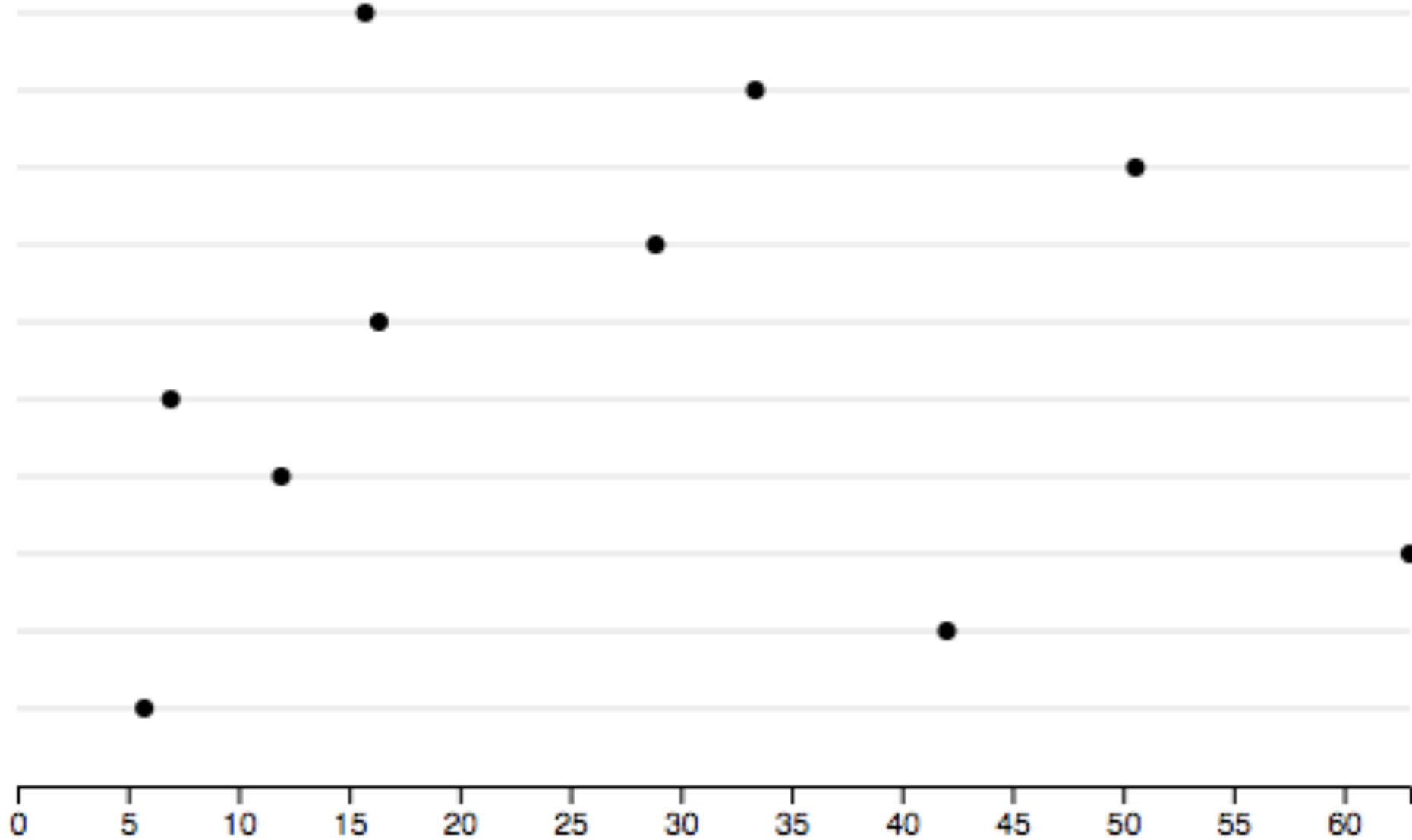
