|  |
| --- |
| js |
| {  sample: function() {  var list = taiwanmap.town.names;  return {  county: [{name: "縣市", data: list.map(function(d,i) { return d[0]; })}],  town: [{name: "鄉鎮", data: list.map(function(d,i) { return d[1]; })}],  value: [{name: "數值", data: list.map(function(d,i) {  return Math.round(Math.random() \*100); })  }]  };  },  dimension: {  value: { type: [plotdb.Number], require: true, desc: "該鄉鎮的數值" },  county: { type: [plotdb.String], require: true, desc: "縣市的名字，用來對應地圖區塊" },  town: { type: [plotdb.String], require: true, desc: "鄉鎮的名字，用來對應地圖區塊" }  },  config: {  fontFamily: {},  background: {},  textFill: {},  fontSize: {},  margin: {},  colorEmpty: {},  palette: {},  stroke: {},  strokeWidth: {},  legendLabel: {},  legendShow: {},  popupShow: {},  aggregateMode: {  name: "縣市數值",  type: [plotdb.Choice(["加總","平均"])],  default: "平均",  category: "Value"  },  showAll: {  name: "顯示所有鄉鎮",  type: [plotdb.Boolean],  default: true,  category: "Global Settings"  },  unit: {}  },  init: function() {  var that = this, i;  this.id = Math.random().toString(16).substring(2);  this.names = taiwanmap.town.names;  d3.select(this.root).select("defs filter#innerStroke").attr({id: "innerStroke-" + this.id});  d3.select(this.root).select("defs filter#shadow").attr({id: "shadow-" + this.id});  this.infoPanel = d3.select(this.root).append("div").style({  position: "absolute",  bottom: "10px",  left: "80px",  "min-width": "120px",  display: "inline-block",  float: "left",  width: "180px",  padding: "6px",  "border-radius": "5px",  background: "rgba(255,255,255,0.8)",  "text-align": "center",  height: "18px",  "line-height": "18px",  opacity: 1,  "pointer-event": "none"  }).text("ⓘ 點擊縣市看鄉鎮");  this.backBtn = d3.select(this.root).append("div").style({  position: "absolute",  bottom: "10px",  left: "20px",  padding: "6px",  height: "18px",  "border-radius": "5px",  border: "1px solid rgba(0,0,0,0.5)",  cursor: "pointer",  background: "rgba(255,255,255,0.8)",  "line-height": "18px",  opacity: 0.2  }).text("縮小").on("click", function(d,i) {  that.lastActiveCounty = that.activeCounty;  that.activeCounty = null;  that.render();  });  this.svg = d3.select(this.root).append("svg");  this.bkrect = this.svg.append("rect").on("click", function(d,i) {  that.lastActiveCounty = that.activeCounty;  that.activeCounty = null;  that.render();  });  this.popup = plotd3.html.popup(this.root).on("mousemove", function(d,i,popup) {  popup.select(".title").text(d.properties.C\_Name + (d.properties.T\_Name || ""));  popup.select(".value").text(  (d.properties.value == undefined ? "無數值"  :parseInt(d.properties.value\*100)/100) + " " + (that.config.unit || "")  );  return true;  });  this.dataGroup = this.svg.append("g").attr({class: "data-group"});  this.legendGroup = this.svg.append("g").attr({class: "legend-group"});  this.counties = [];  this.features = {  town: topojson.feature(  taiwanmap.town.topojson, taiwanmap.town.topojson.objects.town  ).features,  county: topojson.feature(  taiwanmap.county.topojson, taiwanmap.county.topojson.objects.county  ).features  };  this.features.all = this.features.town.concat(this.features.county);  this.map = {town: {}, county: {}};  for(var i=0,c,t;i<this.features.town.length;i++) {  c = this.features.town[i].properties.C\_Name;  t = this.features.town[i].properties.T\_Name;  if(!this.map.town[c]) this.map.town[c] = {};  this.map.town[c][t] = this.features.town[i];  }  for(i=0,c;i<this.features.county.length;i++) {  c = this.features.county[i].properties.C\_Name;  if(!this.map.county[c]) this.map.county[c] = {};  this.map.county[c] = this.features.county[i];  }  },  parse: function() {  var