hci+d lab.

Lab 10: Map Visualization

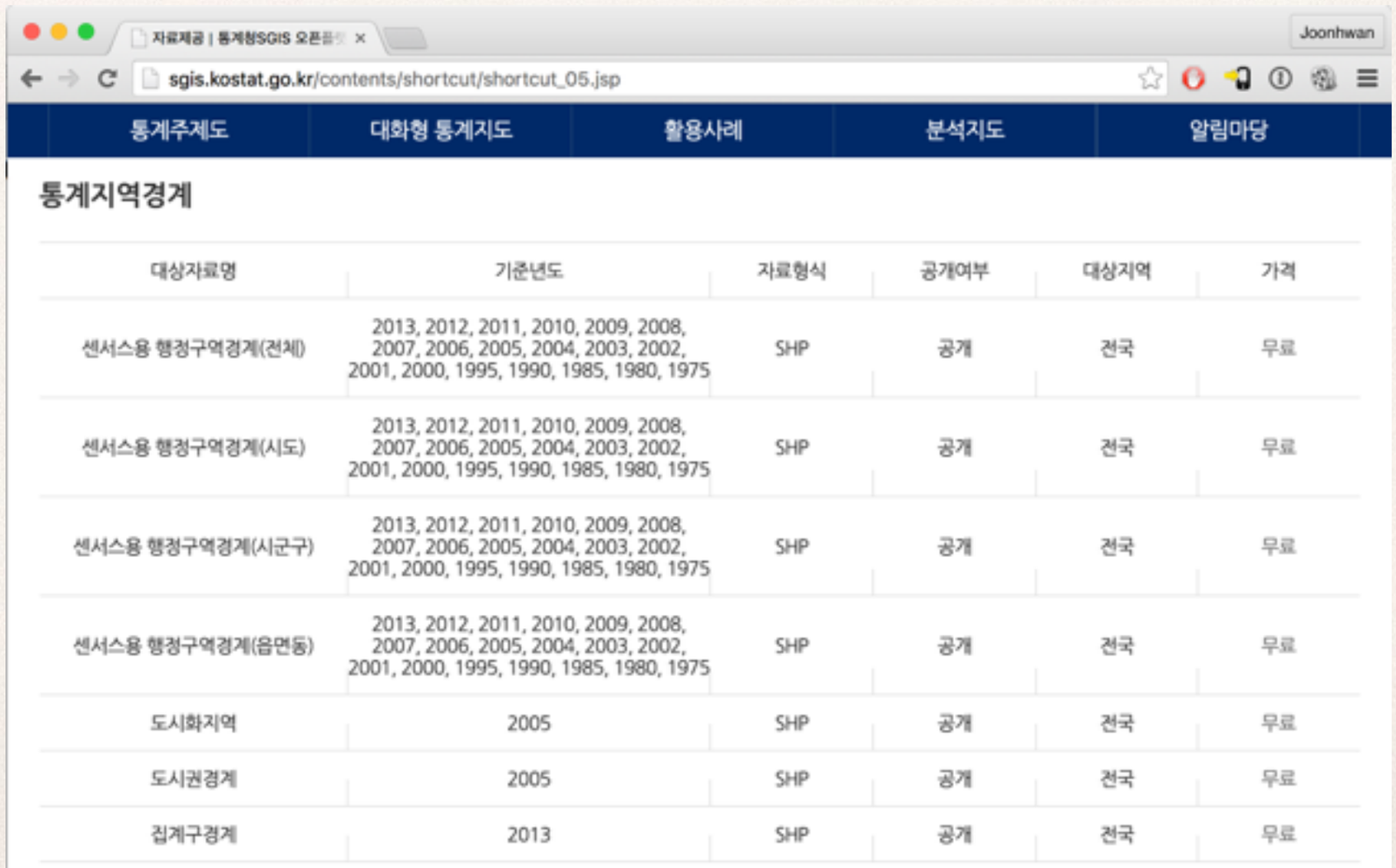
정보 비주얼라이제이션 2015 Fall

human-computer interaction + design lab.
Joonhwan Lee

Map Visualization

지도 파일

- 지도 파일은 일반적으로 Shape 파일의 형태로 제공됨
  - Shape (.shp): ESRI의 지도 데이터 포맷
  - 국내 지도는 통계청에서 shp 형태로 제공
    http://sgis.kostat.go.kr/html/index.html



