

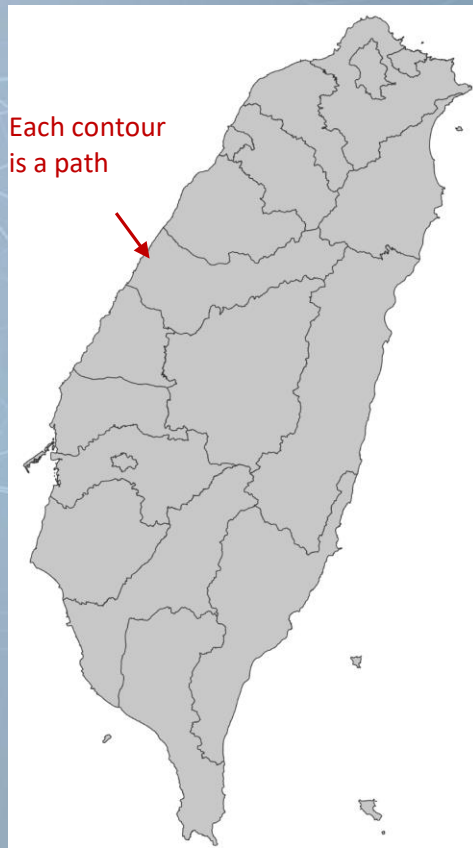


Map

Data Visualization

Three Key Concept of Map in D3

- GeoJSON
 - A json-based format for specifying geographic data
 - D3 creates map based on GeoJSON data
- Map projections
 - Functions that convert from latitude/longitude coordinate to x and y coordinates
- Geographic path generators – **d3.geoPath()**
 - Functions that convert GeoJSON shapes into SVG “paths”
 - Similar to shape generator e.g. d3.line(), d3.area(), etc.
 - Note: d3.geoPath() only recognize **WGS84** geodetic system. Make sure the file you import uses WGS84



GeoJSON

- A JSON-based format for specifying geographic data
- A segment of GeoJSON data for Taipei City in “taiwan.json”
 - You can find “taiwan.json” in Ex06-01 folder

