

Choropleth

Choropleth

1. 데이터 시각화
2. jubing 코로플래스

1. 데이터 시각화

1. 데이터 시각화 툴

- 테이블 형태 <https://datatables.net/>
- 데이터 비주얼 차트 <https://plotdb.com/>

2. 코로플래스

3. 참고 문서

- 서울시 생활 인구 코로플래스맵 시각화 <https://story.pxd.co.kr/1408>
- d3-scale-chromatic <https://github.com/d3/d3-scale-chromatic>

2. jubing 코로플래스

1. 참고 문서

- 한국 맵 <https://github.com/southkorea/southkorea-maps>
- D3로 우리나라 인구밀도 시각화하기 <http://bl.ocks.org/e9t/55699e9fa8c3eb7fe40c>

2. topojson

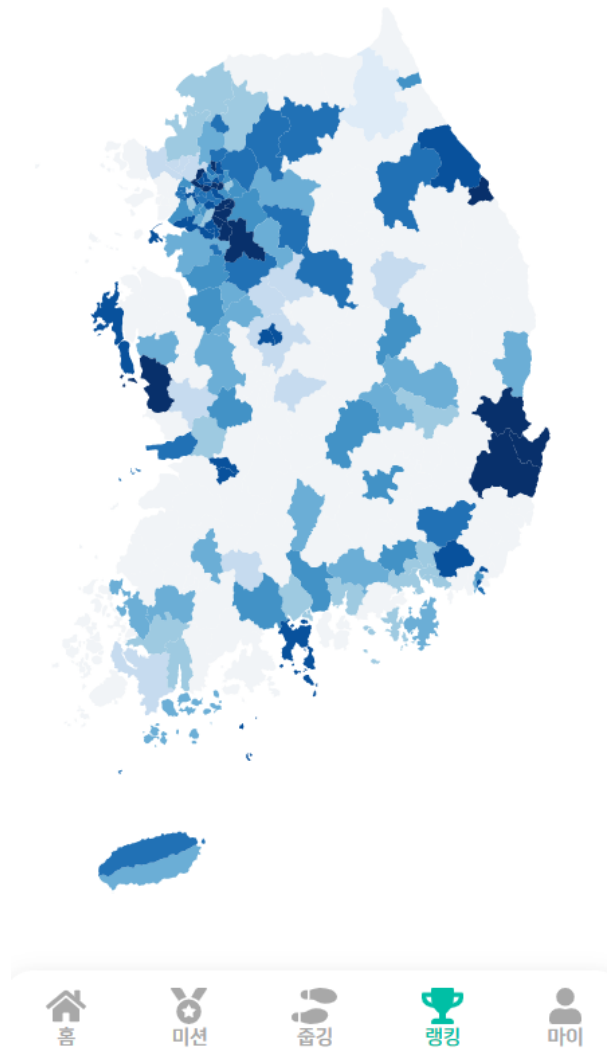
- GeoJSON은 위치정보를 갖는 점을 기반으로 체계적으로 지형을 표현하기 위해 설계된 개방형 공개 표준 형식이다. 이것은 JSON인 자바스크립트 Object Notation을 사용하는 파일 포맷이다. TopoJSON은 GeoJSON의 확장형식으로 각 영역을 아크(arcs)들의 영역으로 구분하여 표시하는 기능을 제공해주어 연산량을 적게 해주는 장점을 가지고 있다.
- 줍깅에서는 행정구역 경계 topojson파일을 사용한다. (크기를 줄이기 위해)

3. jubing 코로플래스

- d3가 v7까지 나왔는데 전반적으로 다들 v3을 사용하는 듯. (v3과 v4~과 꽤 다르다는 것 같다. <https://iros.github.io/d3-v4-whats-new/#1> 사라진 함수나 이름 바뀐 함수도 있고... 레거시 문제나 그런 이유로 v3을 주로 쓰는 듯)
- import (v3)

```
// public/index.html
<script src="https://d3js.org/d3.v3.min.js"></script>
<script src="https://d3js.org/queue.v1.min.js"></script>
<script src="https://d3js.org/topojson.v1.min.js"></script>
```

