Week 11

Force Simulation

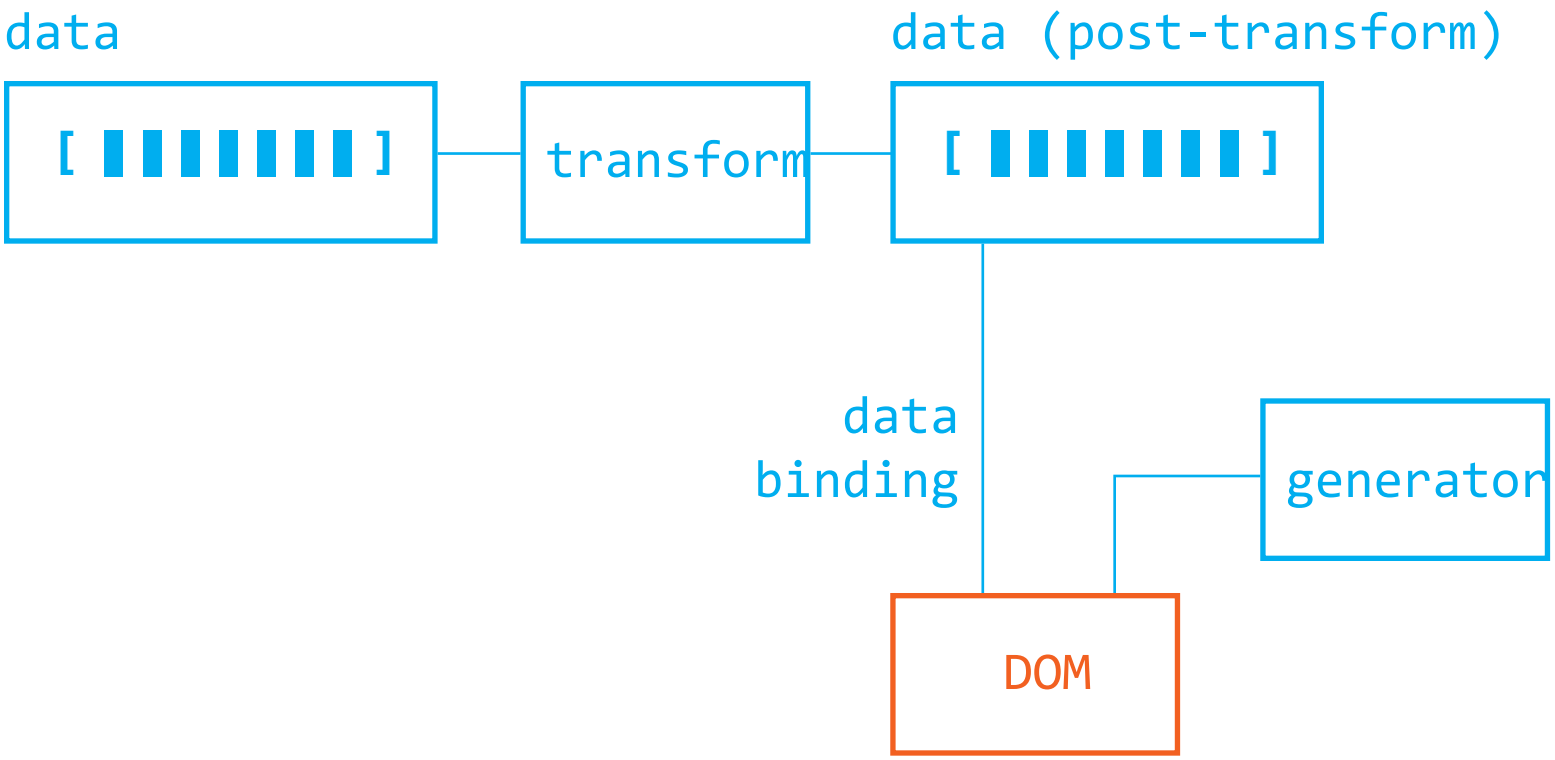
Today

1. Understand the basics of a force simulation, including how it transforms data, what “ticking” does, and how to set up different forces