통계지역경계

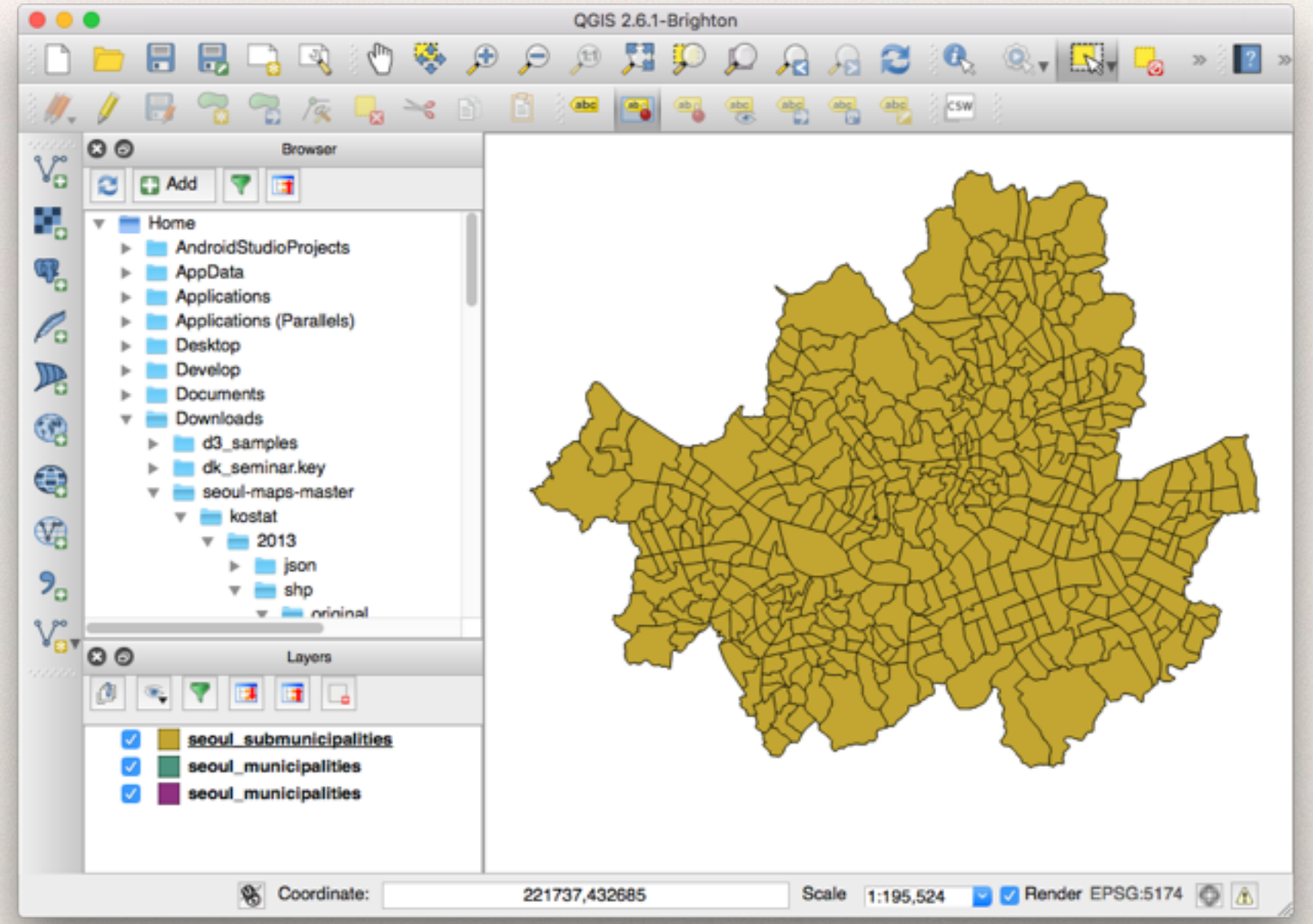
대상자료명	기준년도	자료형식	공개여부	대상지역	가격
센서스용 행정구역경계(전체)	2013, 2012, 2011, 2010, 2009, 2008, 2007, 2006, 2005, 2004, 2003, 2002, 2001, 2000, 1995, 1990, 1985, 1980, 1975	SHP	공개	전국	무료
센서스용 행정구역경계(시도)	2013, 2012, 2011, 2010, 2009, 2008, 2007, 2006, 2005, 2004, 2003, 2002, 2001, 2000, 1995, 1990, 1985, 1980, 1975	SHP	공개	전국	무료
센서스용 행정구역경계(시군구)	2013, 2012, 2011, 2010, 2009, 2008, 2007, 2006, 2005, 2004, 2003, 2002, 2001, 2000, 1995, 1990, 1985, 1980, 1975	SHP	공개	전국	무료
센서스용 행정구역경계(읍면동)	2013, 2012, 2011, 2010, 2009, 2008, 2007, 2006, 2005, 2004, 2003, 2002, 2001, 2000, 1995, 1990, 1985, 1980, 1975	SHP	공개	전국	무료
도시화지역	2005	SHP	공개	전국	무료
도시권경계	2005	SHP	공개	전국	무료
집계구경계	2013	SHP	공개	전국	무료