4. 결과



5. 코드

```
<template>
  <div>
    <div id="chart"></div>
  </div>
</template>

// public/index.html import했는데 이 쪽에도 ? 써줘야하는 듯. 됐다 안됐다해서..
<script src="https://d3js.org/queue.v1.min.js"></script>

<script>
import { HTTP } from '@util/http-common';

export default {
  name: 'RegionRank',
  mounted() {
    // 너비, 높이 설정 & 속성값 부여
    var width = 380,
        height = 600;
    var svg = d3.select("#chart").append("svg")
      .attr("width", width)
      .attr("height", height);
```

```

// 맵 중앙 & 확대 축소 스케일
var projection = d3.geo.mercator()
    .center([127.8, 36])
    .scale(5000)
    .translate([width/2, height/2]);

// 맵 투사
var path = d3.geo.path()
    .projection(projection);

// 0~100까지 9단계로 나누기
var quantize = d3.scale.quantize()
    .domain([0, 100])
    .range(d3.range(9).map(function(i) { return "p" + i; }));

var popByName = d3.map();

// 동기 처리. municipalities-topo-simple.json 이라는 topojson파일을 읽은 후
// data_request 함수 실행 => axios 요청 보내기
// 그 후 ready 함수 실행 => 맵에 데이터 저장
queue()
    .defer(d3.json, "municipalities-topo-simple.json")
    .defer(data_request)
    .await(ready);

// 백의 /map으로 요청 보낸 후 응답 받기 (지역코드 별 줍깅 수 반환)
function data_request(callback) {
    HTTP.get(`/map`)
        .then(res => {
            return res.data.data
        })
        .then(datas => {
            // popByName: {지역코드: 줍깅수, 지역코드: 줍깅수 ...} 형태로 구조화 됨
            datas.forEach((data) => {
                popByName.set(data.code, +data.count)
            })
            // 호출된 곳으로 돌아가서 await(ready) 실행
            callback()
        })
}

function ready(error, data) {
    var features = topojson.feature(data, data.objects["municipalities-geo"]).features;

    features.forEach(function(d) {
        // popByName: {지역코드: 줍깅수, 지역코드: 줍깅수 ...} 형태로 쓰여졌으니
        // popByName.get(d.properties.code)라고 하면 해당 지역코드의 줍깅수가 나
        온다.
        d.properties.jubgingCnt = popByName.get(d.properties.code);
        // 줍깅 수를 밀도에 저장하고 quantize (지정된 도메인을 바탕으로 출력 범위 해
        당 값 반환)
        d.properties.density = d.properties.jubgingCnt;
        d.properties.quantized = quantize(d.properties.density);
    });

    svg.selectAll("path")
        .data(features)

```

```

        .enter().append("path")
        // svg에 클래스로 색 입히기
        .attr("class", function(d) { return "municipality " +
d.properties.quantized; })
        .attr("d", path)
        .attr("id", function(d) { return d.properties.name; })
        .append("title")
        // svg에 마우스 올려놓으면 00구 00중강 이라고 텍스트 뜨게하기
        .text(function(d) {
            return d.properties.name + " " + d.properties.jubgingCnt + "중강"
        });
    }
}
}
</script>

```

// fill로 svg 색 채우는 css 색은 <https://github.com/d3/d3-scale-chromatic> 참고

```

<style>
    svg .municipality { fill: red; }
    svg .municipality: hover { stroke: #333; }
    svg .municipality.p0 { fill: rgb(241, 244, 247); }
    svg .municipality.p1 { fill: rgb(222,235,247); }
    svg .municipality.p2 { fill: rgb(198,219,239); }
    svg .municipality.p3 { fill: rgb(158,202,225); }
    svg .municipality.p4 { fill: rgb(107,174,214); }
    svg .municipality.p5 { fill: rgb(66,146,198); }
    svg .municipality.p6 { fill: rgb(33,113,181); }
    svg .municipality.p7 { fill: rgb(8,81,156); }
    svg .municipality.p8 { fill: rgb(8,48,107); }
    svg text { font-size: 10px; }
</style>

```