

# Visualisierung Workshop - Layouts und Geomapping

---

Lali Nurtaev und Daniel Heuser

# Ablauf des Workshops

---

## 1. Etappe

1. Einführung Layouts
2. Live Coding Vorführung
3. Ausblick (Layouts)

## 2. Etappe

1. Einführung Geomapping
2. Do it yourself (15 Minuten)
3. Ausblick (Geomapping)

## 3. Etappe

1. Diskurs

# Einführung Layouts

---

## Was sind Layouts?

In D3 haben Layouts keine direkte visuelle Ausgabe. D3 Layouts wandeln bereitgestellte Daten auf und wandeln sie auf andere Weise um, wodurch neue Daten erzeugt werden.

# Einführung Layouts

---

Beispiel:

```
const dataset = [5, 10, 20, 45, 6, 25];  
console.log(dataset);
```

```
▼ Array(6) [ 5, 10, 20, 45, 6, 25 ]  
  0: 5  
  1: 10  
  2: 20  
  3: 45  
  4: 6  
  5: 25  
  length: 6
```

# Einführung Layouts

---

Beispiel:

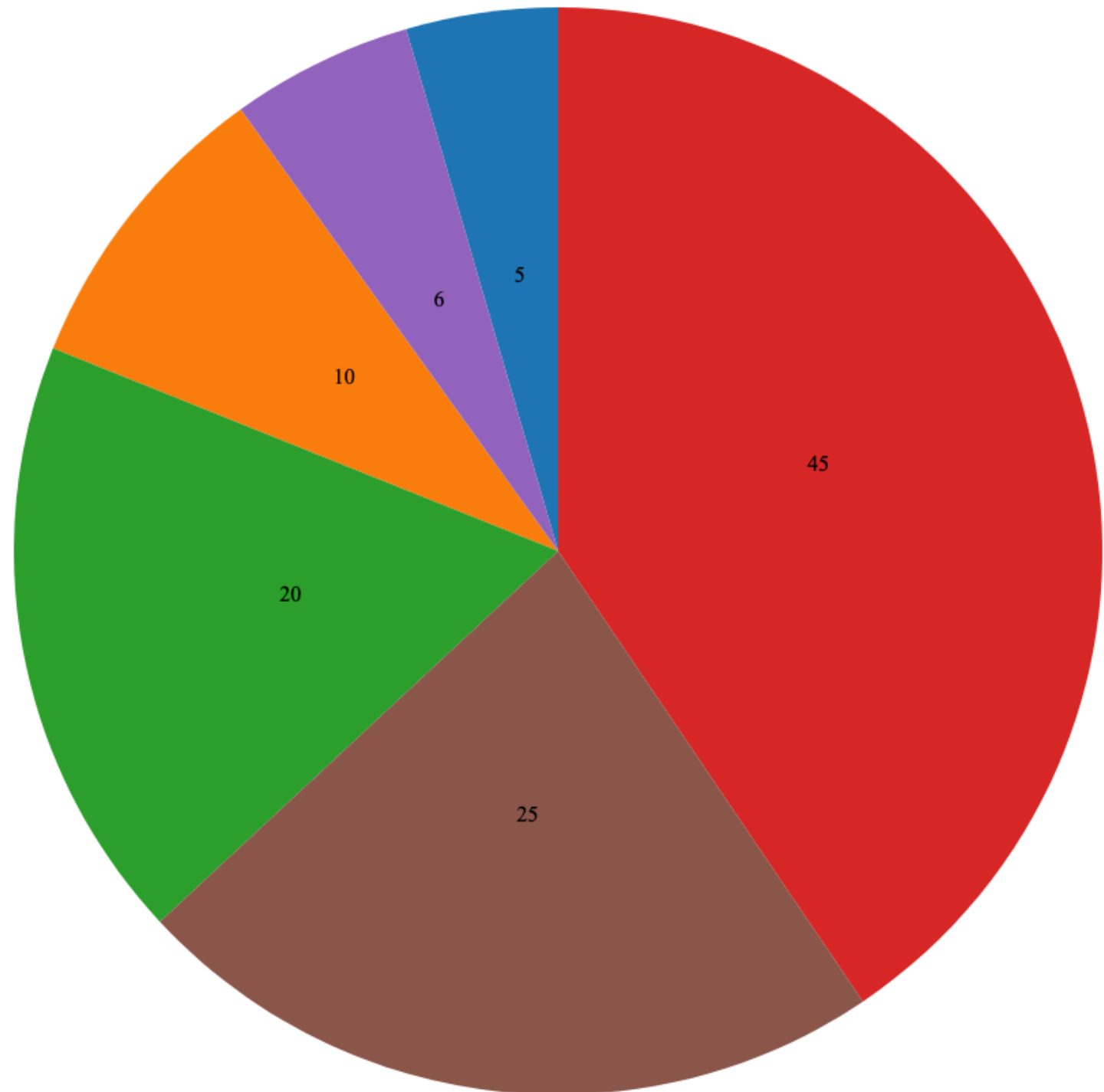
```
const pie = d3.pie();  
console.log(pie(dataset));
```

```
▼ Array(6) [ {...}, {...}, {...}, {...}, {...}, {...} ]  
  ▼ 0: Object { data: 5, index: 5, value: 5, ... }  
    data: 5  
    endAngle: 6.283185307179586  
    index: 5  
    padAngle: 0  
    startAngle: 6.000158941991317  
    value: 5  
    ▶ <prototype>: Object { ... }  
  ▼ 1: Object { data: 10, index: 3, value: 10, ... }  
    data: 10  
    endAngle: 5.660527303765393  
    index: 3  
    padAngle: 0  
    startAngle: 5.094474573388854  
    value: 10  
    ▶ <prototype>: Object { ... }  
  ▶ 2: Object { data: 20, index: 2, value: 20, ... }  
  ▶ 3: Object { data: 45, index: 0, value: 45, ... }  
  ▶ 4: Object { data: 6, index: 4, value: 6, ... }  
  ▶ 5: Object { data: 25, index: 1, value: 25, ... }  
    length: 6
```

