

# Handout - Layouts

---

## Pie Chart

```
var arc = d3.arc()  
    .innerRadius(innerRadius)  
    .outerRadius(outerRadius);
```

```
var color =  
d3.scaleOrdinal(d3.schemeCategory10);
```

```
var arcs = svg.selectAll("g.arc")  
    .data(pie(dataset))  
    .enter()  
    .append("g")  
    .attr("class", "arc")  
    .attr("transform", "translate(" +  
        outerRadius + ", " + outerRadius +  
        ")");
```

```
arcs.append("path")  
    .attr("fill", color)  
    .attr("d", arc);
```

- `d3.arc()` ist eine built-in Funktion um Bögen (arcs) als SVG Path Elemente zu zeichnen. Die Berechnung für jeden arc wird durch eine Subtraktion des start- und endAngle vorgenommen.
- `.innerRadius()` legt den inneren Radius der Chart fest.
- `.outerRadius()` legt den äußeren Radius der Chart fest.
- `d3.scaleOrdinal()` kategorisiert automatisch übergebene Werte auf einer [Ordinalskala](#).
- `d3.schemeCategory10` ist ein Array mit 10 verschiedenen Farben. Es wird genutzt um schnell Farben im Chart zu haben. Also für Testzwecke gut geeignet.
- Erstellt neue Gruppen für jedes einzelne „Kuchenstück“, bindet diese an die neuen Elemente und jede Gruppe wird in die Mitte des Diagramms „übersetzt“, damit die Daten an der richtigen Stelle erscheinen.
- In jeder Gruppe (g) wird ein Path angehängt. Das Attribut `d` enthält die Beschreibung des Paths. Aus diesem Grund wird der arc Generator (`d3.arc()`) an dieser Stelle aufgerufen.

# Handout - Geomapping

---

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

- für das Arbeiten mit Geodaten wird eine Projection definiert. `D3.geoAlbersUsa()` stellt die USA mit Hawaii und Alaska unten links in der Ansicht dar.
- `.translate()` sorgt für die Plazierung der Karte in der SVG
- `.scale()` skaliert die Karte. Der Wert 1000 ist

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

- `d3.geoPath()` definiert einen Path Generator, der es ermöglicht Geodaten in SVG Path Code umzuwandeln
- dabei wird die Projektion definiert

```
d3.json("shape.json").then(json => {  
  svg.selectAll("path")  
    .data(json.features)  
    .enter()  
    .append("path")  
})
```

- `d3.json()` lädt eine JSON-Datei
- über `.attr("d", path)` wird für jedes Element in der geoJSON ein Path Element erstellt

```
var color = d3.scaleLinear()  
  .range(["red", "blue"]);
```

- `d3.scaleLinear()` definiert eine Skala mit einer Output range, die Farben beinhaltet

```
d3.csv("data.csv").then(function (data) {  
  color.domain([  
    d3.min(data, function (d) { return  
d.value; })),  
    d3.max(data, function (d) { return d.value;
```

- `d3.csv()` lädt eine CSV-Datei
- `color.domain()` definiert die Input Domain einer Skala
- hier wird der kleinste und größte Wert eines Datensatzes genommen

# Handout - Geomapping

---

```
d3.json("shape.json").then(function (json) {  
  for (var i = 0; i < data.length; i++) {  
    var dataState = data[i].name1;  
    var dataValue =  
parseFloat(data[i].value);  
    for (var j = 0; j < json.features.length; j+  
+ ) {  
      var jsonState =  
json.features[j].properties.name2;  
      if (dataState == jsonState) {  
        json.features[j].properties.value =  
dataValue; break;}}});
```

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

```
.attr("r", function(d) {  
  return Math.sqrt(parseInt(d.value) *  
0.000005);  
  })
```

- in dem Beispiel werden die Daten einer GeoJSON mit einer CSV verbunden
- name1 und value aus der CSV extrahiert
- für jedes Objekt aus der GeoJSON wird der name2 extrahiert
- wenn ein name1 aus der CSV einem name2 aus der GeoJSON entspricht, wird der value aus der CSV der GeoJSON hinzugefügt

- über projection() werden lon und lat in x- und y-Werte umgewandelt
- der erste Wert der Ausgabe entspricht cx, der zweite Wert entspricht cy

- in dem Beispiel wird ein Wert in einen Integer umgewandelt und kleiner skaliert
- über Math.sqrt() wird ein Flächenwert in einen Radiuswert umgeformt
- so kann eine Bubble Chart erstellt werden