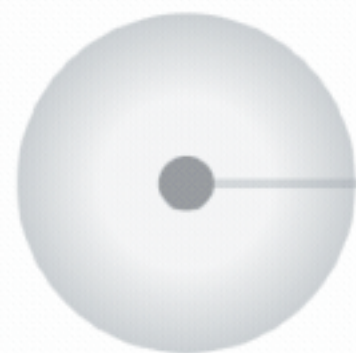


Animating with

Customer



1

Active carts



0

Checking out



0

Purchased

✕ Close custom settings

Customer behavior settings

Page views settings

Total Sessions settings

Polling interval: 10000 (ms)



Increment range: 1-2




```
import { useSprings, animated, config } from "react-spring";
```

```
const radiusSprings = useSprings(  
  data.length,  
  data.map((dataPoint, index) => ({  
    radius: radiusScale(dataPoint.count),  
    config: config.molasses,  
  })))  
);
```

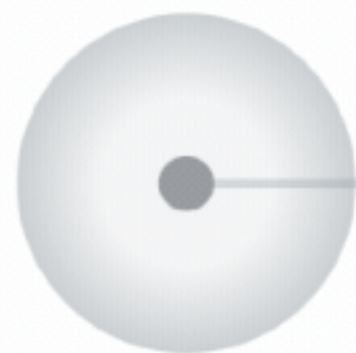


```
{data.map((dataPoint, index) => (  
  <circle  
    cx={xScale(index)}  
    r={radiusScale(dataPoint.value)}  
    // ...  
  >  
))}
```



```
{data.map((dataPoint, index) => (  
  <animated.circle  
    cx={xScale(index)}  
    r={springs[index].radius}  
    // ...  
  />  
))}
```

Customer



1

Active carts



0

Checking out



0

Purchased

✕ Close custom settings

Customer behavior settings

Page views settings

Total Sessions settings

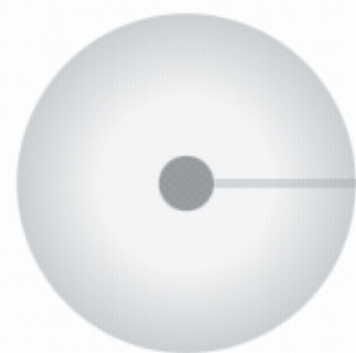
Polling interval: 10000 (ms)



Increment range: 1-2



Customer



1

Active carts



0

Checking out



0

Purchased

✕ Close custom settings

Customer behavior settings

Page views settings

Total Sessions settings

Polling interval: 10000 (ms)

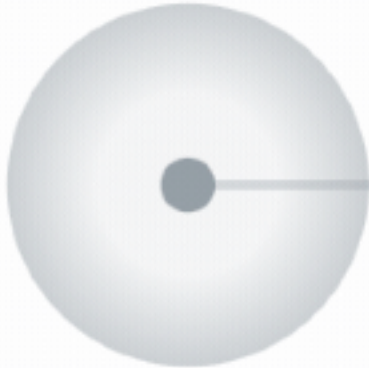


Increment range: 1-2



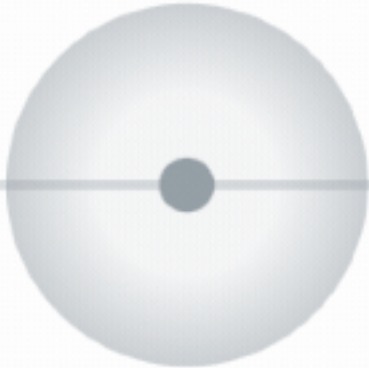
Keeping track of

Customer



1

Active carts



1

Checking out



0

Purchased

← Open custom settings


```
const radiusSprings = useSprings(  
  data.length,  
  data.map((dataPoint, index) => ({  
    radius: radiusScale(dataPoint.count),  
    config: config.molasses,  
  })))  
);
```

```
export function usePrevious(value) {  
  const ref = useRef();  
  useEffect(() => {  
    ref.current = value;  
  });  
  return ref.current;  
}
```

```
const previousData = usePrevious(data);
```