# Live Coding Vorführung

---

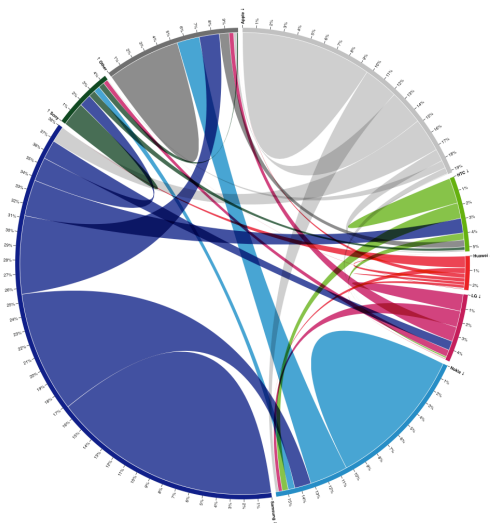
Hinweis: Packt das Handout aus



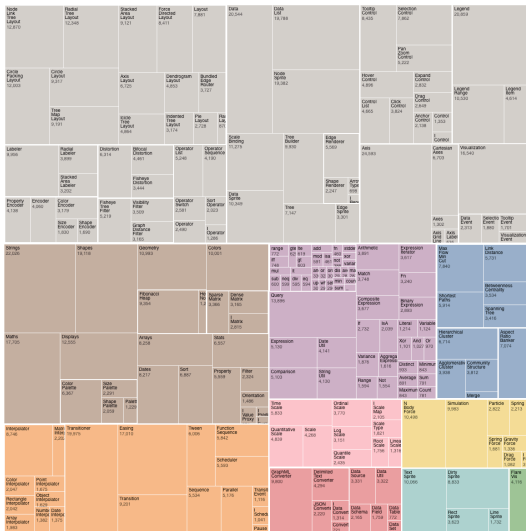
# Ausblick (Layouts)

Welche Layouts gibt es noch?