지도 파일

- d3.js 는 SVG, GeoJSON, TopoJSON 의 형태를 사용
- GeoJSON
  - type, features 두개의 속성을 가짐
  - features: 각 시도에 대한 정보 오브젝트들을 배열을 담고 있음
  - multipolygon을 이용하여 좌표 기록
- TopoJSON
  - GeoJSON과 비슷한 구조
  - arc를 이용하여 좌표 기록
  - 일반적으로 GeoJSON 보다 70% 이상 데이터 크기가 작음

지도 파일

- SHP to TopoJSON (or GeoJSON)
  - QGIS 사용



지도 파일

- Open Source GIS data repositories
  - https://github.com/southkorea
  - https://github.com/southkorea/seoul-maps
  - https://github.com/southkorea/southkorea-maps

지도 파일 (TopoJSON)

Key	Type	Value
Content	Object	Object
type	String	Topology
transform	Object	Object
scale	Array	Array
translate	Array	Array
objects	Object	Object
seoul_municipalities_geo	Object	Object
type	String	GeometryCollection
geometries	Array	Array
Item[0]	Object	Object
arcs	Array	Array
type	String	Polygon
properties	Object	Object
code	String	11250
name	String	강동구
name_eng	String	Gangdong-gu
base_year	String	2013
Item[1]	Object	Object
arcs	Array	Array
type	String	Polygon
properties	Object	Object
code	String	11240
name	String	송파구
name_eng	String	Songpa-gu
base_year	String	2013
Item[2]	Object	Object
Item[3]	Object	Object
Item[4]	Object	Object
Item[5]	Object	Object

Lab1: TopoJSON 이용하여 지도 그리기

- Step1: Setup

index.html

```
<script src="http://d3js.org/
topojson.v1.min.js" charset="utf-8"></script>
```

mycode.js

```
var width = 800, height = 700
var svg = d3.select("#map").append("svg")
    .attr("width", width)
    .attr("height", height)
var seoul = svg.append("g").attr("id",
"seoul")
```

Lab1: TopoJSON 이용하여 지도 그리기

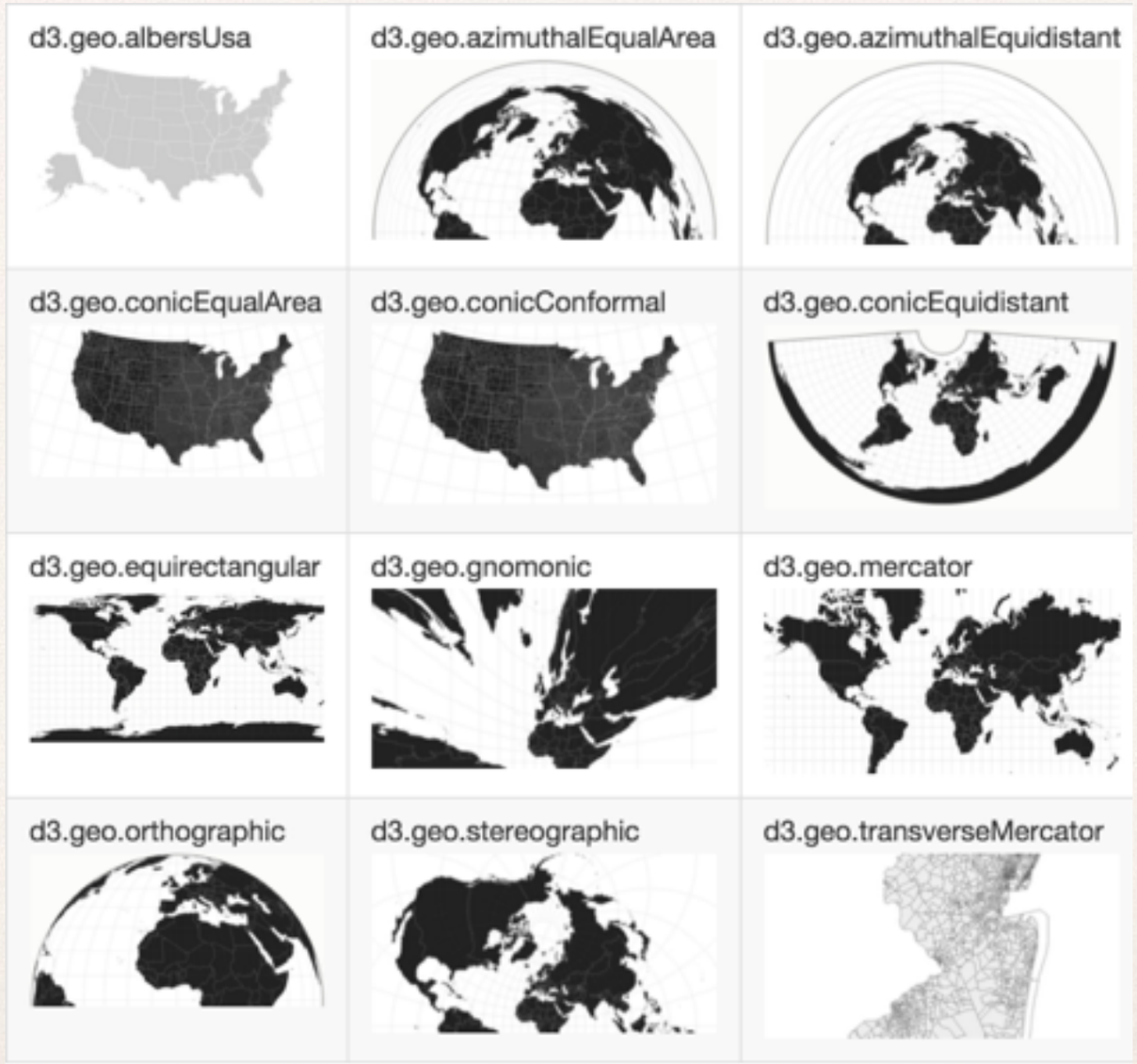
- Step2: Projection

```
var projection = d3.geo.mercator()
    .center([126.9895, 37.5651])
    .scale(100000)
    .translate([width/2, height/2])

var path =
    d3.geo.path().projection(projection)
```