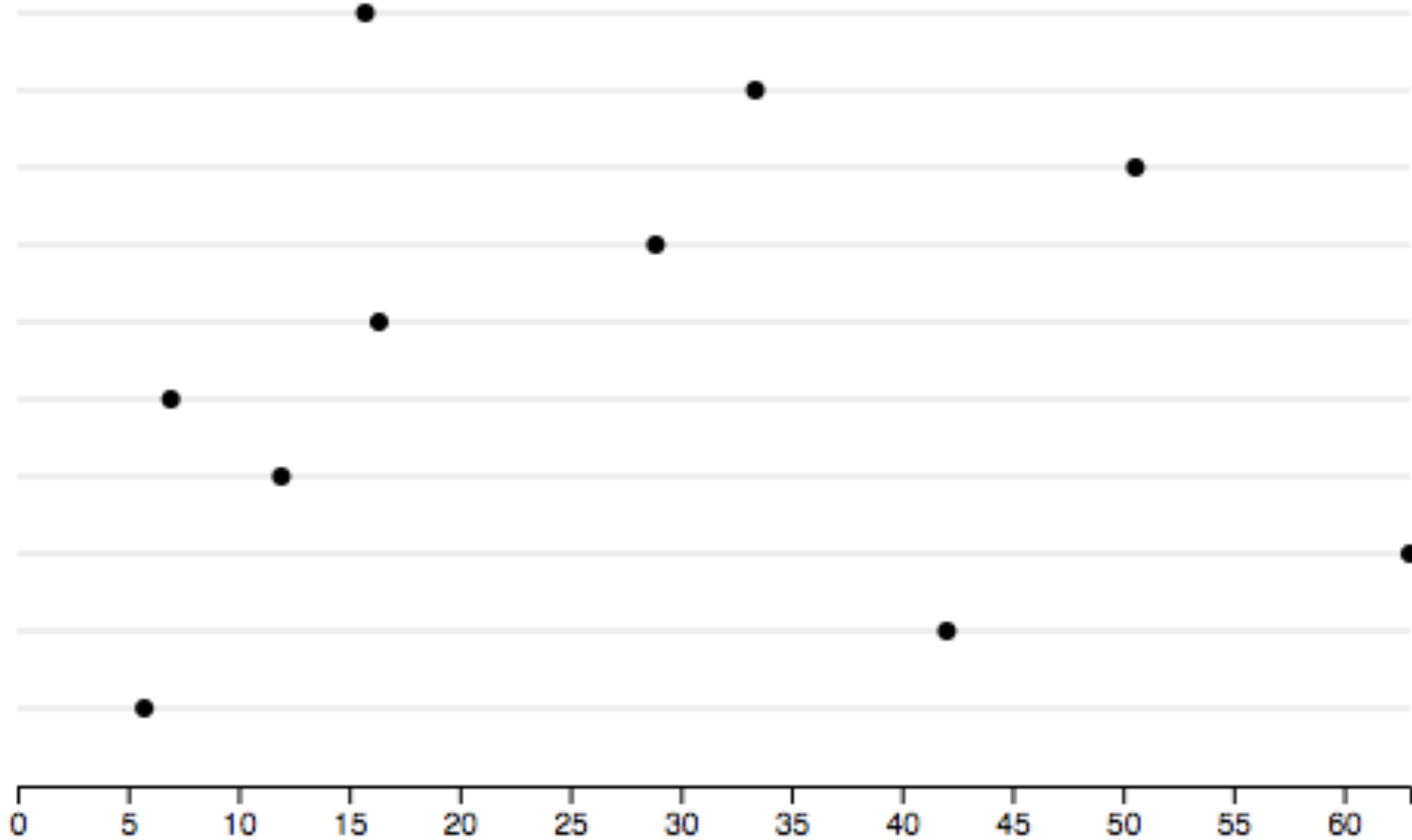