that = this;  if(!this.dimension.value.fields.length) this.data.map(function(d,i) { d.value = 0; });  var cvalue = {};  this.data.map(function(d,i) {  d.county = that.normalname(d.county.trim());  d.town = d.town.trim();  d.feature = that.map.town[d.county][d.town];  if(d.feature) d.feature.properties.value = d.value;  if(!cvalue[d.county]) cvalue[d.county] = {value: d.value, count: 1};  else {  cvalue[d.county].value += d.value;  cvalue[d.county].count += 1;  }  });  for(k in cvalue) {  if(this.config.aggregateMode == "平均") {  cvalue[k].value /= (cvalue[k].count || 1);  }  if(cvalue[k].value) that.map.county[k].properties.value = cvalue[k].value;  }  this.data = this.data.filter(function(d,i) { return d.feature});  this.valueRange = d3.extent(this.data.map(function(d,i) { return d.value; }));  if(this.valueRange[0] == this.valueRange[1]) this.valueRange[1]++;  },  normalname: function(it) {  if(!it) return it;  it = it.replace("台","臺");  if(it == "臺北縣") return "新北市";  if(it == "臺中縣") return "臺中市";  if(it == "高雄縣") return "高雄市";  if(it == "臺南縣") return "臺南市";  if(it == "桃園縣") return "桃園市";  return it;  },  bind: function() {  var that = this, sel,sel2;  sel = this.dataGroup.selectAll("g.towns").data(taiwanmap.county.names);  sel.exit().remove();  sel = sel.enter().append("g").attr({class: "towns"});  this.dataGroup.selectAll("g.towns").each(function(d,i) {  var sel,node = d3.select(this);  sel = node.selectAll("path.data.geoblock.town").data(that.features.town.filter(function(e,j) {  return e.properties.C\_Name == d;  }));  sel.exit().remove();  sel.enter().append("path").attr({class: "geoblock"});  });  sel = this.dataGroup.selectAll("path.data.geoblock.county").data(this.features.county);  sel.exit().remove();  sel = sel.enter().append("path").attr({class: "geoblock"});  sel = this.dataGroup.selectAll("path.geoblock");  sel.attr({class: function(d,i) {  return "data geoblock " + (d.properties.T\_Name ? "town" : "county");  }}).filter(function(d,i) { return d3.select(this).classed("county"); }).on("click", function(d,i) {  if(that.smallScreen) that.infoPanel.text("ⓘ 點擊鄉鎮看數值");  else that.infoPanel.transition("fadeout").duration(500).style({opacity: 0});  that.lastActiveCounty = that.activeCounty;  that.activeCounty = that.activeCounty == d ? null : d;  that.render();  });  sel = sel.filter(function(d,i) { return d3.select(this).classed("town"); }).on("click", function(d,i) {  that.infoPanel.text(d.properties.C\_Name + d.properties.T\_Name + " : " + d.properties.value + " " + (that.config.unit || ""));  that.lastActiveTown = that.activeTown;  that.activeTown = d3.select(this);  that.activeTown.style({ filter: "url(#innerStroke-" + that.id + ")" });  if(that.lastActiveTown) that.lastActiveTown.style({filter: "none"});  /\* click town to zoom out. disabled \*/  if(false) {  var ret = (that.config.showAll ? that.map.county[d.properties.C\_Name]: null);  that.lastActiveCounty = that.activeCounty;  that.activeCounty = that.activeCounty == ret ? null : ret;  that.render();  }  });  if(!this.smallScreen && this.config.popupShow) this.popup.nodes(sel);  },  resize: function() {  var that = this;  var box = this.root.getBoundingClientRect();  var width = this.width = box.width;  var height = this.height = box.height;  this.margin = Math.min(this.config.margin, width/4, height/4);  this.svg.attr({  width: width + "px", height: height + "px",  viewBox: [0,0,width,height].join(" "),  preserveAspectRatio: "xMidYMid"  }).style({overflow: "hidden"});  this.smallScreen = (/android|webos|iphone|ipad|ipod|blackberry|iemobile|opera mini/i.test(navigator.userAgent.toLowerCase()));  this.popup.fontSize(this.config.fontSize);  