d3.js Projection

- Geo Projections
  - https://github.com/mbostock/d3/wiki/Geo-Projections



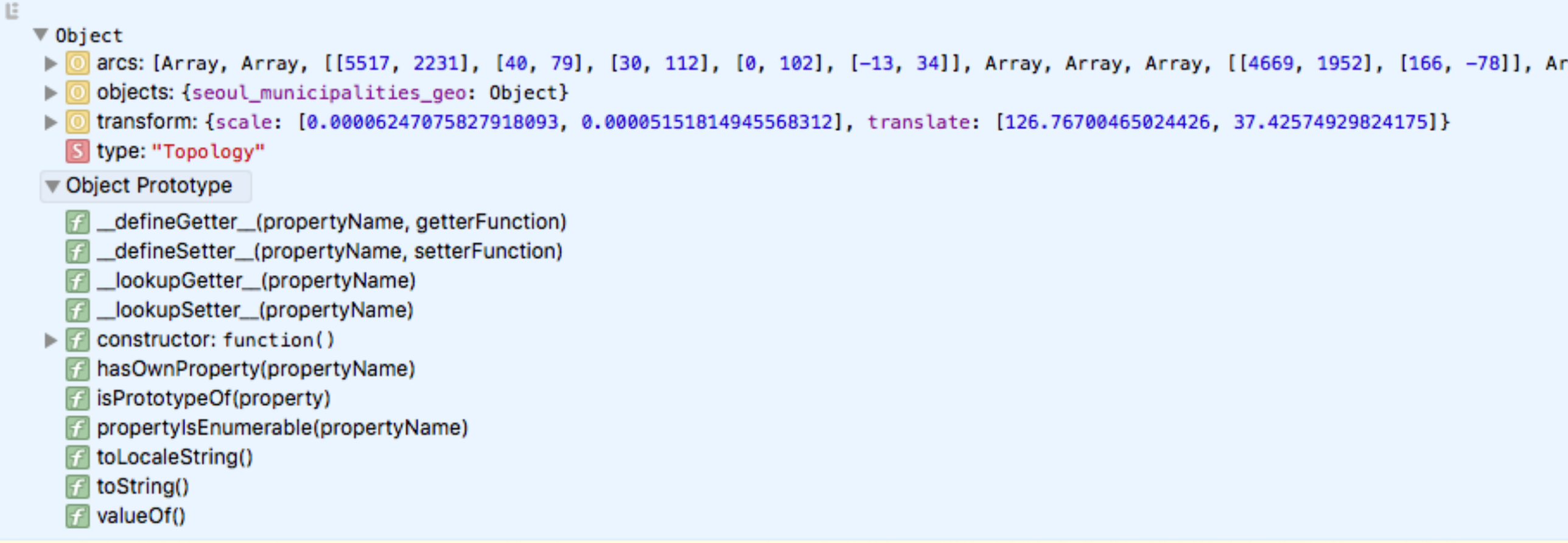
Lab1: TopoJSON 이용하여 지도 그리기

- Step3: Data Loading

```
d3.json("map/
seoul_municipalities_topo_simple.json",
function(error, data) {
 console.log(data)
 var features
    = topojson.feature(data,
      data.objects.seoul_municipalities_geo)
      .features
 console.log(features)
})
```