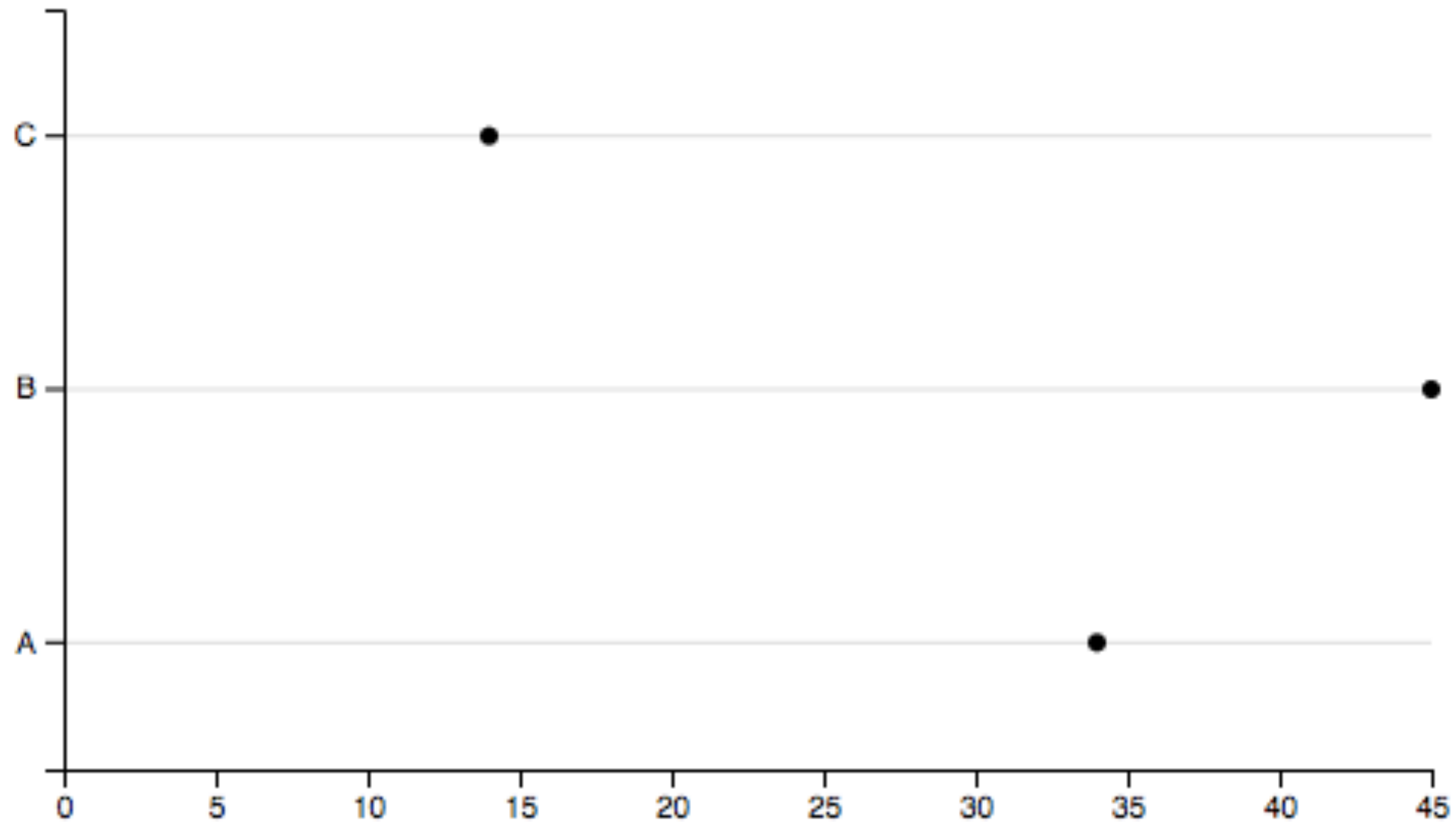
Homework Sneak Preview



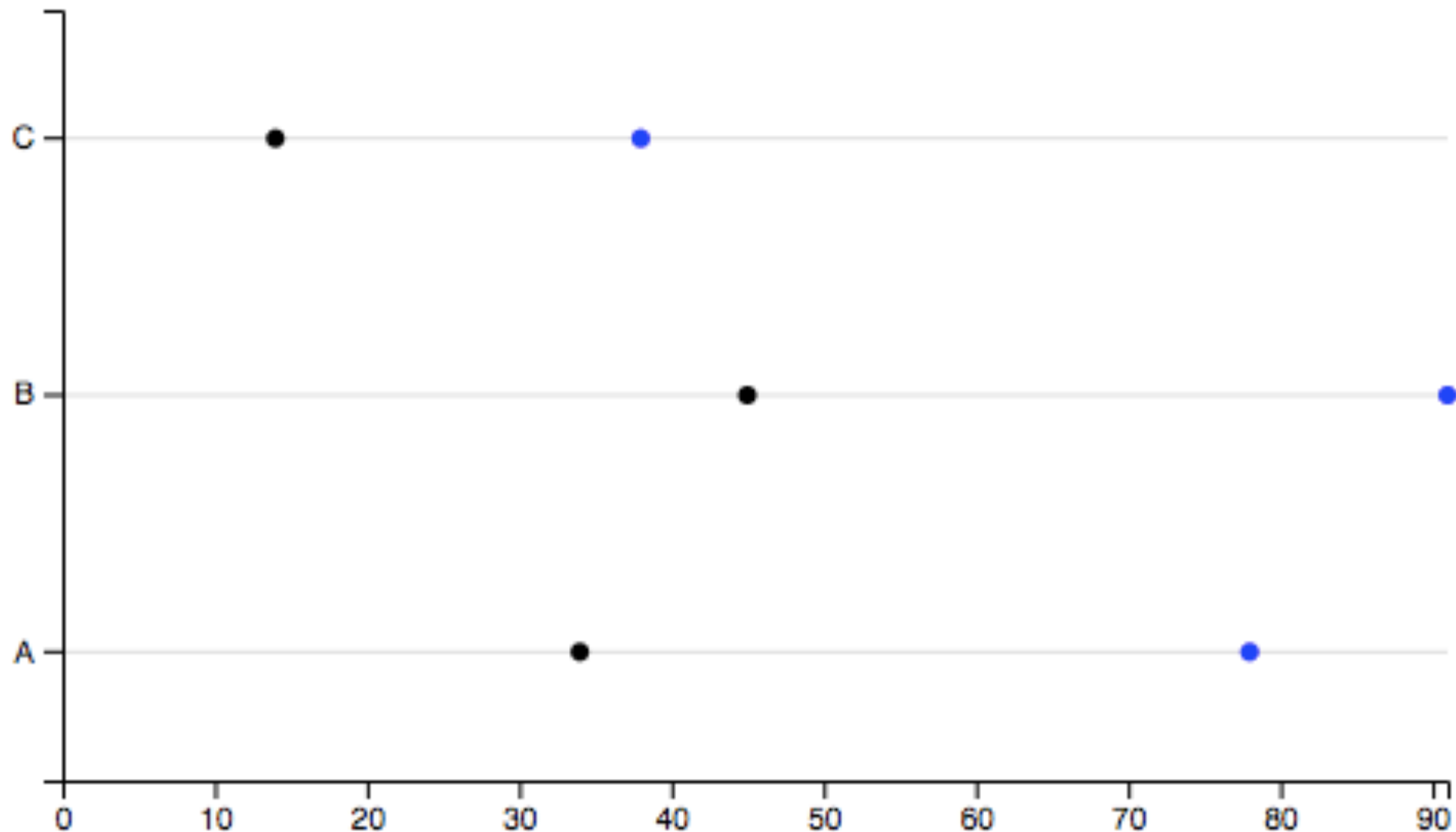
Homework Sneak Preview



Homework Sneak Preview



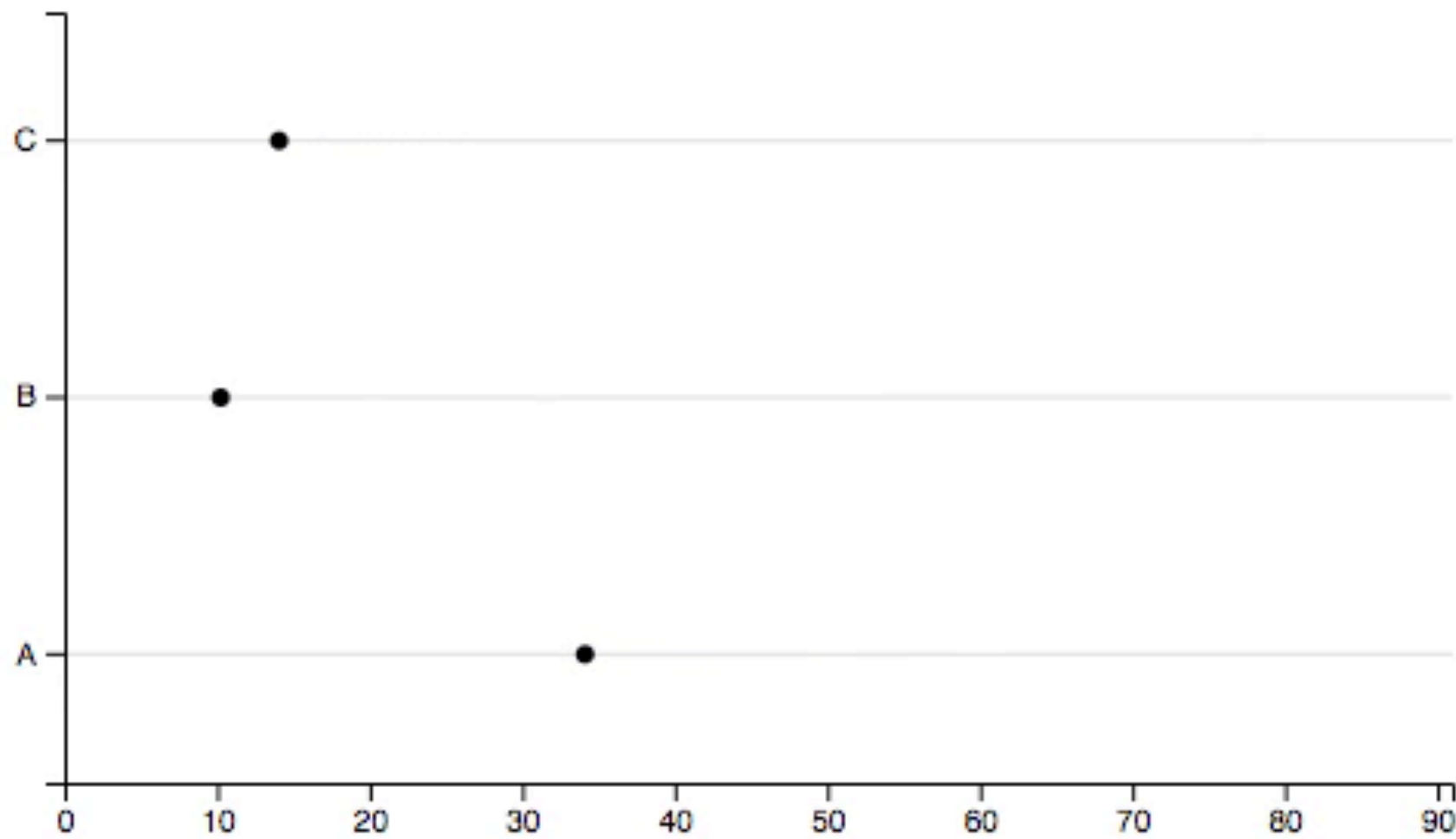
Homework Sneak Preview



Homework Sneak Preview

1975

Click here to change year



Transitions



Transitions

- **Use to show process, how something moved from A to B**

<https://www.bloomberg.com/graphics/2015-whats-warming-the-world/>

<https://www.nytimes.com/interactive/2018/03/27/upshot/make-your-own-mobility-animation.html>

Transitions

- **Do not run two transitions on the same selection at the same time (see p. 172)**