this.cScale = plotdb.Palette.scale.auto(this.config.palette, this.dimension.value.fields);  this.legend = plotd3.rwd.legend()  .orient("right")  .scale(this.cScale)  .label(this.config.legendLabel || "")  .fontSize(this.config.fontSize || 12);  this.legendGroup.call(this.legend);  this.legendSize = this.legend.offset();  this.projection = d3.geo.mercator()  .translate([width / 2, height / 2])  .scale(1)  .precision(.1);  this.path = d3.geo.path().projection(  function(coord) {  if(coord[1]>25.5) coord[1] = coord[1] - 1;  if(coord[0]<119) coord[0] = coord[0] + 1.5;  if(coord[0]>122.1) coord[0] = coord[0] - 1.2;  if(coord[0]>121.4 && coord[1]< 22.1) {  coord[0] -= 0.3;  coord[1] += 0.3;  }  return that.projection(coord);  }  );  var rateW = (that.width - this.margin \* 2) \* 16;  var rateH = (that.height - this.margin \* 2) \* 14;  this.rate = this.allRate = Math.min(rateW, rateH);  var center = this.center = [121,23.6];  this.projection.scale(this.rate).center(center);  bounds = this.path.bounds(this.features.county[0]);  this.posfix = [0,0];  this.projection.translate([width/2 + this.posfix[0], height/2 + this.posfix[1]]);  this.bkrect.attr({  x: 0, y: 0, width: this.width, height: this.height, fill: "rgba(255,255,255,0.001)"  });  },  render: function() {  var that = this;  var dx, dy, x, y, bounds, scale, translate;  d3.select(this.root).selectAll("div").style({  "font-size": this.config.fontSize + "px",  "font-family": this.config.fontFamily  });  if(this.activeCounty && that.backBtn.style("opacity")<1) {  that.backBtn.transition("opacity").duration(500).style({opacity: 1});  } else if(!this.activeCounty && that.backBtn.style("opacity")==1) {  that.backBtn.transition("opacity").duration(500).style({opacity: 0.2});  }  this.dataGroup.selectAll("g.data-group > g.towns, g.data-group > path").sort(function(a,b) {  if(that.activeCounty && that.activeCounty.properties.C\_Name == a) return 1;  if(that.activeCounty && that.activeCounty.properties.C\_Name == b) return -1;  if(a && a.properties && !a.properties.T\_Name) return 1;  if(b && b.properties && !b.properties.T\_Name) return -1;  return 0;  });  this.dataGroup.selectAll("path.data").filter(function(d,i) {  return (  (that.activeCounty && that.activeCounty.properties.C\_Name == d.properties.C\_Name) ||  (that.lastActiveCounty && that.lastActiveCounty.C\_Name == d.properties.C\_Name) ||  (that.activeCounty == d) ||  (that.lastActiveCounty == d)  );  }).attr({d: that.path}).filter(function(d,i) {  return d3.select(this).classed("town");  }).attr({opacity: 1});  if(this.activeCounty) {  bounds = that.path.bounds(this.activeCounty);  dx = bounds[1][0] - bounds[0][0],  dy = bounds[1][1] - bounds[0][1],  scale = 0.9 \* this.rate / Math.max(  dx / (this.width - this.margin \* 2),  dy / (this.height - this.margin \* 2)  );  this.rate = scale;  this.projection.scale(scale).center(d3.geo.centroid(this.activeCounty)).translate([this.width/2, this.height/2]);  bounds = that.path.bounds(this.activeCounty);  var posfix = [  this.width/2 - (bounds[1][0] + bounds[0][0])/2,  this.height/2 - (bounds[1][1] + bounds[0][1])/2  ];  this.projection.translate([this.width/2 + posfix[0], this.height/2 + posfix[1]]);  } else {  if(this.rate != this.allRate) {  this.rate = this.allRate;  this.projection.scale(this.rate).center(this.center).translate([this.width/2, this.height/2]);  }  }  if(this.config.fontFamily) d3.select(this.root).style("font-family", this.config.fontFamily);  d3.select(this.root).style("background-color", this.config.background);  this.svg.selectAll("text").attr({  "font-size": that.config.fontSize,  "fill": that.config.textFill  });  