Lab1: TopoJSON 이용하여 지도 그리기

- `console.log(data)`



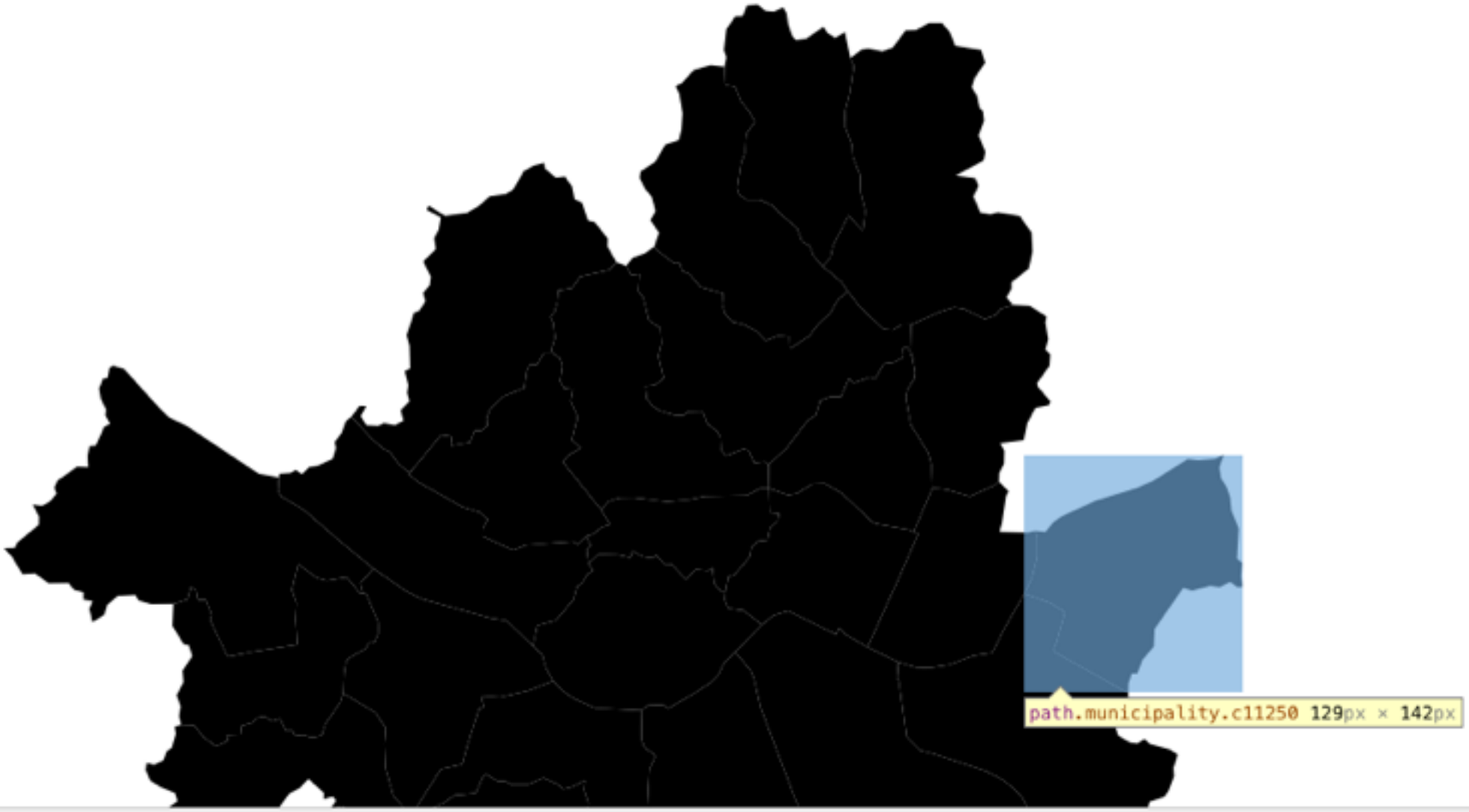
Lab1: TopoJSON 이용하여 지도 그리기

- `console.log(features)`

Lab1: TopoJSON 이용하여 지도 그리기

- Step4: Path 그리기

```
seoul.selectAll("path")
.data(features)
.enter()
.append("path")
.attr("class", function(d) {
    return "municipality c" +
            d.properties.code
})
.attr("d", path)
```