Properties: name, id, and other attributes of the region

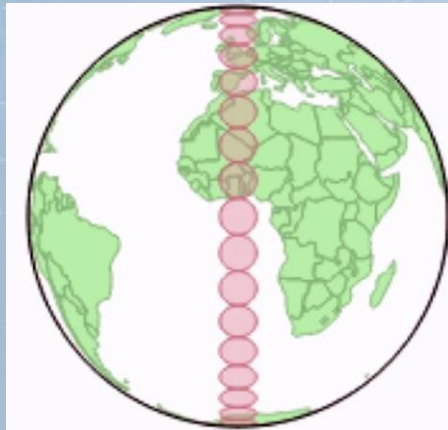
```
{ "type": "Feature", "properties": { "COUNTYID": "63", "NAME_1984": "臺北市", "NAME_1984_ALIAS": "台北市", "NAME_2010": "臺北市", "NAME_2010_ALIAS": "台北市", "NAME_2014": "臺北市", "NAME_2014_ALIAS": "台北市", "ISO3166": "TPE", "SEGIS_COUNTY_ID": "63000", "AREA_ID": "A", "id": "18744641", "geometry": { "type": "Polygon", "coordinates": [[[ [121.570980505, 25.197168115], [121.570972325, 25.197035112], [121.570933739, 25.196892124], [121.570916396, 25.196824628], [121.570896707, 25.196769777], [121.570873123, 25.196706945], [121.570838015, 25.196631625], [121.570810569, 25.196563093], [121.570794274, 25.196480668], [121.570809417, 25.196392146], [121.570871837, 25.196258319], [121.57093985, 25.196163419], [121.571050807, 25.196073786], [121.571103112, 25.196027289], [121.571178172, 25.19598511], [121.571279499, 25.195927707], [121.571360862, 25.19588545], [121.571414521, 25.195845818], [121.571479026, 25.195770484], [121.571527208, 25.195699925], [121.571567938, 25.195637512], [121.571599922, 25.195580893], [121.571661955, 25.195509035], [121.571694973, 25.195436367], [121.5717422, 25.195387625], [121.571887893, 25.195346915], [121.572022694, 25.195338467], [121.572156089, 25.195319697], [121.572267904, 25.195290863], [121.572404217, 25.195211251], [121.572506625, 25.195141223], [121.572623032, 25.195080196], [121.572711925, 25.195035561], [121.572904134, 25.194985127], [121.573071745, 25.194978591], [121.573180199, 25.194979625], [121.573264975, 25.19500044], [121.573344693, 25.195020172], [121.57342179, 25.195033045], [121.573534334, 25.195055829], [121.57359519, 25.195079218], [121.573724591, 25.195135087], [121.573824513, 25.195156862], [121.573918002, 25.195169545], [121.574003551, 25.195155933], [121.574094063, 25.195136515], [121.574160911, 25.195138034], [121.574246998, 25.195162264], [121.574470178, 25.195251468], [121.574590627, 25.195298253], [121.574660911, 25.195338467], [121.574746998, 25.195369992], [121.574824513, 25.19540044], [121.574904063, 25.195436367], [121.574978591, 25.195469992], [121.575055829, 25.19550044], [121.575135087, 25.195530893], [121.575211251, 25.195561428], [121.575290413, 25.195592146], [121.575369992, 25.195622694], [121.575446998, 25.195653093], [121.575524513, 25.19568319], [121.575603551, 25.195713786], [121.575682591, 25.195744274], [121.575761955, 25.195774707], [121.575841428, 25.19580511], [121.575920893, 25.19583545], [121.57600044, 25.195865818], [121.576079922, 25.19589625], [121.576159367, 25.195926694], [121.576238829, 25.195957124], [121.576318396, 25.195987591], [121.576397938, 25.196018002], [121.576477422, 25.196048417], [121.576556915, 25.196078829], [121.576636367, 25.196109242], [121.576715829, 25.196139657], [121.576795289, 25.196170071], [121.576874751, 25.196200484], [121.576954213, 25.196230893], [121.577033675, 25.196261302], [121.577113127, 25.196291715], [121.577192591, 25.196322124], [121.577272054, 25.196352531], [121.577351478, 25.196382938], [121.577430941, 25.196413345], [121.577510354, 25.196443752], [121.577589767, 25.196474159], [121.577669229, 25.196504566], [121.577748692, 25.196534973], [121.577828154, 25.19656538], [121.577907617, 25.196595787], [121.577987079, 25.196626194], [121.578066542, 25.196656601], [121.578146004, 25.196687008], [121.578225467, 25.196717415], [121.578304929, 25.196747822], [121.578384392, 25.196778229], [121.578463854, 25.196808636], [121.578543317, 25.196839043], [121.578622779, 25.19686945], [121.578702242, 25.196900008], [121.578781704, 25.196930415], [121.578861167, 25.196960822], [121.578940629, 25.196991229], [121.579020092, 25.197021636], [121.579099554, 25.197052043], [121.579179017, 25.19708245], [121.579258479, 25.197112857], [121.579337942, 25.197143264], [121.579417404, 25.197173671], [121.579496867, 25.197204078], [121.579576329, 25.197234485], [121.579655792, 25.197264892], [121.579735254, 25.197295299], [121.579814717, 25.197325706], [121.579894179, 25.197356113], [121.579973642, 25.19738652], [121.580053104, 25.197416927], [121.580132567, 25.197447334], [121.580212029, 25.197477741], [121.580291492, 25.197508148], [121.580370954, 25.197538555], [121.580450417, 25.197568962], [121.580529879, 25.197599369], [121.580609342, 25.197629776], [121.580688804, 25.197660183], [121.580768267, 25.19769059], [121.580847729, 25.197721007], [121.580927192, 