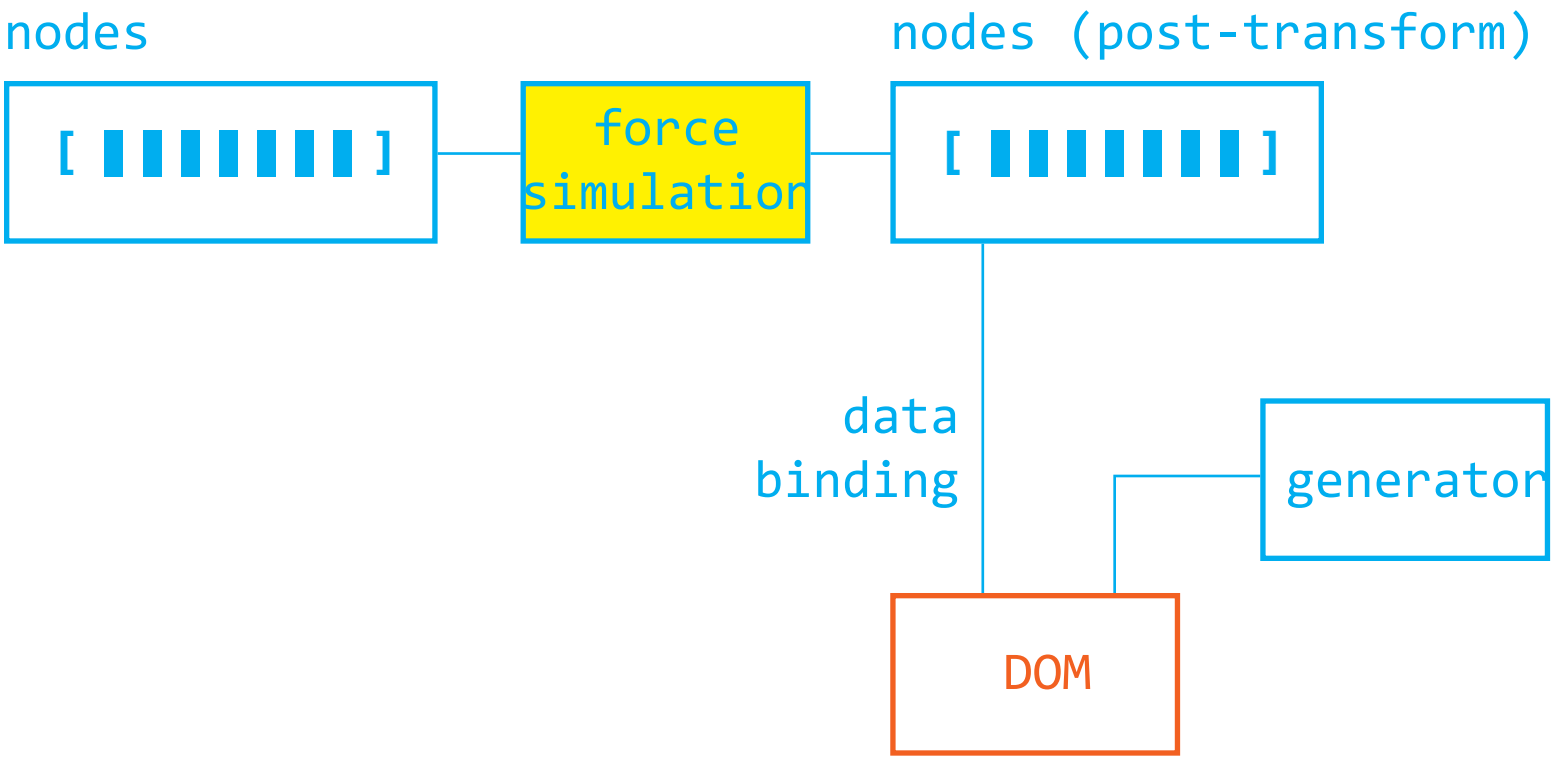
2. Use cases for force simulation

3. Understand the relationship between data structure and visualization types

Force Simulation Transforms Data



Force Simulation Transforms Data

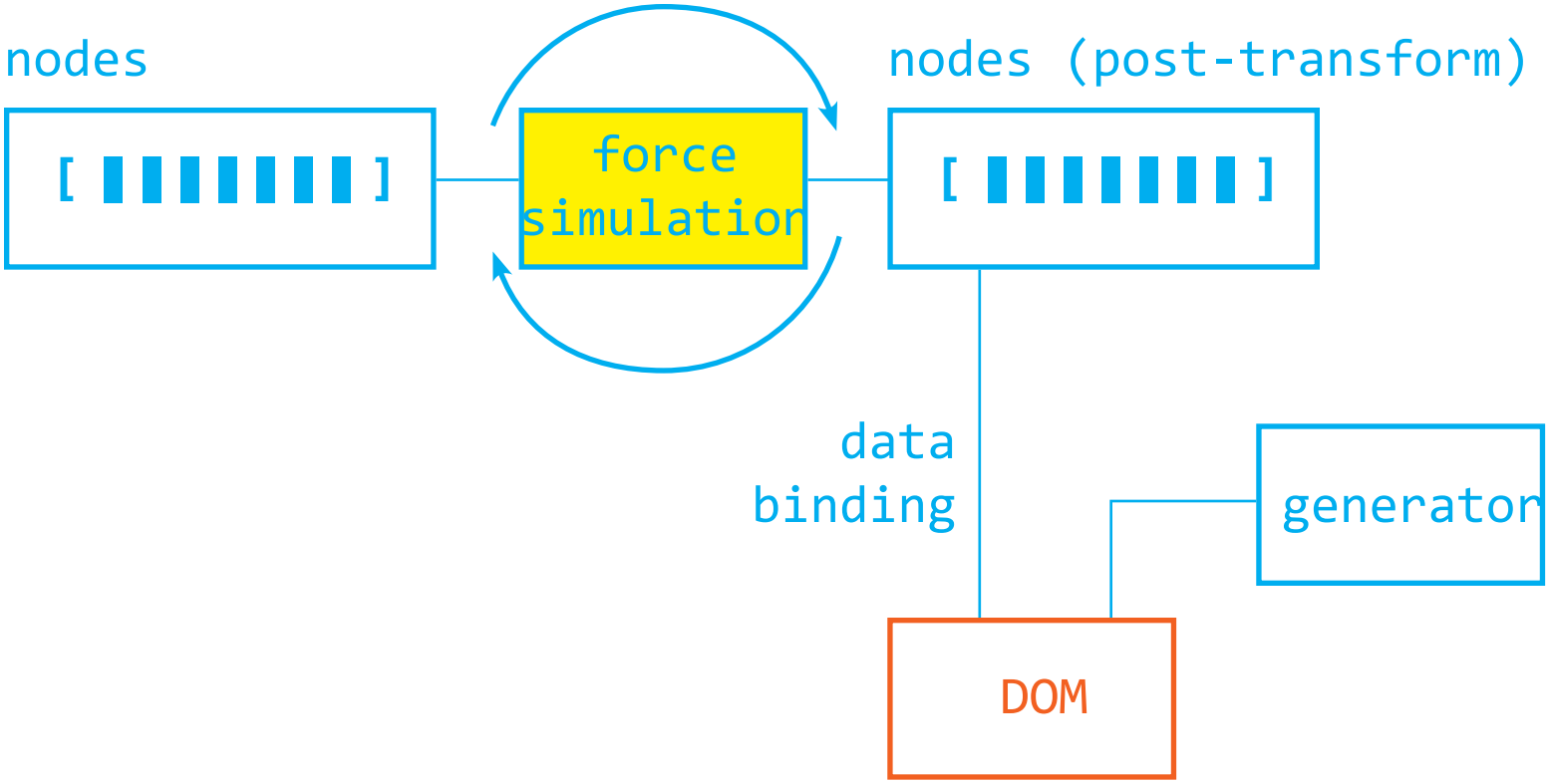


Force Simulation Transforms Data

Post-force simulation, elements in the `nodes` array will have the following properties automatically assigned:

```
index
x --> current position-x
y --> current position-y
vx --> vector pointing to where node aiming for
vy
```

Force Simulation Runs Repeatedly



Force Simulation Runs Repeatedly

Each iteration of the force simulation is called a **tick**, and at each tick, the `x,y,vx,vy` attributes of each node is recomputed.

