4	...
width	w_1	w_2	w_3	w_4	...
height	h_1	h_2	h_3	h_4	...
fillStyle	f_1	f_2	f_3	f_4	...
...	...	...	...	...	...

Property evaluation

Computed property values stored in scene graph

defaults

data	λ_1
visible	λ_2
width	
height	
fillStyle	λ_3
...	...

proto

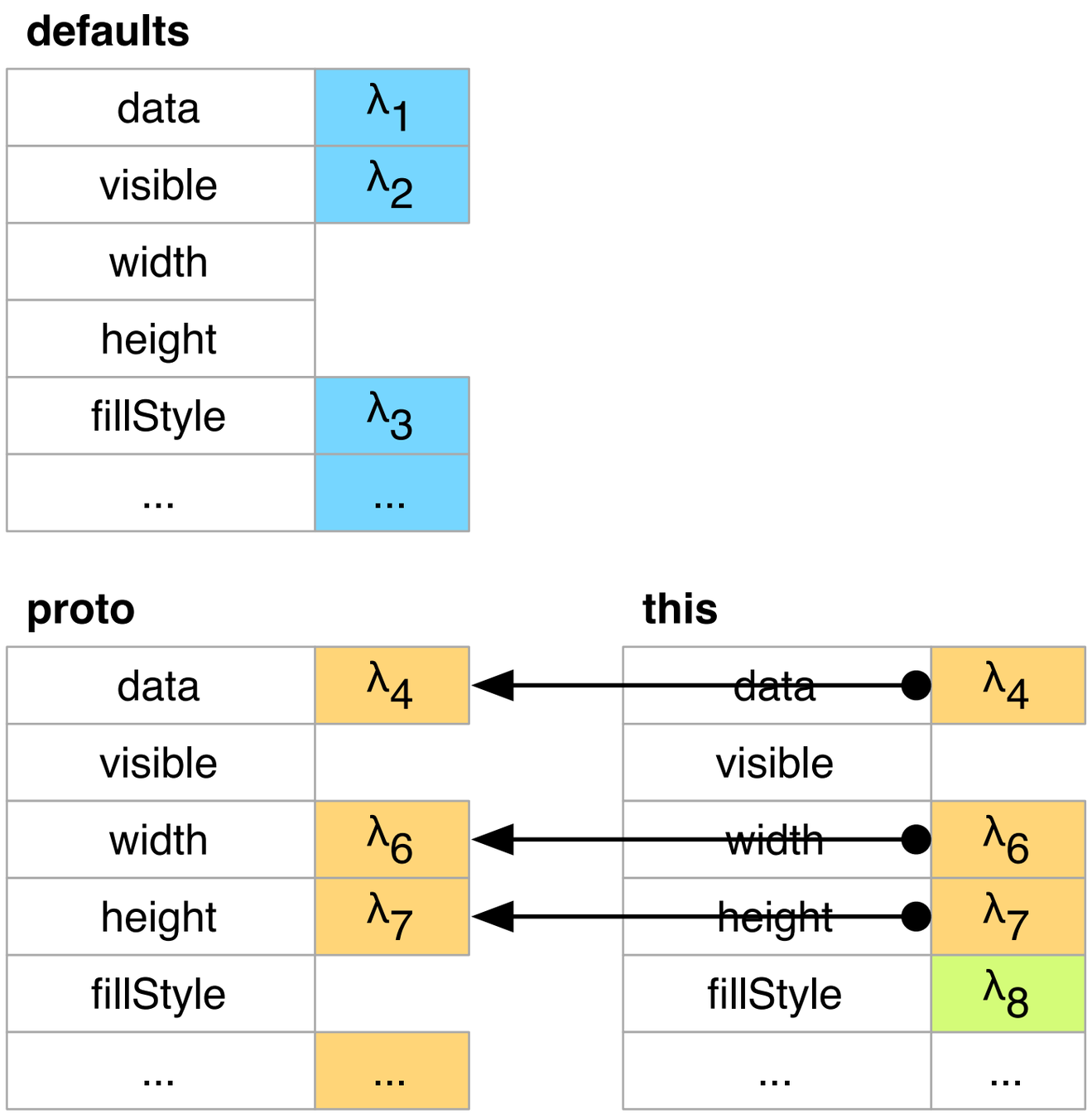
data	λ_4
visible	
width	λ_6
height	λ_7
fillStyle	
...	...

this

data	
visible	
width	
height	
fillStyle	λ_8
...	...