25.197751414], [121.581006654, 25.197781821], [121.581086117, 25.197812228], [121.581165579, 25.197842635], [121.581245042, 25.197873042], [121.581324504, 25.197903449], [121.581403967, 25.197933856], [121.581483429, 25.197964263], [121.581562892, 25.19799467], [121.581642354, 25.198025077], [121.581721817, 25.198055484], [121.581801279, 25.198085891], [121.581880742, 25.198116298], [121.581960204, 25.198146705], [121.582039667, 25.198177112], [121.582119129, 25.198207519], [121.582198592, 25.198237926], [121.582278054, 25.198268333], [121.582357517, 25.19829874], [121.582437079, 25.198329147], [121.582516542, 25.198359554], [121.582596004, 25.198389961], [121.582675467, 25.198420368], [121.582754929, 25.198450775], [121.582834392, 25.198481182], [121.582913854, 25.198511589], [121.582993317, 25.198541996], [121.583072779, 25.198572403], [121.583152242, 25.19860281], [121.583231704, 25.198633217], [121.583311167, 25.198663624], [121.583390629, 25.198694031], [121.583470092, 25.198724438], [121.583549554, 25.198754845], [121.583629017, 25.198785252], [121.583708579, 25.198815659], [121.583788042, 25.198846066], [121.583867504, 25.198876473], [121.583947067, 25.19890688], [121.584026529, 25.198937287], [121.584106092, 25.198967694], [121.584185554, 25.198998101], [121.584265017, 25.199028508], [121.584344579, 25.199058915], [121.584424042, 25.199089322], [121.584503504, 25.199119729], [121.584583067, 25.199150136], [121.584662529, 25.199180543], [121.584742092, 25.19921095], [121.584821554, 25.199241357], [121.584901017, 25.199271764], [121.584980579, 25.199302171], [121.585060042, 25.199332578], [121.585139504, 25.199362985], [121.585219067, 25.199393392], [121.585298529, 25.199423799], [121.585378092, 25.199454206], [121.585457554, 25.199484613], [121.585537017, 25.19951502], [121.585616579, 25.199545427], [121.585696042, 25.199575834], [121.585775504, 25.199606241], [121.585855067, 25.199636648], [121.585934529, 25.199667055], [121.586014092, 25.199697462], [121.586093554, 25.199727869], [121.586173017, 25.199758276], [121.586252579, 25.199788683], [121.586332042, 25.19981909], [121.586411504, 25.199849497], [121.586491067, 25.199879904], [121.586570529, 25.199910311], [121.586650092, 25.199940718], [121.586729554, 25.199971125], [121.586809117, 25.200001532], [121.586888579, 25.200031939], [121.586968142, 25.200062346], [121.587047604, 25.200092753], [121.587127167, 25.20012316], [121.587206629, 25.200153567], [121.587286192, 25.200183974], [121.587365654, 25.200214381], [121.587445217, 25.200244788], [121.587524679, 25.200275195], [121.587604242, 25.200305602], [121.587683704, 25.200336009], [121.587763267, 25.200366416], [121.587842729, 25.200396823], [121.587922292, 25.20042723], [121.588001754, 25.200457637], [121.588081317, 25.200488044], [121.588160779, 25.200518451], [121.588240342, 25.200548858], [121.588319804, 25.200579265], [121.588399367, 25.200609672], [121.588478829, 25.200640079], [121.588558392, 25.200670486], [121.588637854, 25.200700893], [121.588717417, 25.2007313], [121.588796879, 25.200761707], [121.588876442, 25.200792114], [121.588955904, 25.200822521], [121.589035467, 25.200852928], [121.589114929, 25.200883335], [121.589194492, 25.200913742], [121.589273954, 25.200944149], [121.589353517, 25.200974556], [121.589432979, 25.201004963], [121.589512542, 25.20103537], [121.589592004, 25.201065777], [121.589671567, 25.201096184], [121.589751029, 25.201126591], [121.589830592, 25.201157008], [121.589910054, 25.201187415], [121.589989617, 25.201217822], [121.590069179, 25.201248229], [121.590148742, 25.201278636], [121.590228304, 25.201309043], [121.590307867, 25.20133945], [121.590387429, 25.201369857], [121.590466992, 25.201400264], [121.590546554, 25.201430671], [121.590626117, 25.201461078], [121.590705679, 25.201491485], [121.590785242, 25.201521892], [121.590864804, 25.201552299], [121.590944367, 25.201582706], [121.591023929, 25.201613113], [121.591103492, 25.20164352], [121.591183054, 25.201673927], [121.591262617, 25.201704334], [121.591342179, 25.201734741], [121.591421742, 25.201765148], [121.591501304, 25.201795555], [121.591580867, 25.201825962], [121.591660429, 25.201856369], [121.591739992, 25.201886776], [121.591819554, 25.201917183], [121.591899117, 25.20194759], [121.591978679, 25.201977997], [121.592058242, 25.202008404], [121.592137804, 25.202038811], [121.592217367, 25.202069218], [121.592296929, 25.202100008], [121.592376492, 