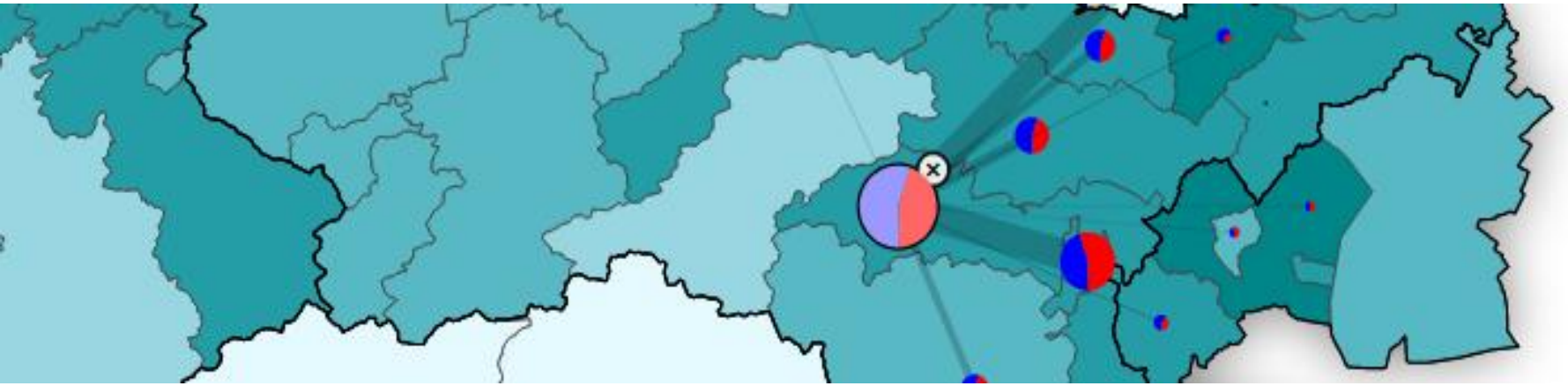
FORSCHUNGSGRUPPE
KARTOGRAPHIE
Research Group Cartography



mapmap.js: Ein kartographisches API für interaktive thematische Karten

Florian Ledermann, Forschungsgruppe Kartographie, TU Wien

florian.ledermann@tuwien.ac.at
@floledermann

Interaktive thematische Karten

- Anwendungsfelder:
  - (Daten-) Journalismus
  - Thematische online-Atlanten
  - Einsatz in der Lehre (Kartographie, Geographie)

- Anforderungen:
  - Vielfältige thematische Visualisierungen
  - Standardtechniken + eigene Erweiterungen
  - Animation & “Storytelling”
  - Komplette kartographische Pipeline am Client

Karten machen mit D3

Mike Bostock "Let's make a map"
http://bost.ocks.org/mike/map/

```
var map = createAMapOfUKCountries();
```

```
var width = 960,
    height = 1160;

var projection = d3.geo.albers().center([0, 55.4]).rotate([4.4, 0]).parallels([50, 60])
    .scale(1200 * 5).translate([width / 2, height / 2]);

var path = d3.geo.path().projection(projection).pointRadius(2);

var svg = d3.select("body").append("svg")
    .attr("width", width)
    .attr("height", height);

d3.json("uk.json", function(error, uk) {
  var subunits = topojson.feature(uk, uk.objects.subunits),
      places = topojson.feature(uk, uk.objects.places);

  svg.selectAll(".subunit")
      .data(subunits.features)
    .enter().append("path")
      .attr("class", function(d) { return "subunit " + d.id; })
      .attr("d", path);

  svg.append("path")
      .datum(topojson.mesh(uk, uk.objects.subunits, function(a, b) {
          return a !== b && a.id !== "IRL";
      }))
      .attr("d", path)
      .attr("class", "subunit-boundary");

  svg.append("path")
      .datum(topojson.mesh(uk, uk.objects.subunits, function(a, b) { return a === b &&
      .attr("d", path)
      .attr("class", "subunit-boundary IRL");

  svg.selectAll(".subunit-label")
      .data(subunits.features)
    .enter().append("text")
      .attr("class", function(d) { return "subunit-label " + d.id; })
      .attr("transform", function(d) { return "translate(" + path.centroid(d) + ")"; })
      .attr("dy", ".35em")
      .text(function(d) { return d.properties.name; });

  svg.append("path")
      .datum(places)
      .attr("d", path)
      .attr("class", "place");

  svg.selectAll(".place-label")
      .data(places.features)
    .enter().append("text")
      .attr("class", "place-label")
      .attr("transform", function(d) { return "translate(" + projection(d.geometry.coor
      .attr("x", function(d) { return d.geometry.coordinates[0] > -1 ? 6 : -6; })
      .attr("dy", ".35em")
      .style("text-anchor", function(d) { return d.geometry.coordinates[0] > -1 ? "star
      .text(function(d) { return d.properties.name; });
});
```