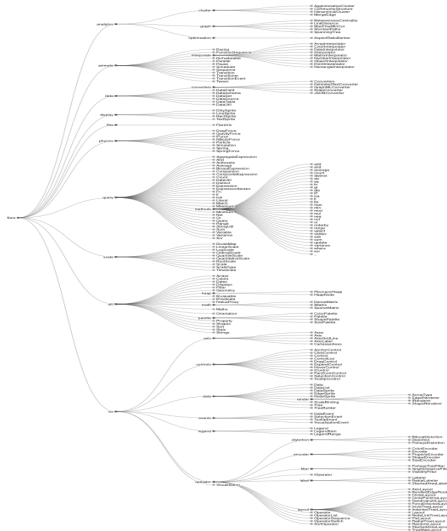
Chord



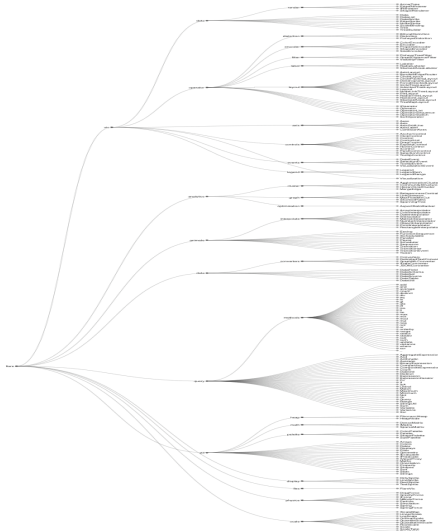
Treemap



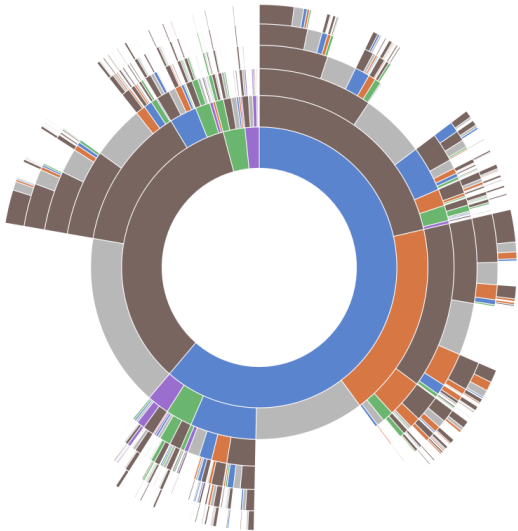
Tree



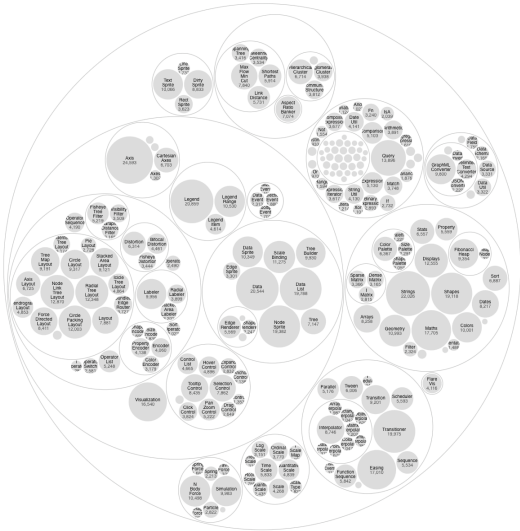
Cluster



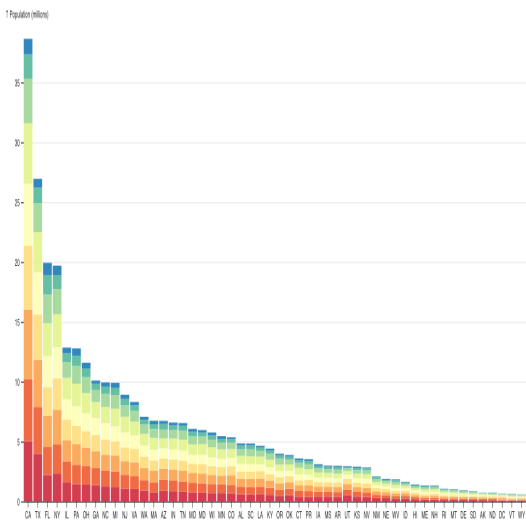
Partition



Pack



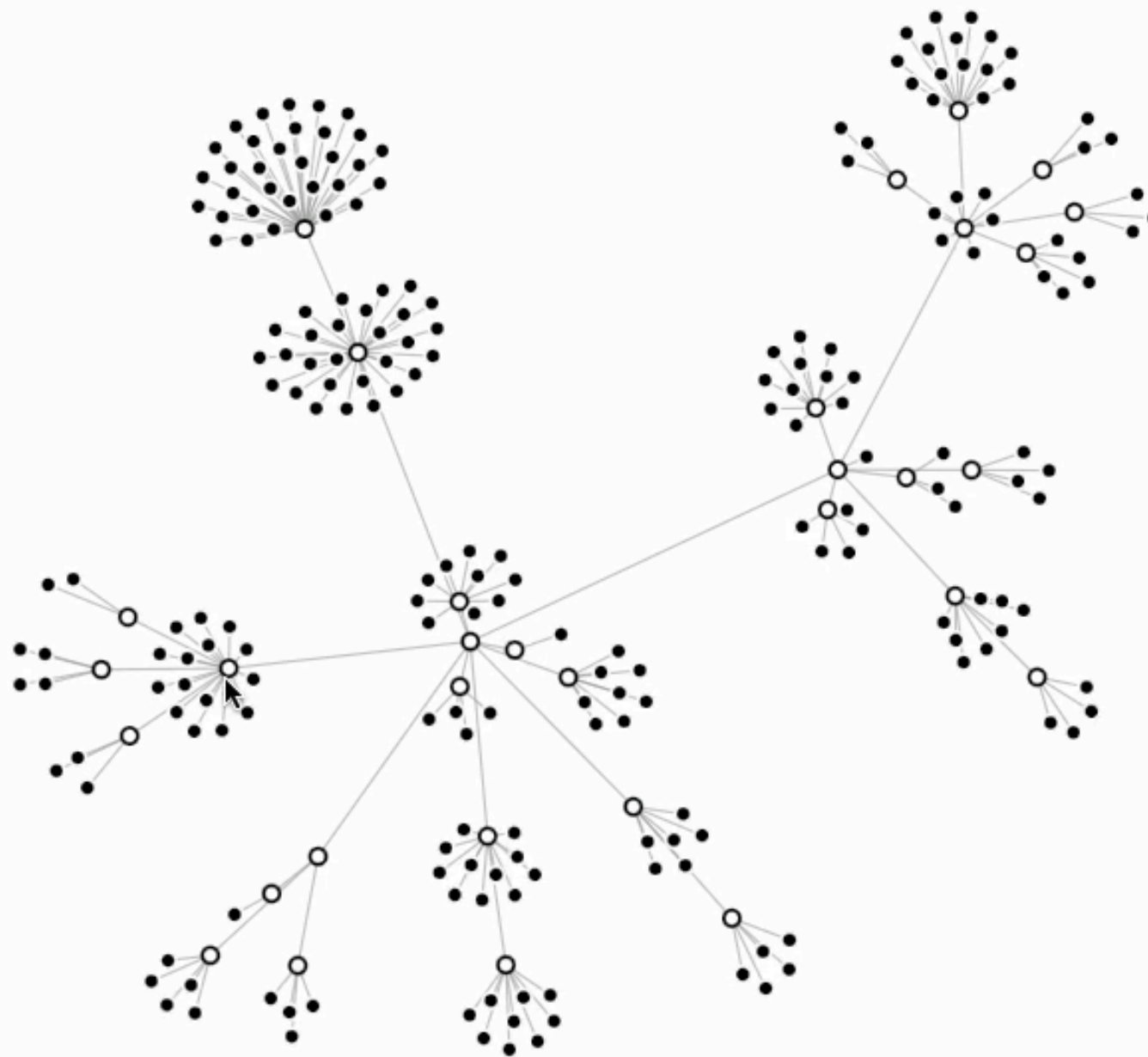
Stack



## Ausblick (Layouts)

---

Force





# Einführung Geomapping

## Was sind GeoJSONs?

- Arbeiten mit Geodaten
- Type: Feature oder FeatureCollection
- Features beinhalten: type, geometry, (properties)
- Geometry:
  - Type: Point, LineString, Polygon, MultiPoint, MultiLineString oder MultiPolygon
  - Coordinates: longitude/latitude-Paar

```
1  {
2      "type": "FeatureCollection",
3      "features": [
4          {
5              "type": "Feature",
6              "id": "01",
7              "properties": {
8                  "name": "Alabama"
9              },
10             "geometry": {
11                 "type": "Polygon",
12                 "coordinates": [
13                     [
14                         [
15                             -87.359296,
16                             35.00118
17                         ],
18                         [
19                             -85.606675,
20                             34.984749
21                         ],
22                         [
23                             -85.431413,
24                             34.124869
25                         ],

```

# Einführung Geomapping

---

## Projection

- Abbilden von 3D Räumen auf 2D Flächen
- verschiedene Projektionen
  - `translate()`: zentrieren
  - `scale()`: skalieren (default ist 1000)

```
var projection = d3.geoAlbersUsa()  
  .translate([w/2, h/2])  
  .scale([500]);
```

## Path

- Definieren eines Path Generator
  - Umwandeln von Geokoordinaten in SVG Path Code
- Projektion wird definiert

```
var path = d3.geoPath()  
  .projection(projection);
```

# Einführung Geomapping

## GeoJSON laden

- json(): laden der GeoJSON
- path: Erstellen von path-Elementen
  - attr("d", path): für jedes Element
  - modifizieren über stroke und fill