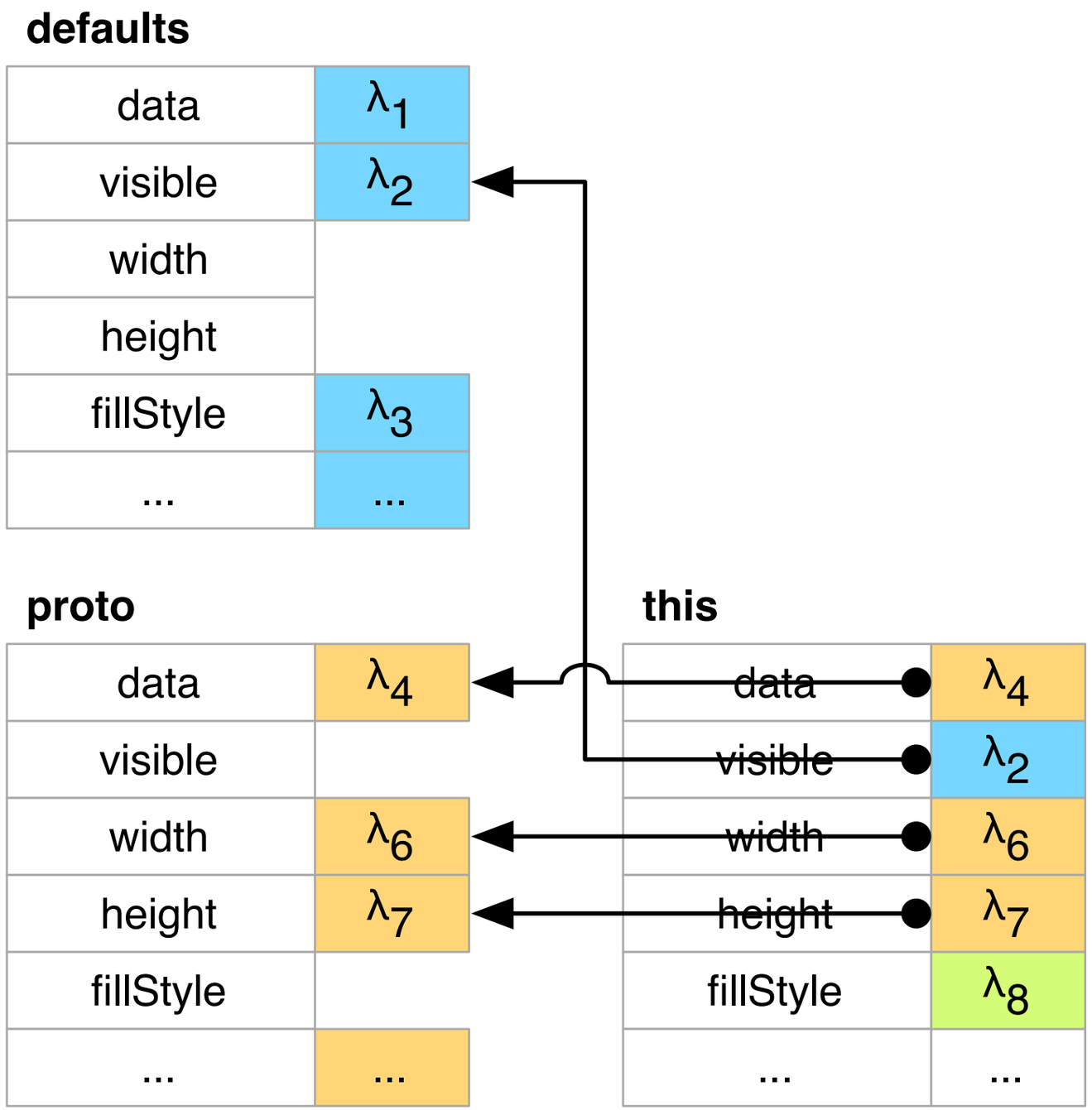
Property inheritance

Prototype chain & type-specific defaults



Property inheritance

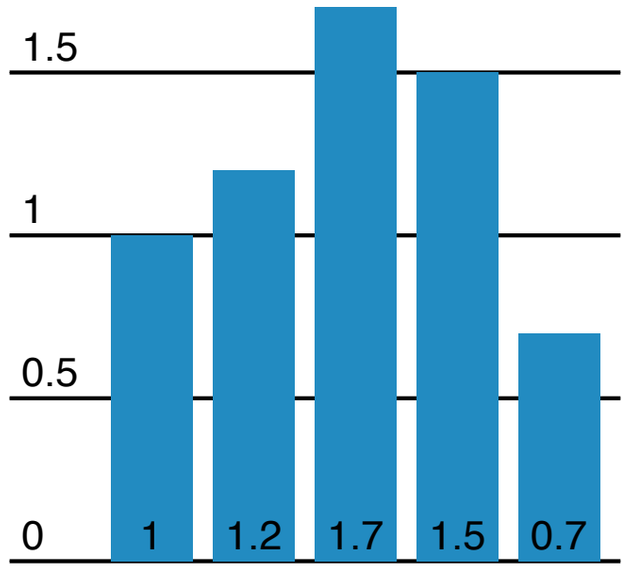
Prototype chain & type-specific defaults