path.municipality.c11250 129px × 142px

13
517.3 KB
1.88s
0
0
1
Elements
Network
Resources
Timelines
Debugger
Storage
html
body
div#geovis
svg
g#map
path.municipality.c11250
<svg width="800" height="600">
<g id="map">
<path class="municipality c11250" d=
"M619.4169587372744,316.6612385391054L625.6317851209315,317.341782664269L630.3201629191462,311.55706761013425L635.3356368428213,308.04076
.73922710142506L678.9484535703959,291.59256252004707L689.0884334595466,287.16828076462843L709.4774252796778,274.3485340259649L715.6922516
141674689134,274.23508046599454L730.8477054761897,271.7390822874586L728.1219044307363,279.34041295699717L729.3212568907475,285.920384844
0272745651L734.6638269398536,297.2645440354536L735.4270512326038,304.97812414293L739.5702688216988,315.3001418192871L738.5889804453473,33
609,335.7154070635006L740.8786533235689,343.6539909357816L741.9689737417502,349.55097606861096L738.8070445289777,349.891180284132l734.59
28.4490005561674,349.891180284132l722.7793343816302,349.09736935068213L712.0941942833306,352.04579060568361706.8606562760542,350.344784
.6107993163896L689.5245616268658,384.9284381432808L682.7645750340598,393.09138521812565L679.820709904976,400.91382867889481676.4407166085
18976414,406.695395200848L674.3691078140109,412.13668362524186L671.9704028939886,411.11645577951276L629.8840347519144,387.64946559847161
62926L637.0801495119522,364.63268559954304L628.793714333704,359.41661200839735L613.2021323535882,353.74677877299837L617.5634140263719,344
6,332.08621222800866L620.834375280916,320.51761792186882"></path> = $0
<path class="municipality c11240" d=
"M538.8422798331012,394.22509576879384L549.2003238058824,397.2860751536937L557.3777269423008,397.6261804516543 6L569.1531874587527,395.698
90.48382124205821594.3395891189284,388.66983926725516L601.3176397953357,374.6107993163896L613.2021323535882,353.74677877299837L628.793714
0801495119522,364.63268559954304L632.1737076300487,382.77425942262926L629.8840347519144,387.64946559847161671.9704028939886,411.116455779
668362524186L674.3691078140109,415.1973291327886L670.7710504339775,417.91785503036226L668.3723455139843,422.338613495609L666.300736719393
5139843,437.413638624391751675.4594282321923,441.72053454564593L679.820709904976,442.2872229706991L684.0729595359007,439.56710070070403L6
3675L699.1193813069258,445.800647572250451703.6987270633108,448.8606658390636L701.081958059629,456.45379829866581701.4090541851183,461.66
,470.16587975413131695.7393880105228,473.79202402125521687.0168246650137,476.62489355262371682.6555429922591,492.82796877114741682.764575
6400690685841,500.87225883608335L669.8987940994266,501.09885285497876L664.6652560921211,507.8965281704295L656.2697888720722,504.4977256 52
3664453983L650.6001226974768,511.63512977257781658.3413976666343,517.0729439605202L653.5439878266188,520.0183515438694L646.6749691920122,
8297,516.5065133751923L640.460142808297,521.9441664319165L635.4446688846801,526.0222881988884L635.4446688846801,521.4910355465836L632.391

Lab1: TopoJSON 이용하여 지도 그리기

- Step5: Label 그리기

```
seoul.selectAll("text")
  .data(features)
  .enter()
  .append("text")
  .attr("class", "municipality-label")
  .attr("transform", function(d) {
     return "translate(" +path.centroid(d)+ ")"
  })
  .attr("dy", ".35em")
  .text(function(d) {
     return d.properties.name
  })
```

Lab1: TopoJSON 이용하여 지도 그리기

- Step6: Styling

```
.municipality {
  fill: silver;
  stroke: #fff;
}

.municipality-label {
  fill: white;
  text-anchor: middle;
  font: bold 10px "Helvetica Neue", Arial,
    Helvetica, Geneva, sans-serif;
}
```

Lab 2: Zoomable Map

- 영역을 클릭하면 Zoom 이 되고 하이라이트된 해당 지역의 맛집 정보를 보여주는 지도

Lab 2: Zoomable Map

- Idea
  - 해당 path가 center 인지 아닌지 판단 (switch)
    - 클릭 했을 때 해당 path를 centered로 지정
    - 다시 클릭하면 centered로 지정 취소
  - transform의 scale 기능을 이용
    - centered로 지정되면 scale 변화

Lab 2: Zoomable Map

- Step1: path 생성 코드에 마우스 이벤트 추가

```
seoul.selectAll("path")
     .data(features)
     ...
     .attr("d", path)
     .on("click", clicked)
```

Lab 2: Zoomable Map

- Step2: `clicked(d)` method

```
function clicked(d) {
    var x, y, k
    if (d && centered != d) {
      var centroid = path.centroid(d)
      x = centroid[0]
      y = centroid[1]
      k = 4
      centered = d
    } else {
      x = width / 2
      y = height / 2
      k = 1
      centered = null
    }
```

Lab 2: Zoomable Map

- Step2: `clicked(d)` method

```
function clicked(d) {
    var x, y, k
    if (d && centered != d) {
      var centroid = path.centroid(d)
      x = centroid[0]
      y = centroid[1]
      k = 4
      centered = d
    } else {
      x = width / 2
      y = height / 2
      k = 1
      centered = null
    }
```

현재 path를
centered 로 지정

Lab 2: Zoomable Map

- Step2: `clicked(d)` method

```
function clicked(d) {
    var x, y, k
    if (d && centered != d) {
      var centroid = path.centroid(d)
      x = centroid[0]
      y = centroid[1]
      k = 4
      centered = d
    } else {
      x = width / 2
      y = height / 2
      k = 1
      centered = null
    }
```