(All CSS omitted)

„Simple things should be simple,
complex things should be possible“
(Alan Kay)

Herausforderungen

- Herausforderungen der technischen Plattform (Browser)
  - DOM Handling
  - Asynchrone Programmierung: Laden, Events
  - Mischen von HTML, CSS & SVG
    - `getScreenCTM()` etc.
  - Responsive Design, mobile Geräte
  - Browser Bugs & Unterschiede
- “Spaghetti Code” für einfache Anwendungen
  - Vermischung kartographischer Aspekte & technischer Details

=> Schwer zu modularisieren für Wiederverwendung

Das mapmap.js API

Ziele:

- Konzeptioniert für interaktive thematische Kartographie
- High level, „simple things should be simple“
  - But „everything“ should be possible!
- Etwas „Magic“
  - Speziell für nicht kartographisch relevante Funktionalität (Laden von Ressourcen & Daten, DOM Manipulation, Events)
- Transparent (Details können angepasst werden)
- Horizontales API, „batteries included“

Basiert auf D3.js & SVG

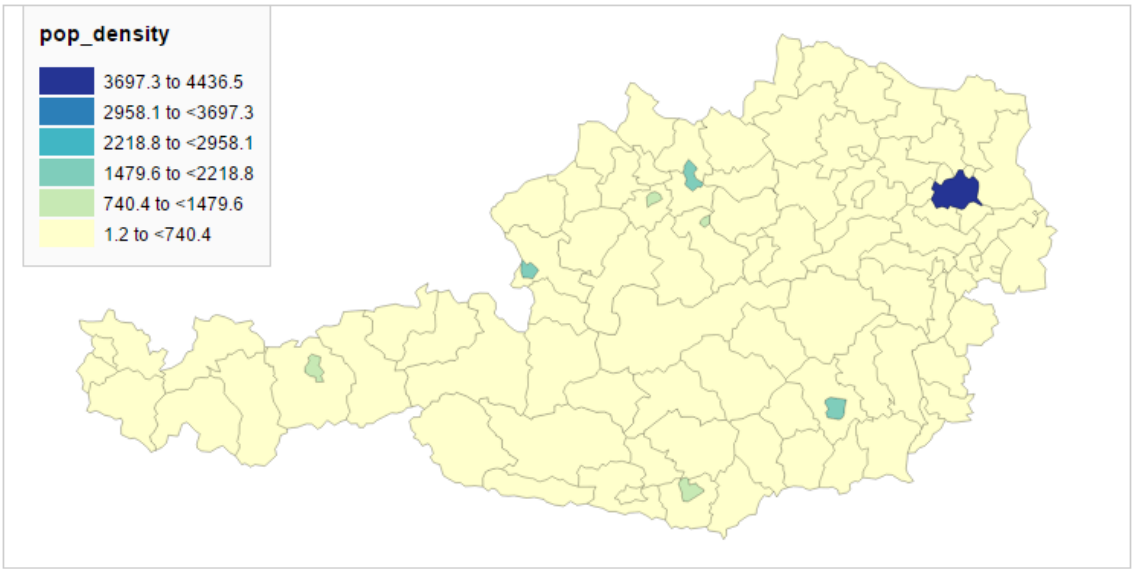
Laden von Geomtrie

```
var map = mapmap('#mapElement')
    .geometry('austria.topojson')
    .projection(d3.geo.conicEqualArea().parallels([46, 49]))
;
```

Laden & „Joinen“ von Daten

```
var map = mapmap('#mapElement')
    .geometry('austria.topojson', 'iso')
    .data('data-AT.csv', 'code')
    .choropleth('pop_density')
    .legend(mapmap.legend.html())
;
```