To listen to each tick:

```
simulation
    .on('tick', function(){
        //each tick
        //Reposition each visual element based on
the latest x,y
    });
```

Force Simulation Runs Repeatedly

Ticking stops when the internal value **alpha** of the simulation runs down to below a certain minimum.

To see what the current simulation alpha is:

```
simulation
    .on('tick', function(){
        console.log(this.alpha);
    });
```

Other Relevant Simulation Settings

```
simulation
    .alpha()
    .alphaDecay()
    .alphaMin()
    .alphaTarget()
    .on('tick', function(){
       console.log(this.alpha);
    })
    .on('end', function(){})
    .stop()
    .restart()
    .nodes(nodes);
```

Exercise 1

Let's set up a simulation and pass in a array of 50 random point data (with x, y attributes)

Adding Forces to Simulation

Force simulations will optimize a layout based on the interaction between various forces. To add a force to a simulation:

```
simulation
    .force('forceName', forceFunction)
    ...
```

d3 has a number of built-in, configurable force functions, and even allows us to build custom force functions!

Exercise 1

Based on existing force simulation, examine the use of the different forces

Adding Forces to Simulation

A couple of force function worth highlighting:

```
d3.forceCollide()
    .radius(function(d){...});

d3.forceX()
    .x(function(d){...});

d3.forceY()
    .y(function(d){...});
```

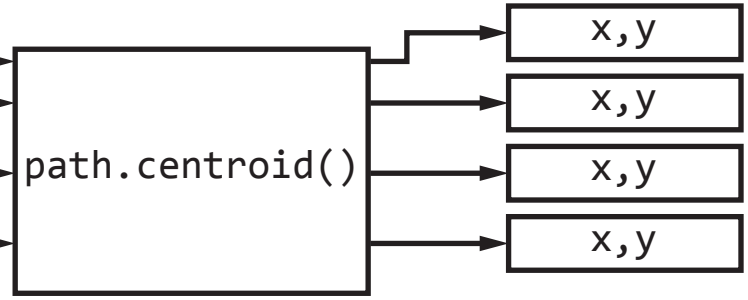
Force Simulation: Use Cases

With these forces, combined with what learned in the previous lecture, we can create a dorling cartogram.