25.202130415], [121.592456054, 25.202160822], [121.592535617, 25.202191229], [121.592615179, 25.202221636], [121.592694742, 25.202252043], [121.592774304, 25.20228245], [121.592853867, 25.202312857], [121.592933429, 25.202343264], [121.593012992, 25.202373671], [121.593092554, 25.202404078], [121.593172117, 25.202434485], [121.593251679, 25.202464892], [121.593331242, 25.202495299], [121.593410804, 25.202525706], [121.593490367, 25.202556113], [121.593569929, 25.20258652], [121.593649492, 25.202616927], [121.593729054, 25.202647334], [121.593808617, 25.202677741], [121.593888179, 25.202708148], [121.593967742, 25.202738555], [121.594047304, 25.202768962], [121.594126867, 25.202799369], [121.594206429, 25.202829776], [121.594286092, 25.202860183], [121.594365654, 25.20289059], [121.594445217, 25.202920997], [121.594524779, 25.202951404], [121.594604342, 25.202981811], [121.594683904, 25.203012218], [121.594763467, 25.203042625], [121.594843029, 25.203073032], [121.594922592, 25.203103439], [121.595002154, 25.203133846], [121.595081717, 25.203164253], [121.595161279, 25.20319466], [121.595240842, 25.203225067], [121.595320404, 25.203255474], [121.595399967, 25.203285881], [121.595479529, 25.203316288], [121.595559092, 25.203346695], [121.595638654, 25.203377102], [121.595718217, 25.203407509], [121.595797779, 25.203437916], [121.595877342, 25.203468323], [121.595956904, 25.20349873], [121.596036467, 25.203529137], [121.596116029, 25.203559544], [121.596195592, 25.203589951], [121.596275154, 25.203620358], [121.596354717, 25.203650765], [121.596434279, 25.203681172], [121.596513842, 25.203711579], [121.596593404, 25.203741986], [121.596672967, 25.203772393], [121.596752529, 25.2038028], [121.596832092, 25.203833207], [121.596911654, 25.203863614], [121.596991217, 25.203894021], [121.597070779, 25.203924428], [121.597150342, 25.203954835], [121.597229904, 25.203985242], [121.597309467, 25.204015649], [121.597389029, 25.204046056], [121.597468592, 25.204076463], [121.597548154, 25.20410687], [121.597627717, 25.204137277], [121.597707279, 25.204167684], [121.597786842, 25.204198091], [121.597866404, 25.204228498], [121.597945967, 25.204258905], [121.598025529, 25.204289312], [121.598105092, 25.204319719], [121.598184654, 25.204350126], [121.598264217, 25.204380533], [121.598343779, 25.20441094], [121.598423342, 25.204441347], [121.598502904, 25.204471754], [121.598582467, 25.204502161], [121.598662029, 25.204532568], [121.598741592, 25.204562975], [121.598821154, 25.204593382], [121.598900717, 25.204623789], [121.598980279, 25.204654196], [121.599059842, 25.204684603], [121.599139404, 25.20471501], [121.599218967, 25.204745417], [121.599298529, 25.204775824], [121.599378092, 25.204806231], [121.599457654, 25.204836638], [121.599537217, 25.204867045], [121.599616779, 25.204897452], [121.599696342, 25.204927859], [121.599775904, 25.204958266], [121.599855467, 25.204988673], [121.599935029, 25.20501908], [121.600014592, 25.205049487], [121.600094154, 25.205079894], [121.600173717, 25.205110301], [121.600253279, 25.205140708], [121.600332842, 25.205171115], [121.600412404, 25.205201522], [121.600491967, 25.205231929], [121.600571529, 25.205262336], [121.600651092, 25.205292743], [121.600730654, 25.20532315], [121.600810217, 25.205353557], [121.600889779, 25.205383964], [121.600969342, 25.205414371], [121.601048904, 25.205444778], [121.601128467, 25.205475185], [121.601208029, 25.205505592], [121.601287592, 25.205536009], [121.601367154, 25.205566416], [121.601446717, 25.205596823], [121.601526279, 25.20562723], [121.601605842, 25.205657637], [121.601685404, 25.205688044], [121.601764967, 25.205718451], [121.601844529, 25.205748858], [121.601924092, 25.205779265], [121.602003654, 25.205809672], [121.602083217, 25.205840079], [121.602162779, 25.205870486], [121.602242342, 25.205900893], [121.602321904, 25.2059313], [121.602401467, 25.205961707], [121.602481029, 25.205992114], [121.602560592, 25.206022521], [121.602640154, 25.206052928], [121.602719717, 25.206083335], [121.602799279, 25.206113742], [121.602878842, 25.206144149], [121.602958404, 25.206174556], [121.603037967, 25.206204963], [121.603117529, 25.20623537], [121.603197092, 25.206265777], [121.603276654, 25.206296184], [121.603356217, 25.206326591], [121.603435779, 25.206357008], [121.603515342, 25.206387415], [121.603594904, 25.206417822], [121.603674467, 25.206448229], [121.603754
```