Data is joined to geometry

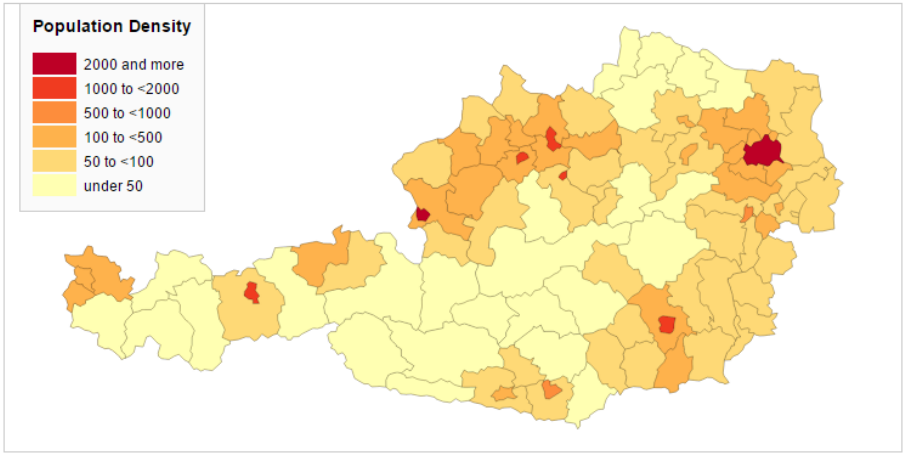


(Non-essential CSS omitted)

Metadaten

```
var map = mapmap('#mapElement')
    .geometry('austria.topojson', 'iso')
    .data('data-AT.csv', 'code')
    .meta({
        'pop_density': {
            label: "Population Density",
            scale: 'threshold',
            domain: [50,100,500,1000,2000],
            color: colorbrewer.YlOrRd[6]
        }
    })
    .choropleth('pop_density')
    .legend(mapmap.legend.html())
;
```

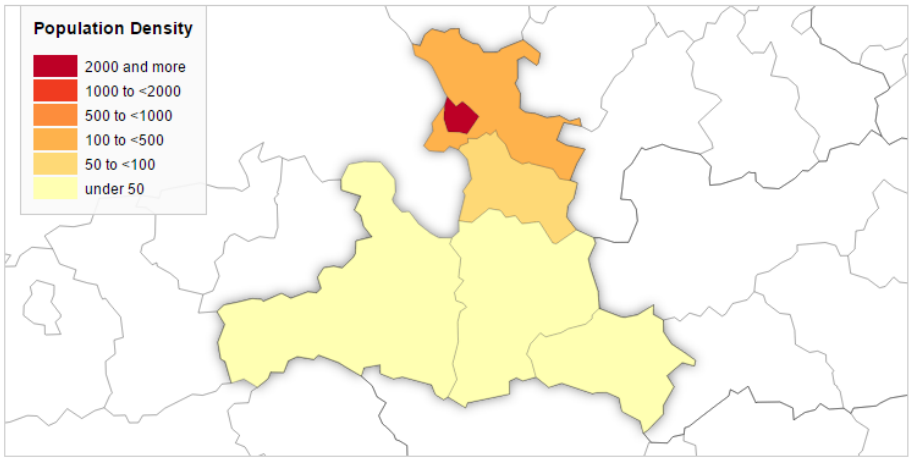
(Non-essential CSS omitted)



Selections

```
function isInSalzburg(d) {
    return d.iso && d.iso.length == 3 && d.iso[0] == '5';
}
var map = mapmap('#mapElement')
    .geometry('austria.topojson', 'iso')
    .data('data-AT.csv', 'code')
    // .meta(...)
    .select(isInSalzburg)
    .extent()
    .highlight()
    .choropleth('pop_density')
    .legend(mapmap.legend.html())
;
```