ex. (doesn't work)

```
d3.selectAll("circle").transition()  
    .duration(2000).attr("cx", "300");
```

```
d3.selectAll("circle").transition()  
    .duration(2000).attr("fill", "green");
```


Transitions

Don't overdo it.



Transitions

- **Transition from something to something, not nothing to something (applies esp. to new elements)**

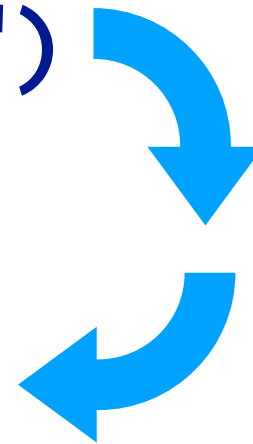
ex. (doesn't work)

```
d3.select("svg").append("circle")  
  .transition()  
  .duration(2000)  
  .attr("cx", "200")  
  .attr("cy", "100")  
  .attr("r", "25")  
  .attr("fill", "red");
```

Transitions

ex. (works)

```
d3.select("svg").append("circle")  
  .attr("cx", "200")  
  .attr("cy", "100")  
  .attr("r", "25")  
  .attr("fill", "lightblue")  
  .transition()  
  .duration(2000)  
  .attr("fill", "red");
```



Transitions

- **Think carefully about what you want to happen, and then decide what goes before and after the transition**

Plan

1. new bars appear
in proper place
immediately
w/ orange fill

2. fill transitions to blue

`.enter().append("rect")`

1. orange fill

`.merge(bars)`

2. reposition all bars

`.transition().duration(2000)`

3. blue fill

Transitions

Plan

1. new bar appears off to the right side
2. all bars gradually scooch left into place