TopoJSON

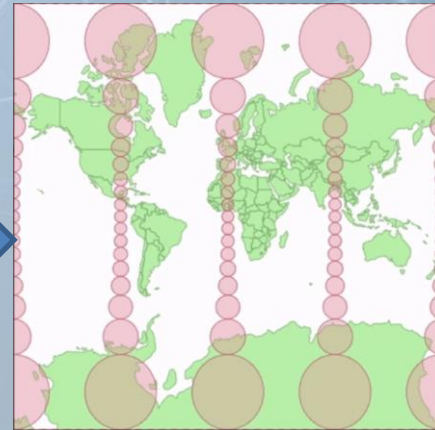
- Comparing with GeoJSON, TopoJSON is smaller
- To load and use topoJSON file to draw map
 - You need <https://d3js.org/topojson.v3.min.js> to convert topojson format to geoJson, then send to data to d3.geoPath
 - We do not use or introduce topoJSON in details in this lecture

Map Projection

- Function that convert from latitude/longitude coordinates to x and y coordinates
- Why?
 - “The true size of” website
 - <https://thetruesize.com/>
 - Our earth is a sphere. If we want to map it to a rectangle, the distortion must exist



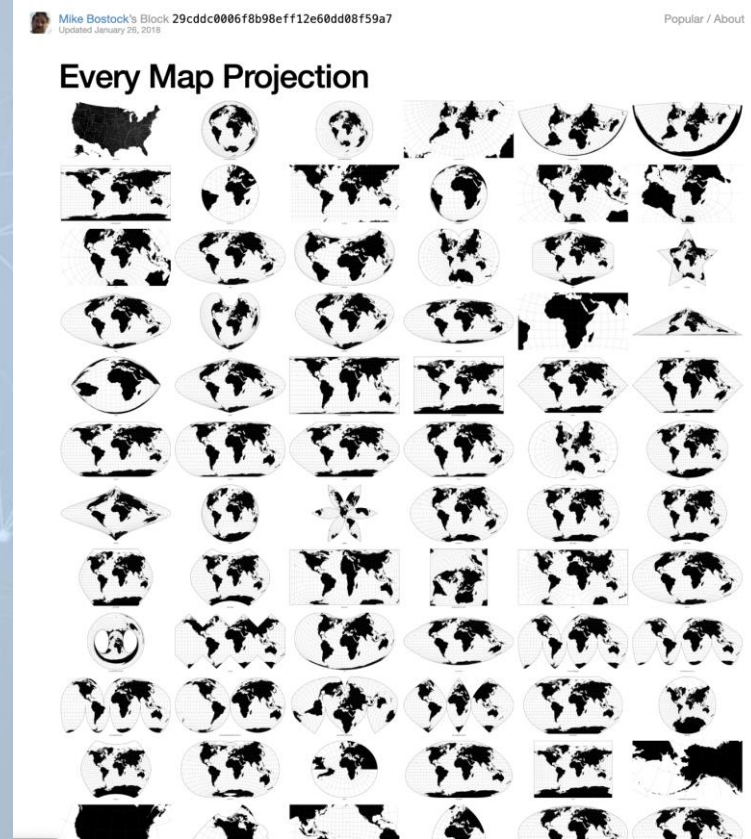
projection



Map Projection

- D3 supports various map projection
 - <https://github.com/d3/d3-geo-projection>
 - Every projection will distort shape, area, distance and/or direction. Every projection also try to keep some attributes have as less distortion as possible
 - Think what properties you do not want to be distorted in your visualization. Then, choose a projection which has as less as possible distortion on these properties