현재 path를
centered 로 지정

centered 해제

Lab 2: Zoomable Map

- Step2: `clicked(d)` method

```
function clicked(d) {

seoul.selectAll("path")
 .classed("active", centered && function(d){
   return d == centered
 })

seoul.transition()
 .duration(750)
 .attr("transform", "translate(" + width/2
       + "," + height / 2 + ")scale(" + k +
       ")translate(" + -x + "," + -y + ")")
```

Lab 2: Zoomable Map

- Ste

fur

현재 path의 클래스를
active 로 지정

```
    seoul.selectAll("path")
      .classed("active", centered && function(d){
        return d == centered
      })

    seoul.transition()
      .duration(750)
      .attr("transform", "translate(" + width/2
            + "," + height / 2 + ")scale(" + k +
            ")translate(" + -x + "," + -y + ")")
```

Lab 2: Zoomable Map

- Ste

```
fur
    seoul.selectAll("path")
     .classed("active", centered && function(d){
       return d == centered
     })

    seoul.transition()
     .duration(750)
     .attr("transform", "tra                    + width/2
           + "," + height / 2 + ")scale(" + k +
           ")translate(" + -x + "," + -y + ")")
```

현재 path의 클래스를
active 로 지정

zoom to path
(x, y) 값은 위에서 계산

Lab 2: Zoomable Map

- Step3: Places

```
d3.csv("data/places.csv", function(error,
data) {
    places.selectAll("circle")
        ...
        .attr("cx", function(d) {
            return projection([d.lon, d.lat])[0]
        })
        .attr("cy", function(d) {
            return projection([d.lon, d.lat])[1]
        })
...
```

Lab 2: Zoomable Map

- Step4: Places의 zoom

```
places.transition()
  .duration(750)
  .attr("transform", "translate(" + width / 2
        + "," + height / 2 + ")scale(" + k +
        ")translate(" + -x + "," + -y + ")")
```

Lab 2: Zoomable Map

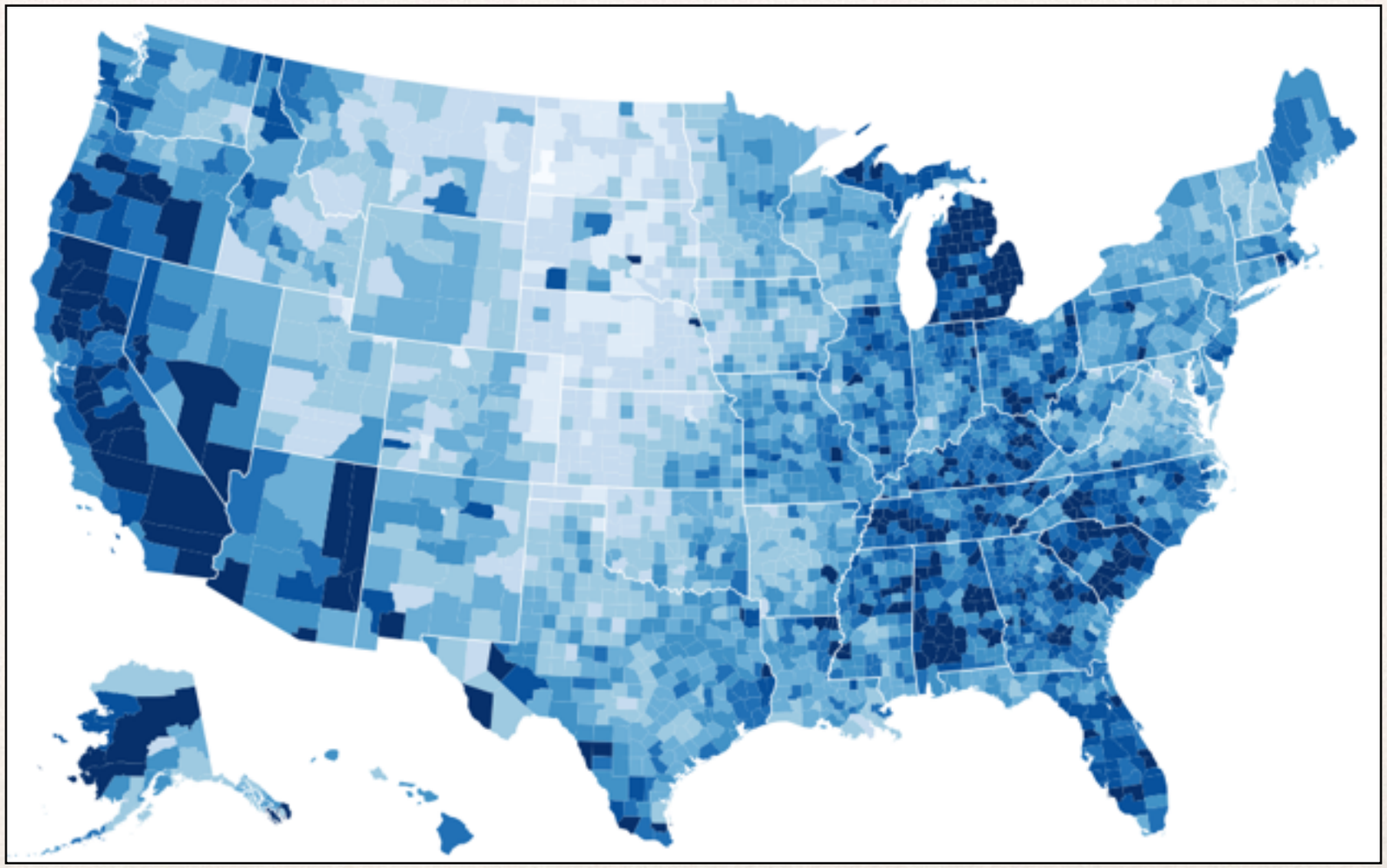
- Step5: Places 숨기기

```
places.selectAll("circle")
    .classed("show", centered && k == 4)

places.selectAll("text")
    .classed("show", centered && k == 4)
```