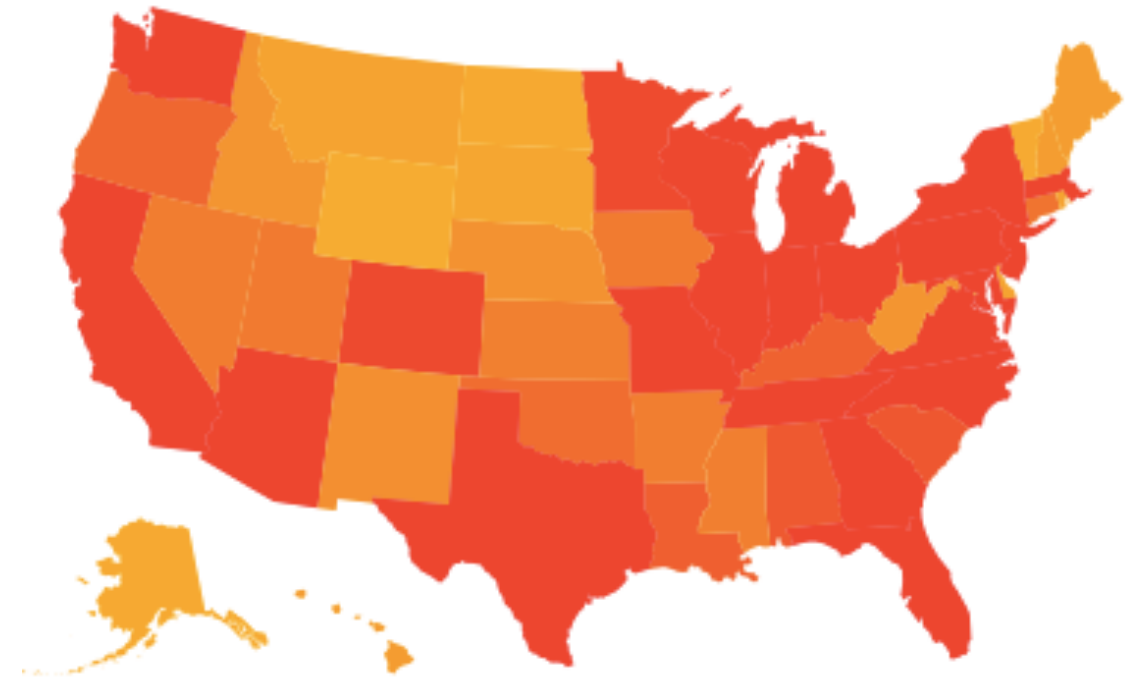
```
d3.json("us-states.json").then(json => {  
  svg.selectAll("path")  
    .data(json.features)  
    .enter()  
    .append("path")  
    .attr("d", path)  
    .attr("stroke", "grey")  
    .attr("fill", "white");  
});
```



# Einführung Geomapping

## Choropleth

- Karten mit Farbflächen nach Wert
- Farbskala
  - scaleLinear()
  - scaleQuantize()



## merge GeoJSON und CSV

- jeder Staatenname in der GeoJSON, der in der csv enthalten ist, wird ein neuer Wert aus der csv zugeschrieben

```
32 d3.json("us-states.json").then(function (json) {
33   for (var i = 0; i < data.length; i++) {
34     //Staatenname aus csv
35     var dataState = data[i].state;
36     //Wert aus csv in float umwandeln
37     var dataValue = parseFloat(data[i].value);
38     //für jedes Objekt in json
39     for (var j = 0; j < json.features.length; j++) {
40       //Staatennamen aus geojson
41       var jsonState = json.features[j].properties.name;
42       //wenn Staat aus json in csv vorhanden
43       if (dataState == jsonState) {
44         //Wert aus csv in json unter properties kopieren
45         json.features[j].properties.value = dataValue;
46         break;
47       }
48     }
49   }
}
```

# Einführung Geomapping

---

## Punkte

- Punkte zur Zentroidendarstellung
- lat/lng-Werte benötigt
- für Position mittels projection() in x/y-Werte umgewandelt

```
.attr("cx", function (d) {  
    return projection([d.lon, d.lat])[0];  
})  
.attr("cy", function (d) {  
    return projection([d.lon, d.lat])[1];  
})
```