Force Simulation: Use Cases

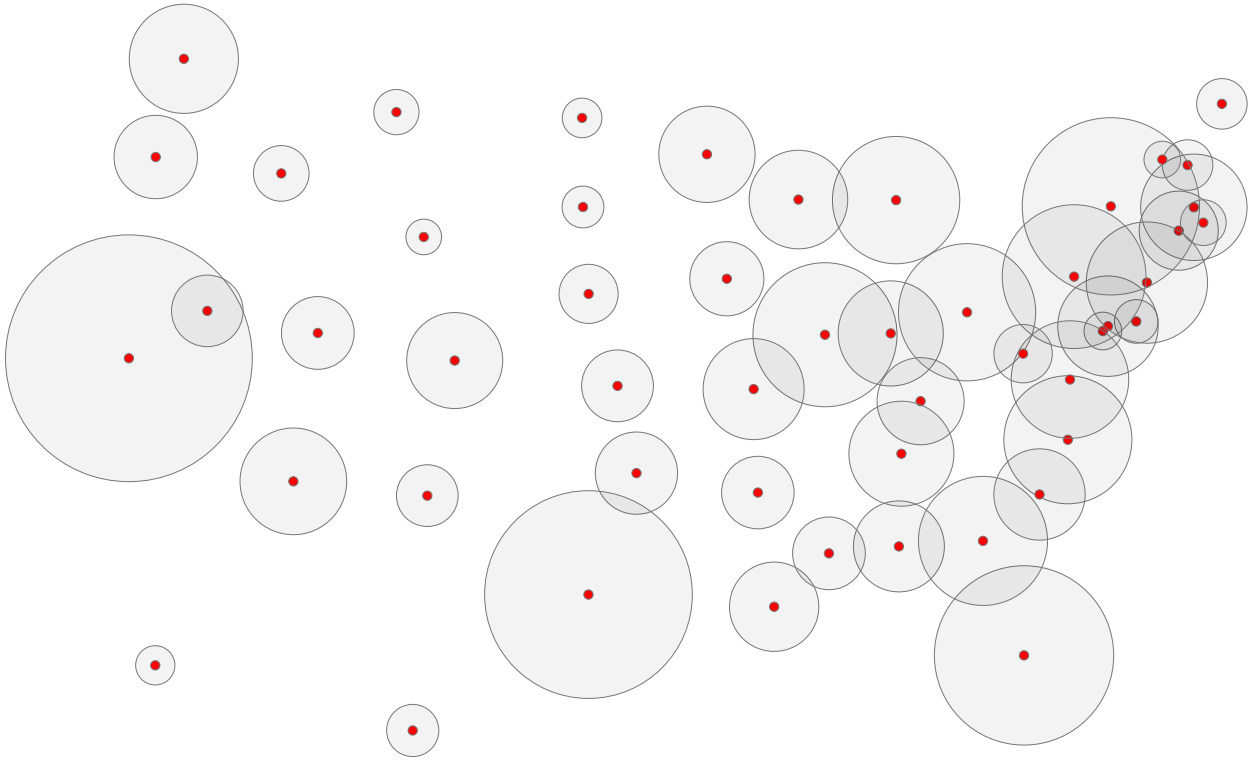
With these forces, combined with what learned in the previous lecture, we can create a dorling cartogram.

```
{
  type: “FeatureCollection”,
  features:[
    feature
    feature
    feature
    feature
  ]
}
```