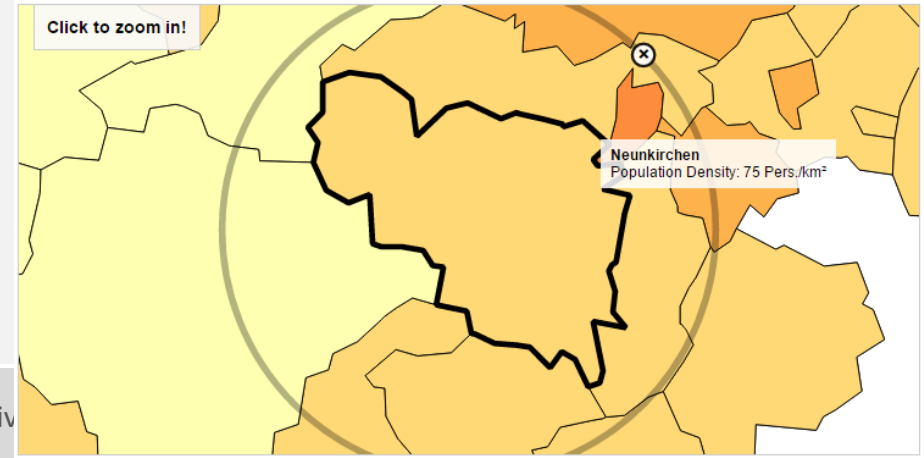
(Non-essential CSS omitted)



Interaktion

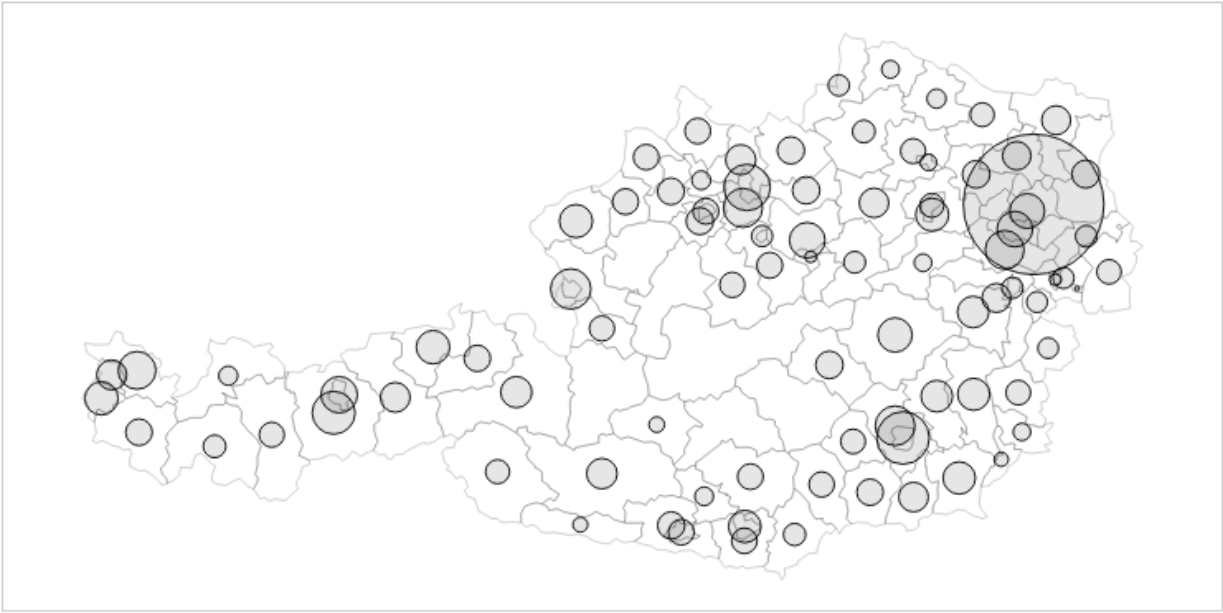
```
var map = mapmap('#mapElement')
    .geometry('../data/austria.topojson', 'iso')
    .data('../data/places-AT.csv', 'code')
    // .meta(...)
    .choropleth('pop_density')
    .hoverInfo(['name','pop_density'])
    .applyBehavior(mapmap.behavior.zoom())
;
```

(Non-essential HTML & CSS omitted)