## Do it yourself (15 Minuten)

---

Für die Bundesländer liegen Einwohnerzahl, Ranking und Zentroide vor

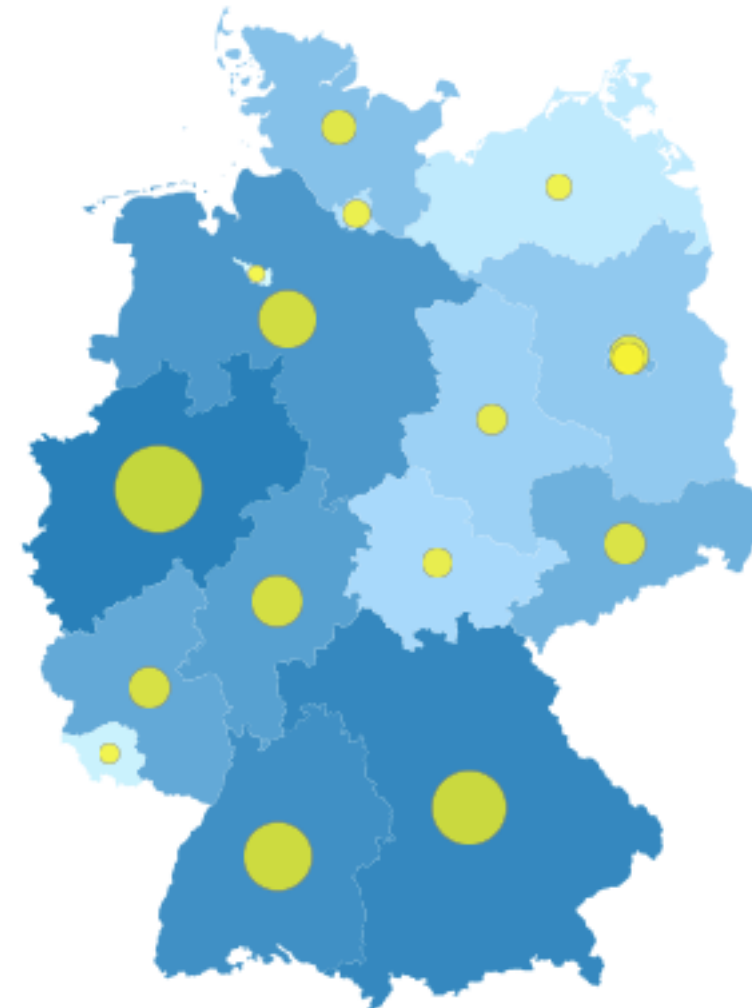
**Erstelle eine Karte, die die Bundesländer eines Rankings entsprechend einfärbt**