<https://bl.ocks.org/mbostock/29cddc0006f8b98eff12e60dd08f59a7>



Ex08-01

- Load "taiwan.json" and draw city/county with their name

金門縣

連江縣



Ex06-01

- In this example, we use `d3.geoEquirectangular()` as our projection function
 - `.fitExtent(extent, GeoJSON)`
 - The specified region will be scaled to fill the extent on the screen
 - `[[0,0], [width, height]]`:
 - map the top-left latitude/longitude in the GeoJSON to `[0,0]` and bottom-right latitude/longitude in the GeoJSON to `[width, height]`

Load taiwan.json. If the loading is done, run `drawTaiwan()`

```
d3.json("taiwan.json").then(drawTaiwan);

function drawTaiwan(taiwan) {
  var width = 1000;
  var height = 800;

  var projection = d3.geoEquirectangular()
    .fitExtent([[0,0], [width, height]], taiwan);

  var geoGenerator = d3.geoPath()
    .projection(projection);

  var paths = d3.select('svg')
    .selectAll('path')
    .data(taiwan.features)
    .enter()
    .append('path')
    .attr('stroke', 'white')
    .attr('fill', 'steelblue')
    .attr('d', geoGenerator);

  var texts = d3.select('svg')
    .selectAll('text')
    .data(taiwan.features)
    .enter()
    .append('text')
    .attr('text-anchor', 'middle')
    .attr('alignment-baseline', 'middle')
    .attr('opacity', 0.5)
    .text(function(d) {
      return d.properties.NAME_2014;
    })
    .attr('transform', function(d) {
      var center = geoGenerator.centroid(d);
      return 'translate (' + center + ')';
    });
}
```

Content of taiwan.json

Ex06-01

- In this example, we use `d3.geoEquirectangular()` as our projection function
 - `.fitExtent(extent, GeoJSON)`
 - The specified region will be scaled to fill the extent on the screen
 - `[[0,0], [width, height]]`:
 - map the top-left latitude/longitude in the GeoJSON to `[0,0]` and bottom-right latitude/longitude in the GeoJSON to `[width, height]`
- Use “`d3.geoPath()`” to create the generator
 - Remember to set the projection function to it by `.projection()`

Load taiwan.json. If the loading is done, run `drawTaiwan()`

```
d3.json("taiwan.json").then(drawTaiwan);

function drawTaiwan(taiwan) {
  var width = 1000;
  var height = 800;

  var projection = d3.geoEquirectangular()
    .fitExtent([[0,0], [width, height]], taiwan);

  var geoGenerator = d3.geoPath()
    .projection(projection);

  var paths = d3.select('svg')
    .selectAll('path')
    .data(taiwan.features)
    .enter()
    .append('path')
    .attr('stroke', 'white')
    .attr('fill', 'steelblue')
    .attr('d', geoGenerator);

  var texts = d3.select('svg')
    .selectAll('text')
    .data(taiwan.features)
    .enter()
    .append('text')
    .attr('text-anchor', 'middle')
    .attr('alignment-baseline', 'middle')
    .attr('opacity', 0.5)
    .text(function(d) {
      return d.properties.NAME_2014;
    })
    .attr('transform', function(d) {
      var center = geoGenerator.centroid(d);
      return 'translate (' + center + ')';
    });
}
```