Lab3: Choropleth Map

- Choropleth Map
  - http://bl.ocks.org/mbostock/4060606
  - Combine two data → use d3 queue library
    https://github.com/mbostock/queue



Lab3: Choropleth Map

- Step1: Combine data

```
var popByName = d3.map()

queue()
    .defer(d3.json, "map/
      seoul_municipalities_topo_simple.json")
    .defer(d3.csv, "data/fire.csv",
      function(d) {
          popByName.set(d.name, +d.value)
      })
    .await(ready)
```

Lab3: Choropleth Map

- Step2: color scale 지정

```
var colorScale = d3.scale.quantize()
      .domain([0, 500])
      .range(d3.range(9).map(function(i) {
            return "p" + i
      }))
```

*range의 값을 0 부터 p8 까지 생성

Lab3: Choropleth Map

- Step3: path 그리기

```
seoul.selectAll("path")
    .data(features)
    .enter().append("path")
    .attr("class", function(d) {
      return "municipality " +
        colorScale(popByName.get(d.properties
        .name))
    })
    .attr("d", path)
    .attr("id", function(d) {
      return d.properties.name
    })
```

Lab3: Choropleth Map

- Step3: path 그리기

```
seoul.selectAll("path")
    .data(features)
  .enter().append("path")
  .attr("class", function(d
    return "municipality " +
      colorScale(popByName.get(d.properties
      .name))
  })
  .attr("d", path)
  .attr("id", function(d) {
    return d.properties.name
  })
```

class 이름에 colorScale을 사용한 값을 할당 (p0-p8)

Lab3: Choropleth Map

- Step4: Styling

```
.municipality.p0 {
 fill: #ffffd9;
}

.municipality.p1 {
 fill: #edf8b1;
}

...
```

Lab4: Google Map + d3.js

- 구글맵에서 제공하는 js api 를 이용하여 d3와 함께 사용 가능
- google api: 지도 데이터 제공
- d3.js: 사용자 데이터의 overlay

Lab4: Google Map + d3.js

- Step1: Google Map API 설치
  - index.html 에 다음의 내용 추가

```
<script type="text/javascript"
src="http://maps.google.com/maps/api/js?
sensor=true"></script>
```

Lab4: Google Map + d3.js

- Step2: 지도 그리기

```
var map = new google.maps.Map(
    d3.select("#map").node(), {
        zoom: 15,
        center: new google.maps.LatLng(37.463842,
                126.949335),
        mapTypeId: google.maps.MapTypeId.ROADMAP
    }
)
```

Lab4: Google Map + d3.js

- Step3: Overlay map 그리기
  - `var overlay = new google.maps.OverlayView()`
    - overlay 생성
  - `overlay.onAdd = function() {…}`
    - overlay의 초기화
  - `overlay.draw = function() {...}`
    - 화면이 업데이트될 때마다 호출됨
  - `function transform(d) {...}`
    - overlay.draw 에서 화면에 요소를 그릴 때 마다 호출되어 각 요소들의 좌표 계산
  - `overlay.setMap(map)`
    - map 위에 overlay 데이터를 설정

Assignment 7: Map Visualization

- 제출: 11/23 (자정)

Assignment 7: Map Visualization

- 데이터: 자유롭게 선택
- Map Visualization을 만들어 보자
- 제출: 다음주 일요일 (11/23) 자정
- 파일 제출 방법
  - 본인의 학번 뒷자리로 폴더를 만들고
  - html 파일은 반드시 index.html 로 이름을 부여한 후, 관련 파일을 모두 같은 폴더에 넣고 (주의: 한글 파일 이름 x)
  - zip 으로 압축해서 제출

Questions...?