Property inheritance

Prototype chain & type-specific defaults

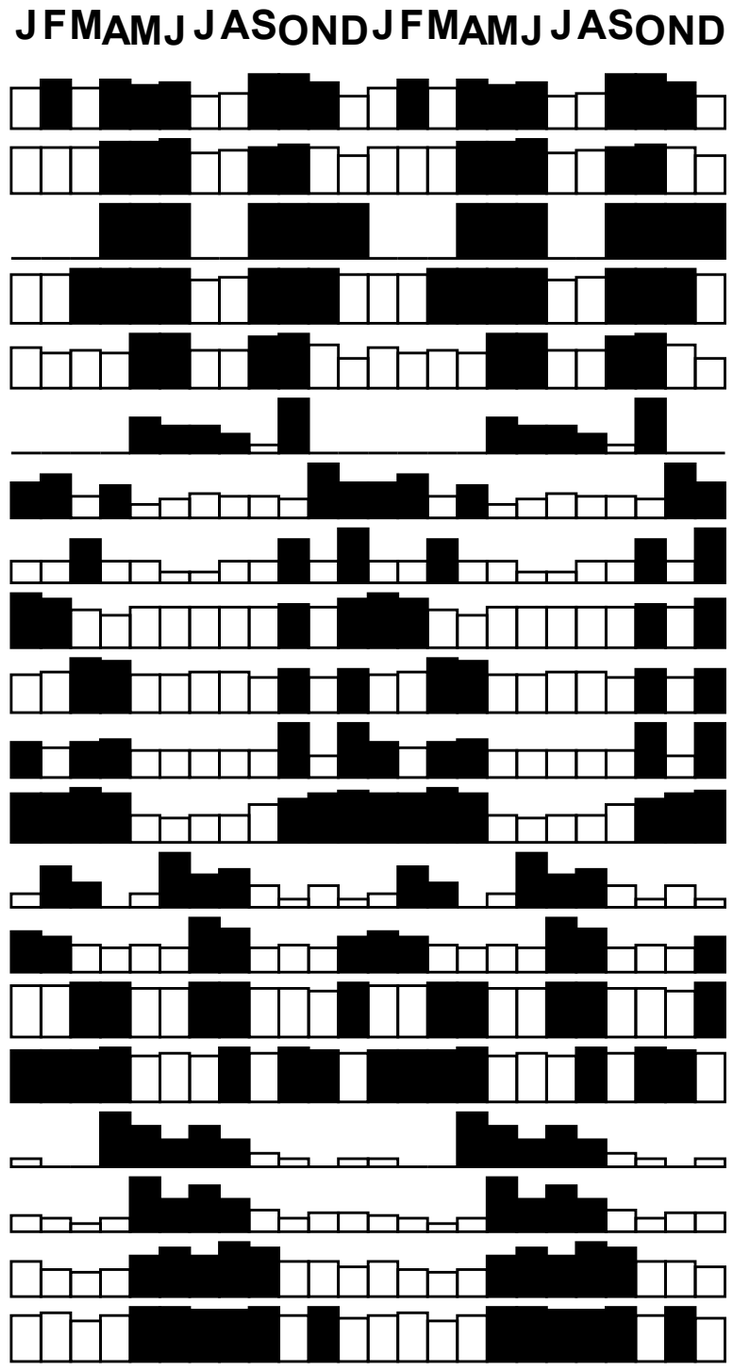
```
vis.add(pv.Rule)
    .data(pv.range(0, 2, .5))
    .bottom(function(d) d * 80)
  .add(pv.Label);

vis.add(pv.Bar)
    .data([1, 1.2, 1.7, 1.5, .7])
    .left(function() this.index * 25 + 25)
    .bottom(0)
    .height(function(d) d * 80)
    .width(20)
  .anchor("bottom").add(pv.Label);
```



Property inheritance

Anchors refer to specific locations on a given mark

```
vis.add(pv.Panel)
    .data(metrics)
    .top(function() this.index * 24)
  .add(pv.Bar)
    .data(function(p) p.values)
    .left(function() this.index * 20)
    .bottom(0)
    .height(function(d) d * 20)
    .width(20)
    ...
```



Bertin's Hotels, 1981

Panels

Replicate marks and traverse hierarchical data

Scales & Layouts

Bidirectional mappings; treemaps, sunbursts, ...

Interaction

Event handlers, tooltips, cursors, local variables...

Data transformation

Nesting, flattening, statistics, sorting, filtering...

Additional features

1. Fundamentals

Language design and features

2. Demo

Examples of what you can do with Protovis

3. Sandbox

Try it out!

Outline

sandbox.protovis.org