Content of taiwan.json

Ex06-01

- Use taiwan.features as the data to draw the paths

Each element in the data array (Taiwan.features) will be sent to “geoGenerator” to generator the path descriptor

```
d3.json("taiwan.json").then(drawTaiwan);

function drawTaiwan(taiwan) {
  var width = 1000;
  var height = 800;

  var projection = d3.geoEquirectangular()
    .fitExtent([[0,0], [width, height]], taiwan);

  var geoGenerator = d3.geoPath()
    .projection(projection);

  var paths = d3.select('svg')
    .selectAll('path')
    .data(taiwan.features)
    .enter()
    .append('path')
    .attr('stroke', 'white')
    .attr('fill', 'steelblue')
    .attr('d', geoGenerator);

  var texts = d3.select('svg')
    .selectAll('text')
    .data(taiwan.features)
    .enter()
    .append('text')
    .attr('text-anchor', 'middle')
    .attr('alignment-baseline', 'middle')
    .attr('opacity', 0.5)
    .text(function(d) {
      return d.properties.NAME_2014;
    })
    .attr('transform', function(d) {
      var center = geoGenerator.centroid(d);
      return 'translate (' + center + ')';
    });
}
```

22 cities/counties (so, 22 paths)

Data to generate a path

Ex06-01

Use Taiwan.features as the data to draw the paths

geoGenerator looks for data in the attribute with the name "geometry" to generate the path

taiwan.feature

```
▼ Array(22)
  ▼ 0:
    ▼ geometry:
      ▼ coordinates: Array(183)
        ► [0 ... 99]
        ► [100 ... 182]
          length: 183
        ► __proto__: Array(0)
      type: "MultiPolygon"
      ► __proto__: Object
    ▼ properties:
      AREA_ID: "Z"
      COUNTYID: "9007"
      IS03166: "LJF"
      NAME_1984: "連江縣"
      NAME_1984_ALIAS: ""
      NAME_2010: "連江縣"
      NAME_2010_ALIAS: ""
      NAME_2014: "連江縣"
      NAME_2014_ALIAS: ""
      SEGIS_COUNTY_ID: "9007"
      _id: 18744625
      ► __proto__: Object
      type: "Feature"
      ► __proto__: Object
  ► 1: {type: "Feature", properties: {...}, geometry: {...}}
  ► 2: {type: "Feature", properties: {...}, geometry: {...}}
  ► 3: {type: "Feature", properties: {...}, geometry: {...}}
  ► 4: {type: "Feature", properties: {...}, geometry: {...}}
  ► 5: {type: "Feature", properties: {...}, geometry: {...}}
  ► 6: {type: "Feature", properties: {...}, geometry: {...}}
  ► 7: {type: "Feature", properties: {...}, geometry: {...}}
  ► 8: {type: "Feature", properties: {...}, geometry: {...}}
  ► 9: {type: "Feature", properties: {...}, geometry: {...}}
  ► 10: {type: "Feature", properties: {...}, geometry: {...}}
  ► 11: {type: "Feature", properties: {...}, geometry: {...}}
  ► 12: {type: "Feature", properties: {...}, geometry: {...}}
  ► 13: {type: "Feature", properties: {...}, geometry: {...}}
  ► 14: {type: "Feature", properties: {...}, geometry: {...}}
  ► 15: {type: "Feature", properties: {...}, geometry: {...}}
  ► 16: {type: "Feature", properties: {...}, geometry: {...}}
  ► 17: {type: "Feature", properties: {...}, geometry: {...}}
  ► 18: {type: "Feature", properties: {...}, geometry: {...}}
  ► 19: {type: "Feature", properties: {...}, geometry: {...}}
  ► 20: {type: "Feature", properties: {...}, geometry: {...}}
  ► 21: {type: "Feature", properties: {...}, geometry: {...}}
  length: 22
  ► __proto__: Array(0)
```

d3.json("taiwan.json").then(drawTaiwan);

```
function drawTaiwan(taiwan) {
  var width = 1000;
  var height = 800;

  var projection = d3.geoEquirectangular()
    .fitExtent([[0,0], [width, height]], taiwan);

  var geoGenerator = d3.geoPath()
    .projection(projection);

  var paths = d3.select('svg')
    .selectAll('path')
    .data(taiwan.features)
    .enter()
    .append('path')
    .attr('stroke', 'white')
    .attr('fill', 'steelblue')
    .attr('d', geoGenerator);

  var texts = d3.select('svg')
    .selectAll('text')
    .data(taiwan.features)
    .enter()
    .append('text')
    .attr('text-anchor', 'middle')
    .attr('alignment-baseline', 'middle')
    .attr('opacity', 0.5)
    .text(function(d) {
      return d.properties.NAME_2014;
    })
    .attr('transform', function(d) {
      var center = geoGenerator.centroid(d);
      return 'translate(' + center + ')';
    });
}
```