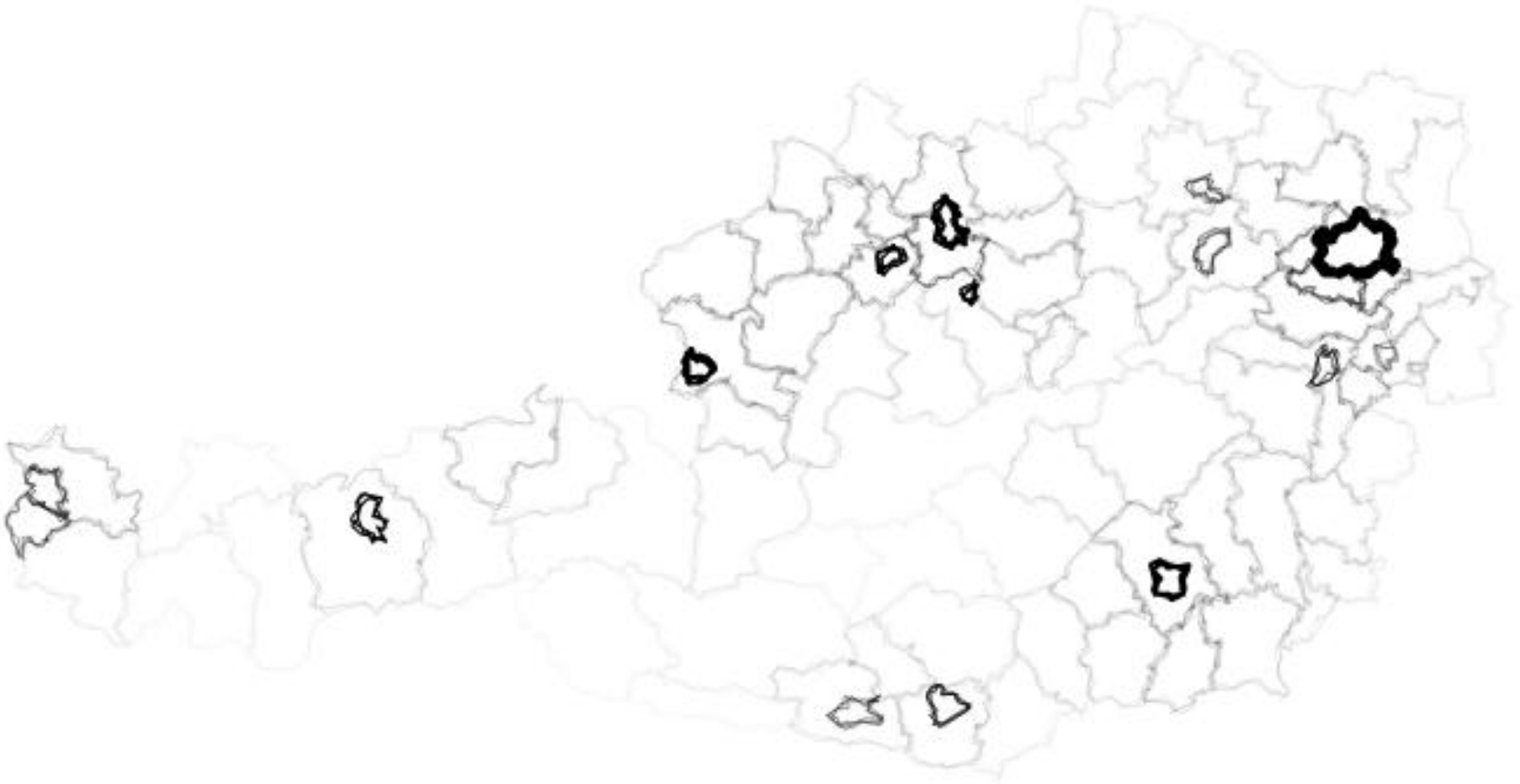
Symbolisierung

Als Attribut
(z.B. Linienfarbe und -Stil)