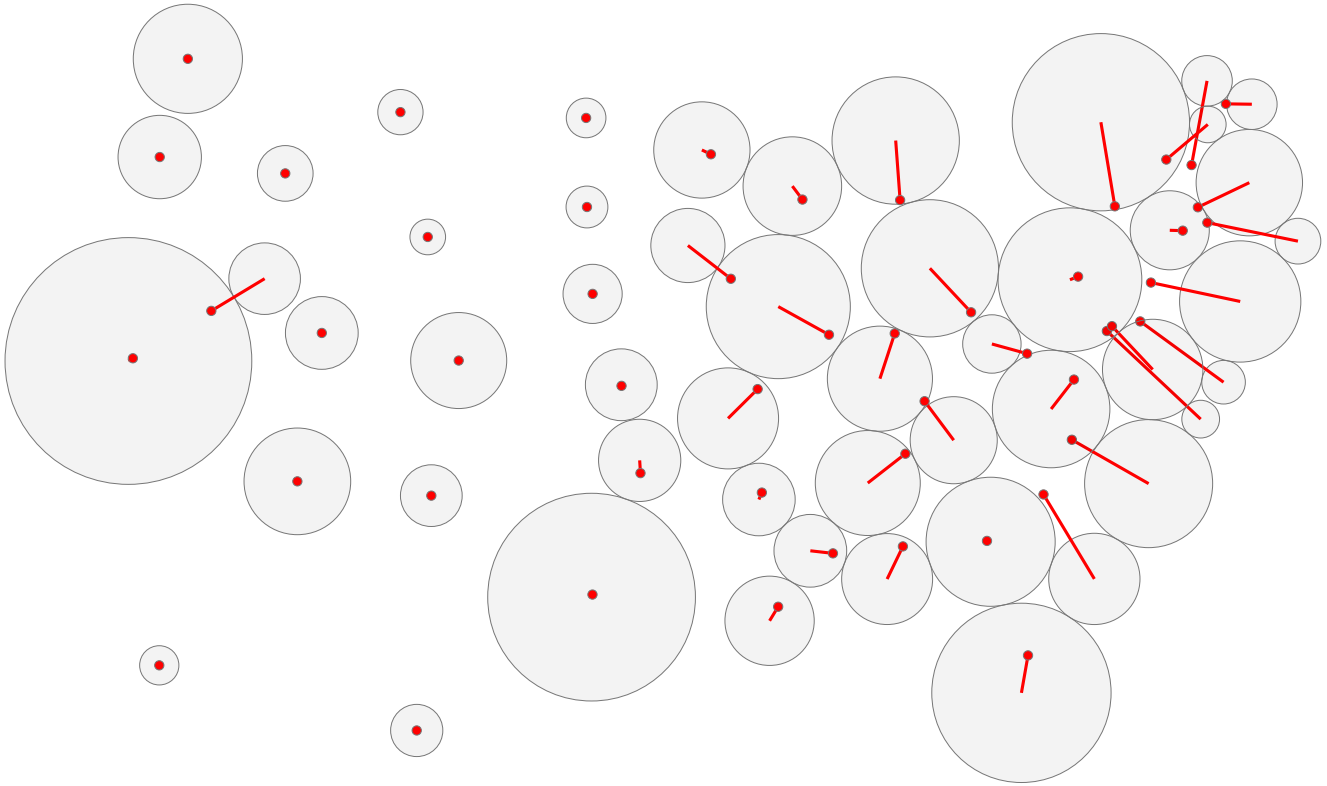
Force Simulation: Use Cases

With these forces, combined with what learned in the previous lecture, we can create a dorling cartogram.



Force Simulation: Use Cases

With these forces, combined with what learned in the previous lecture, we can create a dorling cartogram.



Force Simulation: Use Cases

Other use cases?

Exercise 2

Implement the dorling cartogram

Exercise 3: Force Simulation with Nodes and Links

Force Simulation with Link Forces

For a node-link based force simulation, two arrays are necessary

```
//nodes
[
{name: ‘New York’, ...},
{name: ‘London’,...},
{name: ‘Hong Kong’,...},
{name: ‘Sydney’,...},
{name: ‘New Dehli’,...}
...
]
```

```
//links
[
{source:0, target:4},
{source:1, target:2},
...
]
```

Force Simulation with Link Forces

For a node-link based force simulation, two arrays are necessary

```
var simulation = d3.forceSimulation()
    .nodes(nodes)
    .force('link', d3.forceLink(links))
    .on('tick',...);
```

Force Simulation with Link Forces

Even though these two arrays are previous unrelated, the force simulation will "link them up"

```
//nodes
[
{name: 'New York', ...},
{name: 'London',...},
{name: 'Hong Kong',...},
{name: 'Sydney',...},
{name: 'New Dehli',...}
...
]
```

```
//links
[
{source:0, target:4},
{source:1, target:2},
...
]
```

Force Simulation with Link Forces

What if the links array comes in a different format?

```
//nodes
[
{name: ‘New York’, ...},
{name: ‘London’,...},
{name: ‘Hong Kong’,...},
{name: ‘Sydney’,...},
{name: ‘New Dehli’,...}
...
]
```

```
//links
[
{source:‘New York’, target:‘London’},
{source:‘Hong Kong’, target:‘Sydney’},
...
]
```

Force Simulation with Link Forces

What if the links array comes in a different format?

```
var forceLink = d3.forceLink(links)
    .id(function(d){return d.name});
```

Review: Week 11

Representation

Data Manipulation

`d3.forceSimulation`

`force functions`

Interaction

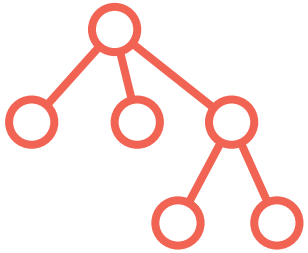
Extras

Finding the centroid of geographic features

Data Structures

Point

Hierarchy



Line/Serial

Graph