Each element in the data array (Taiwan.features) will be sent to "geoGenerator" to generate the path descriptor

Ex06-01

Put the cities/counties name on the map

```
▼ Array(22) 1
  ▼ 0:
    ▼ geometry:
      ▼ coordinates: Array(183) 6
        ► [0 ... 99]
        ► [100 ... 182]
          length: 183
        ► __proto__: Array(0)
      type: "MultiPolygon"
      ► __proto__: Object
    ▼ properties:
      AREA_ID: "Z"
      COUNTYID: "9007"
      IS03166: "LJF"
      NAME_1984: "連江縣"
      NAME_1984_ALIAS: ""
      NAME_2010: "連江縣"
      NAME_2010_ALIAS: ""
      NAME_2014: "連江縣"
      NAME_2014_ALIAS: ""
      SEGIS_COUNTY_ID: "9007"
      _id: 18744625
      ► __proto__: Object
      type: "Feature"
      ► __proto__: Object
    ► 1: {type: "Feature", properties: {...}, geometry: {...}}
    ► 2: {type: "Feature", properties: {...}, geometry: {...}}
    ► 3: {type: "Feature", properties: {...}, geometry: {...}}
    ► 4: {type: "Feature", properties: {...}, geometry: {...}}
    ► 5: {type: "Feature", properties: {...}, geometry: {...}}
    ► 6: {type: "Feature", properties: {...}, geometry: {...}}
    ► 7: {type: "Feature", properties: {...}, geometry: {...}}
    ► 8: {type: "Feature", properties: {...}, geometry: {...}}
    ► 9: {type: "Feature", properties: {...}, geometry: {...}}
    ► 10: {type: "Feature", properties: {...}, geometry: {...}}
    ► 11: {type: "Feature", properties: {...}, geometry: {...}}
    ► 12: {type: "Feature", properties: {...}, geometry: {...}}
    ► 13: {type: "Feature", properties: {...}, geometry: {...}}
    ► 14: {type: "Feature", properties: {...}, geometry: {...}}
    ► 15: {type: "Feature", properties: {...}, geometry: {...}}
    ► 16: {type: "Feature", properties: {...}, geometry: {...}}
    ► 17: {type: "Feature", properties: {...}, geometry: {...}}
    ► 18: {type: "Feature", properties: {...}, geometry: {...}}
    ► 19: {type: "Feature", properties: {...}, geometry: {...}}
    ► 20: {type: "Feature", properties: {...}, geometry: {...}}
    ► 21: {type: "Feature", properties: {...}, geometry: {...}}
    length: 22
    ► __proto__: Array(0)
```

taiwan.feature

```
d3.json("taiwan.json").then(drawTaiwan);
```

```
function drawTaiwan(taiwan) {
  var width = 1000;
  var height = 800;
```

```
  var projection = d3.geoEquirectangular()
    .fitExtent([[0,0], [width, height]], taiwan);
```

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

```
  var paths = d3.select('svg')
    .selectAll('path')
    .data(taiwan.features)
    .enter()
    .append('path')
    .attr('fill', 'steelblue')
    .attr('d', geoGenerator);
```

Calculate centroid of the cities/counties to place the name

```
  var texts = d3.select('svg')
    .selectAll('text')
    .data(taiwan.features)
    .enter()
    .append('text')
    .attr('text-anchor', 'middle')
    .attr('alignment-baseline', 'middle')
    .attr('opacity', 0.5)
    .text(function(d) {
      return d.properties.NAME_2014;
    })
    .attr('transform', function(d) {
      var center = geoGenerator.centroid(d);
      return 'translate(' + center + ')';
    });
```


Ex06-01

- Convert longitude and latitudes to x-y and draw a circle
- Send longitude/latitudes to the projection function. It returns an array with x and y.
 - 120.9575, 23.47: Yushan (玉山)

```
d3.select('svg').append('circle')  
  .attr('cx', projection( [120.9575, 23.47 ])[0] )  
  .attr('cy', projection( [120.9575, 23.47 ])[1] )  
  .attr('fill', 'red')  
  .attr('r', 5);
```