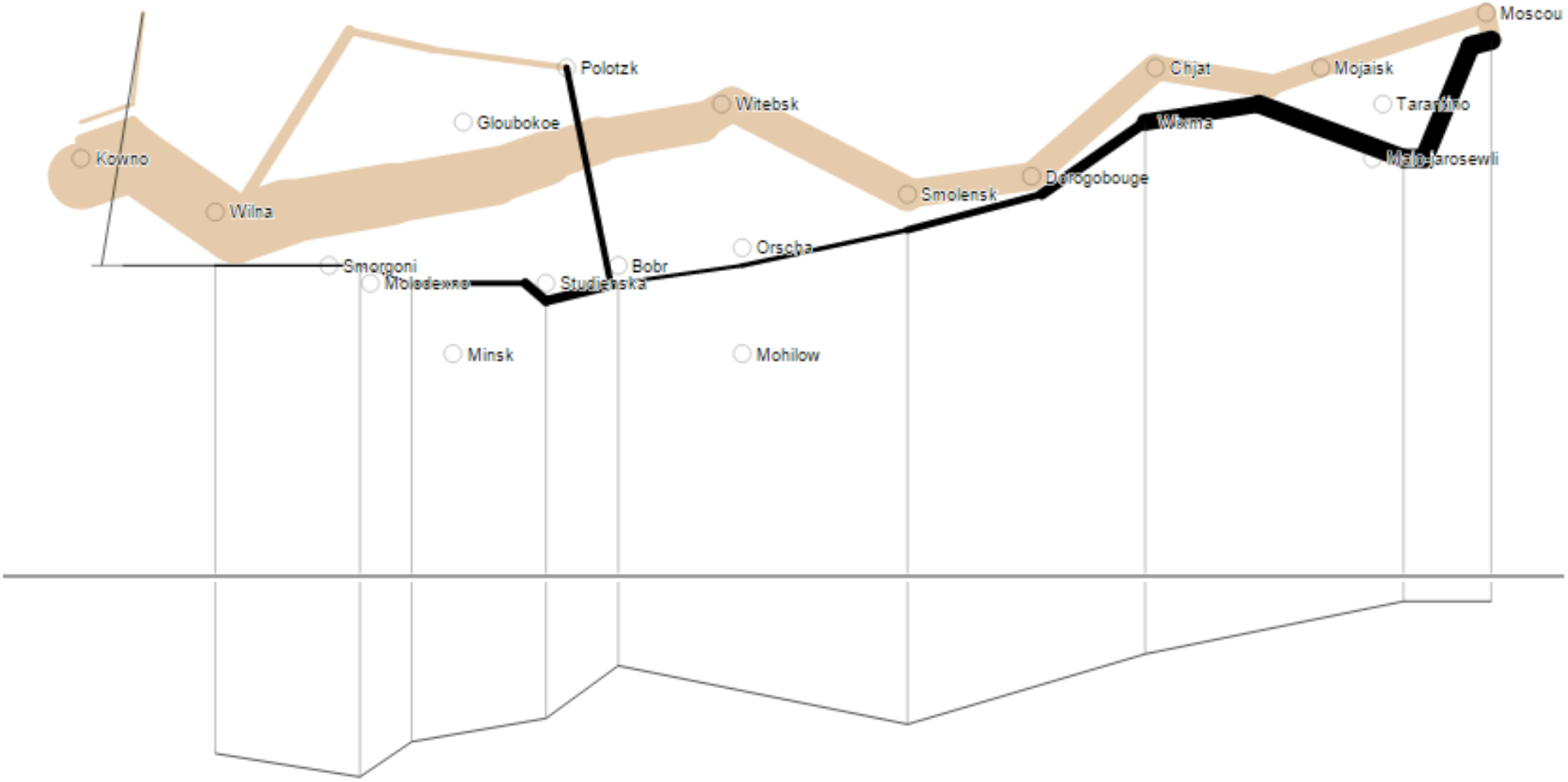


Als SVG-Geometrie

Beispiel: Sketchy Symbolizer



Beispiel: „Minard's Map“



„Minard's Map“: Komplette Implementierung

Map (using mapmap.js)

```
var dd = mapmap.datadata;
var map = mapmap('#mapEl')
    .geometry(napoleon.army, {
        map: dd.map.key('group'),
        reduce: dd.emit.geo.segments()
    })
    .meta({'size': {
        scale: 'linear',
        domain: [0,1000000],
        'stroke-width': [0, 100]
    },'dir': {
        scale: 'ordinal',
        domain: [-1,1],
        'stroke': ['#000000', '#e5cbab'],
        undefinedSymbols: {'stroke': '#000000'}
    }})
    .symbolizeAttribute('size', 'stroke-width')
    .symbolizeAttribute('dir', 'stroke')
    .zOrder('dir')
    .hoverInfo('size')
    .geometry(napoleon.cities, {
        map: dd.map.geo.point('lat','lon')
    })
    .symbolize(mapmap.symbolize.addLabel('name'))
    .anchorFunction(lonAnchors)
;
```

Anchoring Map <-> Diagram

```
function lonAnchors(obj) {
    for (var i=napoleon.army.length - 1; i>=0; i--) {
        var place = napoleon.army[i];
        if (place.lon == obj.lon) {
            return this.project([place.lon, place.lat]);
        }
    }
    return null;
}
```