**- Füge Bubbles hinzu, die die Größe der Einwohnerzahl des jeweiligen Bundeslandes repräsentiert**

Ausgangskarte



Zielkarte

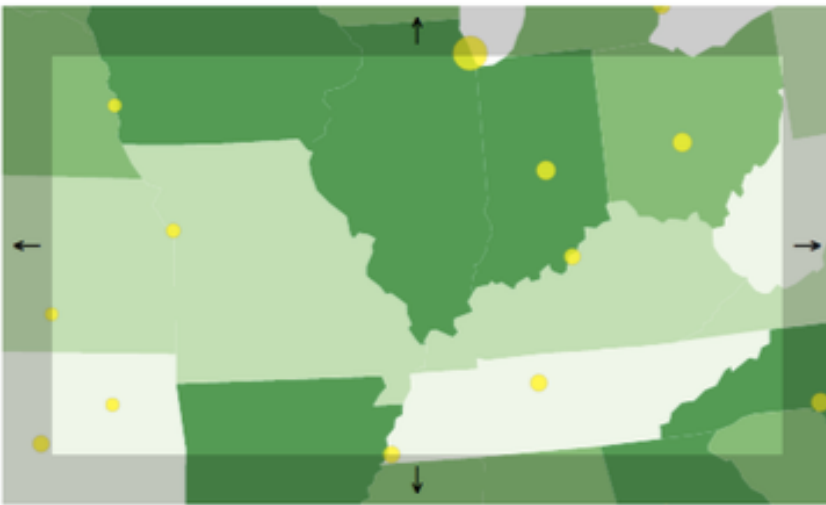


# Ausblick (Geomapping)

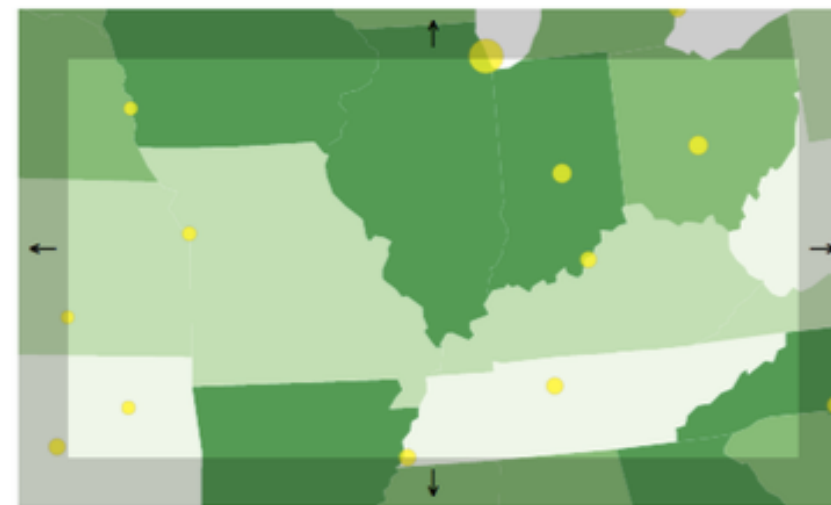
---

## Weitere Funktionen

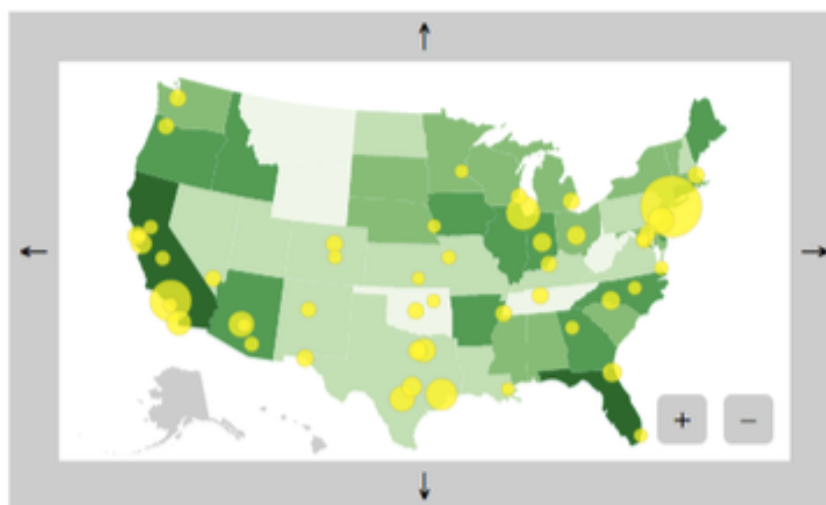
Panning



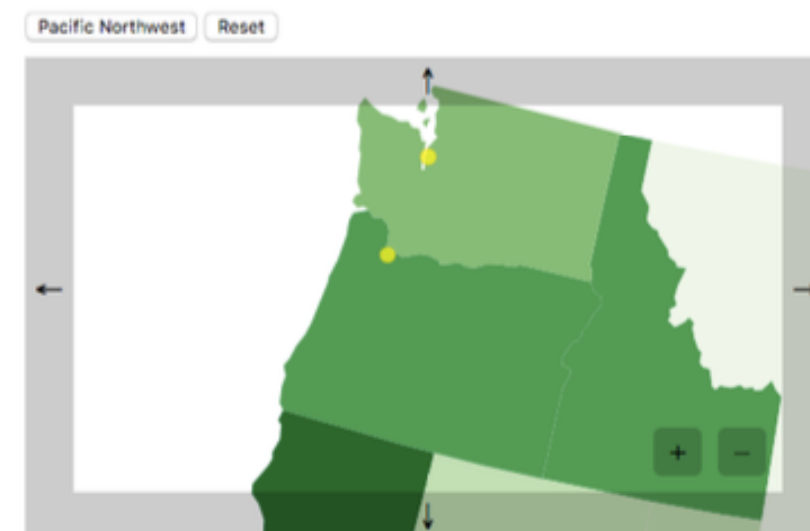
Dragging (d3.drag())



Zooming (d3.zoom())



Preset



**Habt ihr Fragen?**