this.legendGroup.attr({  transform: ["translate(", this.width - this.legendSize[0] - 20, (this.height - this.legendSize[1])/2, ")"].join(" "),  display: this.config.legendShow ? "block" : "none"  });    this.svg.selectAll("g.towns").attr({  filter: function(d,i) {  return that.activeCounty && that.activeCounty.properties.C\_Name == d  ? "url(#shadow-" + that.id + ")" : "";  }  });  this.svg.selectAll("path.data").filter(function(d,i) {  return !d3.select(this).attr("fill");  }).attr({ fill: this.config.colorEmpty });  var matcher = function(d,i) {  if(d3.select(this).classed("county")) return !that.config.showAll;  if(  (that.activeCounty && that.activeCounty.properties.C\_Name == d.properties.C\_Name) ||  (that.lastActiveCounty && that.lastActiveCounty.properties.C\_Name == d.properties.C\_Name)  ) return true;  return that.config.showAll;  };  this.svg.selectAll("path.data").filter(matcher).transition("morph").duration(500).attr({  d: that.path,  opacity: function(d,i) {  if(that.activeCounty == d) return 0;  if(that.activeCounty && d3.select(this).classed("town") && (  (that.activeCounty.properties.C\_Name != d.properties.C\_Name) &&  (that.lastActiveCounty && that.lastActiveCounty.properties.C\_Name != d.properties.C\_Name)  )) return that.config.showAll && d3.select(this).classed("town") ? 1 : 0;  return 1;  },  "pointer-events": function(d,i) { return that.activeCounty == d ? "none" : "initial"; }  });  if(this.config.showAll) this.svg.selectAll("path.county").transition("opacity").duration(500).attr({  display: "none"  });  var nodes = this.svg.selectAll("path.data").filter(function(d,i) { return !matcher.apply(this, [d,i]); });  nodes.attr({  "pointer-events": "none"//function(d,i) { return that.activeCounty == d ? "none" : "initial"; }  });  nodes.filter(function(d,i) { return d3.select(this).classed("county"); }).transition("opacity").duration(500).attr({  opacity: 0,  });  if(this.config.showAll) nodes = nodes.filter(function(d,i) { return false; });  nodes.filter(function(d,i) { return d3.select(this).classed("town"); }).attr({opacity: 0});  this.dataGroup.selectAll("path.data").transition("color").duration(function(d,i) {  return d3.select(this).attr("fill") ? 500 : 0;  }).attr({  fill: function(d,i) {  var v = d.properties.value;  if(!v) return that.config.colorEmpty;  return that.cScale(v);  },  stroke: function(d,i) {  var c = d3.hsl(that.config.stroke);  if(d3.select(this).classed("county")) { c.l = (c.l + 1)/2; }  return c.toString();  },  "stroke-width": function(d,i) {  return that.config.strokeWidth \* (d3.select(this).classed("county") ? 1 : 0.5);  },  });  }  } |
| css |
| path, rect {  -webkit-tap-highlight-color: rgba(0,0,0,0);  }  .legend-group {  pointer-events: none;  } |
| html |
| <svg height="0">  <defs>  <!-- patch id within code -->  <filter id="innerStroke" x="0" y="0" width="1" height="1">  <feMorphology operator="erode" in="SourceGraphic" radius="3" result="erode"/>  <feColorMatrix type="matrix" in="SourceGraphic" result="color"  values="0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0"/>  <feMerge>  <feMergeNode in="color"/>  <feMergeNode in="erode"/>  </feMerge>  </filter>  <!-- patch id within code -->  <filter id="shadow" x="-0.5" y="-0.5" width="2" height="2">  <feGaussianBlur in="SourceAlpha" result="blur" stdDeviation="9.5"/>  <feMorphology operator="dilate" in="SourceGraphic" radius="2" result="dilate"/>  <feColorMatrix type="matrix" in="dilate" result="color" values="0.2 0 0 0 0 0 0.2 0 0 0 0 0 0.2 0 0 0 0 0 1 0"/>  <feMerge>  <feMergeNode in="blur"/>  <feMergeNode in="color"/>  <feMergeNode in="SourceGraphic"/>  </feMerge>  </filter>  </defs>  </svg> |