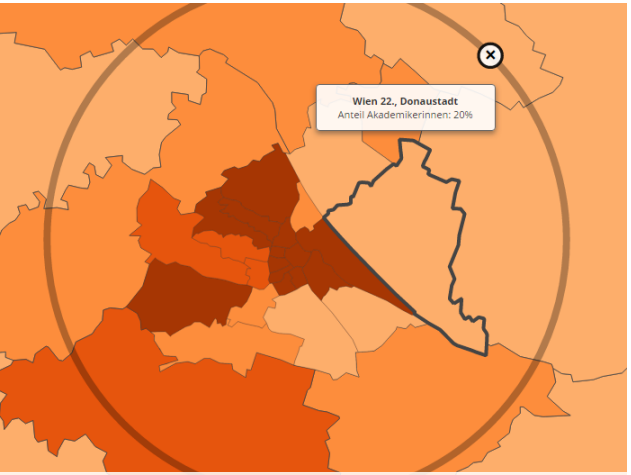
Chart (using D3.js)

```
function createChart(el, data, map) {
    var width = 800,
        height = 100;

    var y = d3.scale.linear().range([height, 10]);
    y.domain(d3.extent(data, function(d) { return d.temp; }));
    var yAxis = d3.svg.axis().scale(y).orient("left");

    el = d3.select('#' + el);
    var path = d3.svg.line()
        .x(function(d){
            return map.anchor(d)[0];
        })
        .y(function(d){ return y(d.temp); });

    el.append('path')
        .datum(data)
        .attr('class', 'temp')
        .attr('d', path);
    el.selectAll('line.anchor')
        .data(data)
        .enter()
        .append('line')
        .attr({
            'class': 'anchor',
            x1:function(d){return map.anchor(d)[0]},
            y1:function(d){ return y(d.temp); },
            x2:function(d){return map.anchor(d)[0]},
            y2:0
        });
    d3.select('#mapEl g.fixed')
        .selectAll('line.anchor')
        .data(data)
        .enter()
        .append('line')
        .attr({
            'class': 'anchor',
            x1:function(d){return map.anchor(d)[0]},
            y1:function(d){ return map.anchor(d)[1]; },
            x2:function(d){return map.anchor(d)[0]},
            y2:400
        });
}
createChart('chartEl', napoleon.temp, map);
```

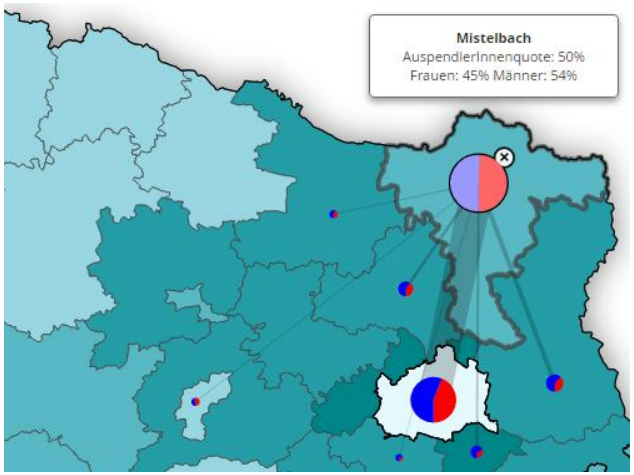
Anwendung: Projekt genderATlas

- genderATlas: Interaktiver online-Atlas für Österreich
  - Projektlaufzeit 2013-2015
  - Heterogene Sammlung thematischer Karten
  - Storytelling, Animation, Interaktivität, linked Diagrams