`.enter().append("rect")`

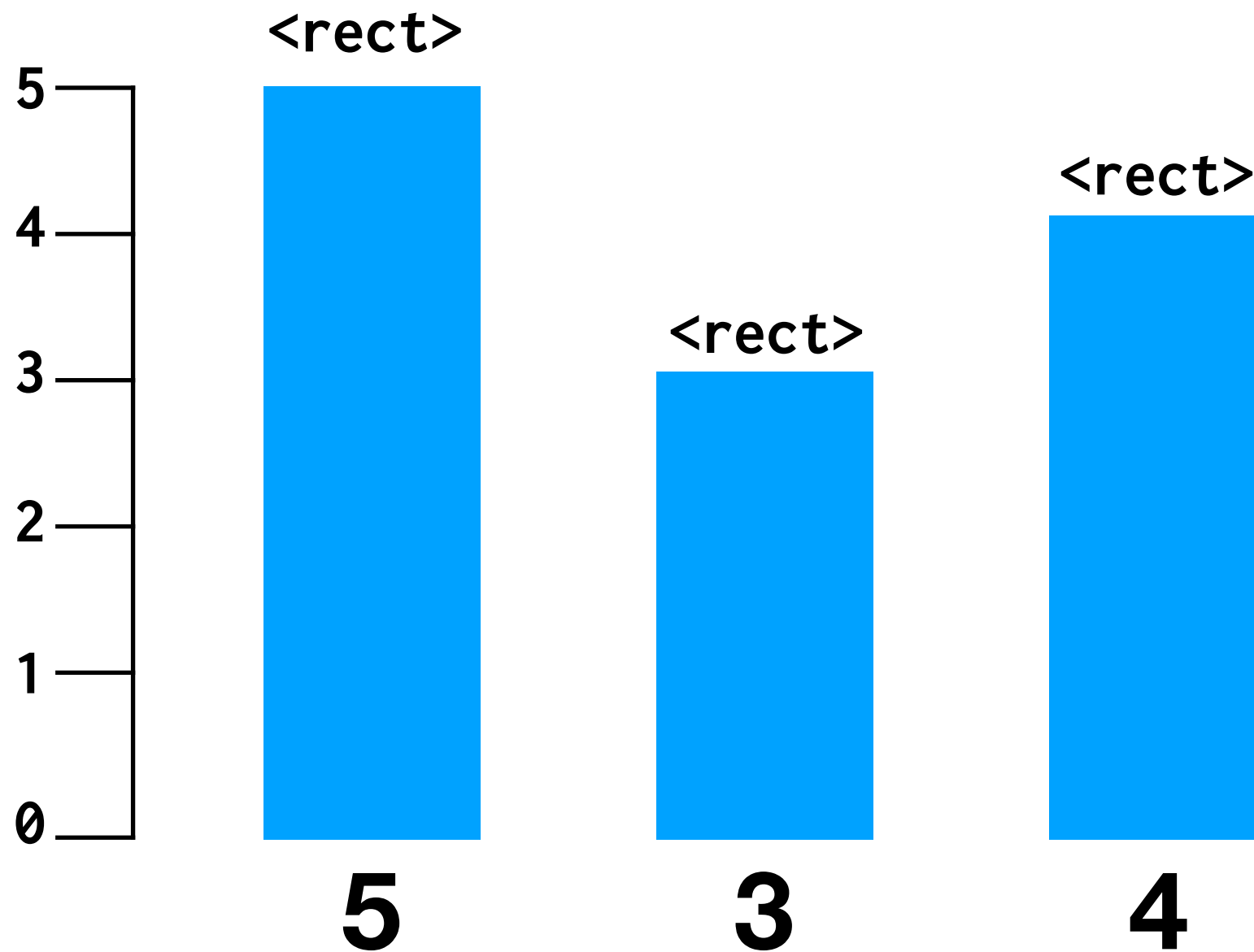
1. position new bar to the right of svg width

`.merge(bars)`

`.transition().duration(2000)`

2. reposition all bars

Maintain object constancy



Object constancy

- **Join data by key, not in order of SVG elements**

ex.

```
var bardata = [ {key: 0, value: 300},  
                 {key: 1, value: 100},  
                 {key: 2, value: 150},  
                 {key: 3, value: 220},  
                 {key: 4, value: 70},  
                 {key: 5, value: 270}];
```

Object constancy

- Replace every "d" with "d.value"
- Specify the key when joining data:

```
.data(bardata, d => d.key);
```

- Use a key when adding data:

```
bardata.push({ key: newkey,  
               value: newvalue });
```