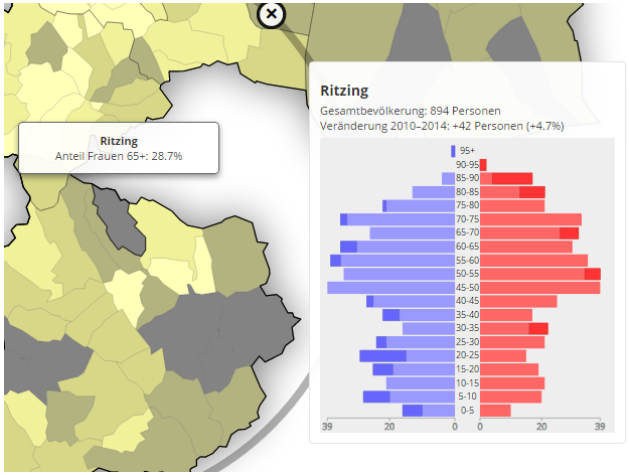
mapmap.js im genderATlas



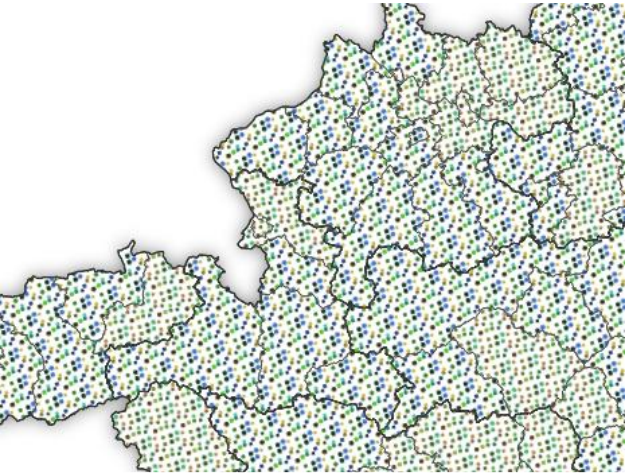
Zoom-and-split



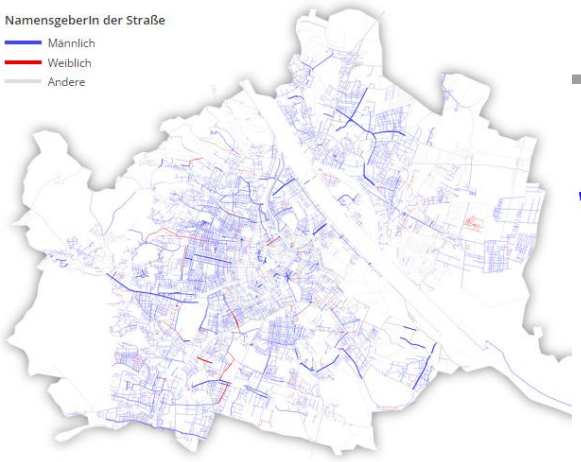
Custom Symbology



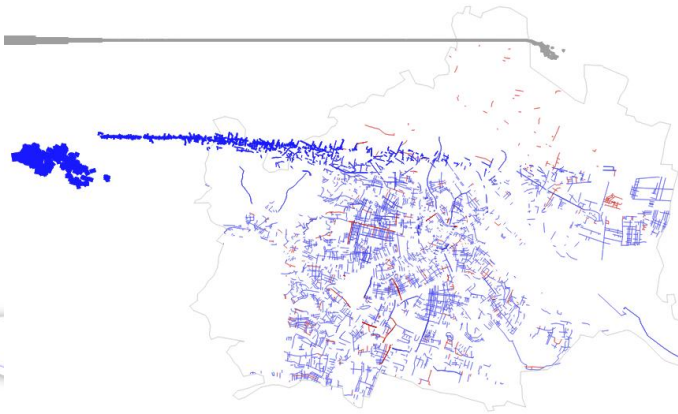
Linked Diagrams



Custom Fill Patterns



Line symbology



Map/Diagram Animation

mapmap.js andernorts

Pilotprojekte von/mit

- wahlatlas.net (Michael Neutze)
- Statistisches Bundesamt Deutschland
- Austria Presseagentur APA

- Einsatz in der Lehre an der TU Wien

Limits

Einschränkungen in der aktuellen Version (0.2.8):

- Monolithisches Map-Objekt, keine “Layers” oder “Selections”
- D3 “Scales” limitiert
- Fix verdrahtetes Rendering als SVG Path
- Legende kann nur ein Attribut abbilden
- Keine Unterstützung von Raster-Daten
- …

mapmap.js 0.3 (Planung)

Neue, modulare Architektur:

- Klarere & modulare kartographische „Pipeline“
  - Streaming Data/Geometrie (z.B. Vector-Tiles)
  - Rasterdaten-Verarbeitung
  - Austauschbare Rendering-Backends (D3+SVG, Canvas, WebGL, …)
- Eigene Implementierung von Classification & Scales
- …

… möglicherweise Parallel zur Weiterentwicklung von 0.2.x

Take a look…

https://github.com/floledermann/mapmap.js

(V 0.2.8, AGPL)

http://floledermann.github.io/mapmap-examples/

florian.ledermann@tuwien.ac.at

@floledermann

Research funded within the framework of FEMtech research projects of Austrian Ministry for Transport, Innovation and Technology (BMVIT).