Object constancy

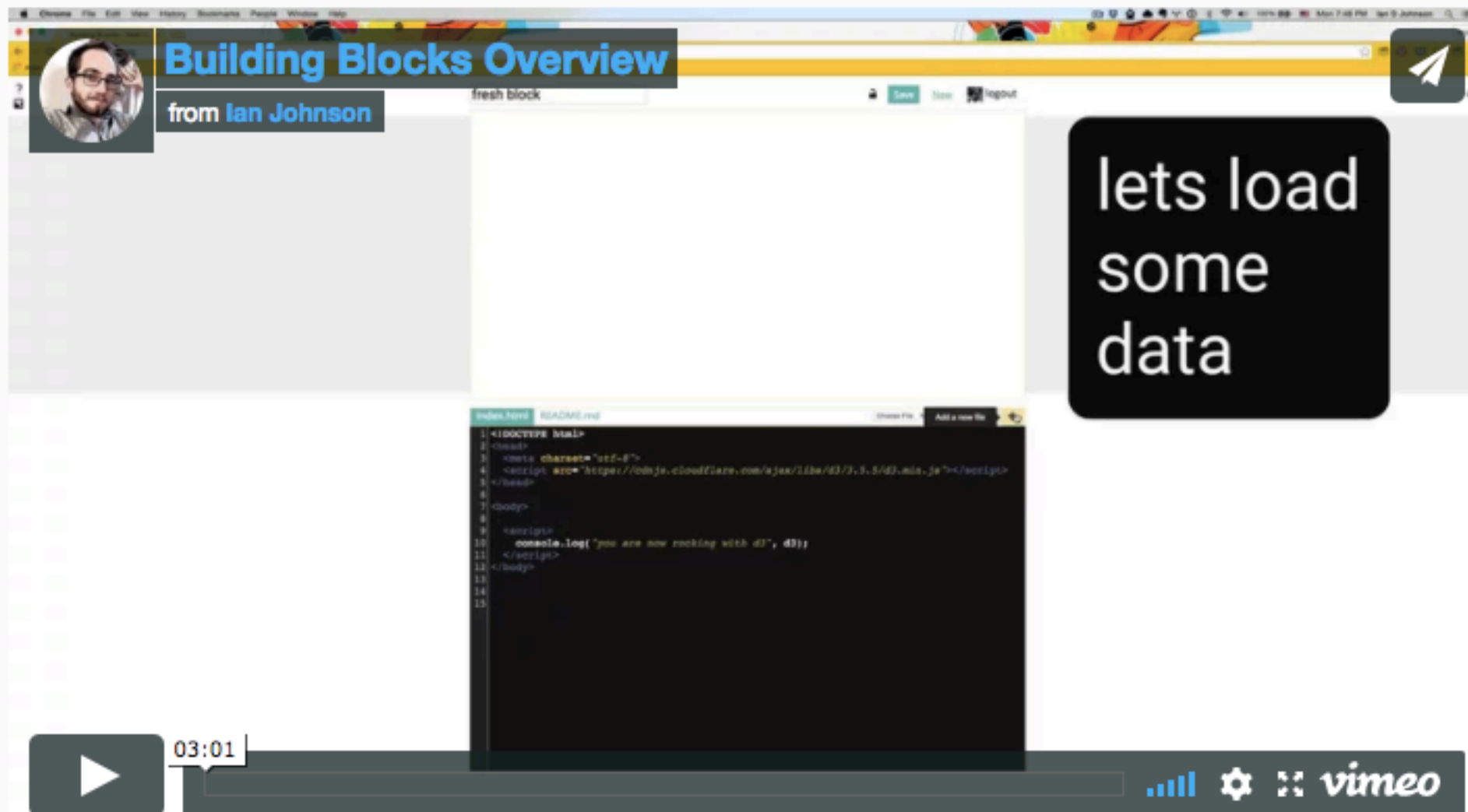
- Use `.data(bardata, d => d.key);` to view keys in console (not `d3.keys()`!)
- Note: in this case, we're still using the index, not the key, as the domain for `xScale`

blocks.org, blockbuilder.org

Block Builder

QUICKLY CREATE, EDIT AND FORK D3.JS EXAMPLES

Are you learning d3 or trying out new ideas? Block Builder is an in-browser code editor built for creating and sharing d3.js examples. Check out this short video for an overview of how it works!



Set up a local server

- **SimpleHTTPServer** (pp. 64-65)
- **http-server**

<https://www.npmjs.com/package/http-server>

first install npm: <https://www.npmjs.com/>