**制作泰森多边形步骤：**

1. 利用d3.geom.voronoi()函数，参数是Point[[x,y]…],[x,y]就是离散点，单位是像素，返回值是每个泰森多边形的边界点数组形成的大数组polygons[]。
2. 添加<svg>画布，在<svg>画布中用<path>构图，有多少个多边形就多少个path，所以采用svg.selectAll("path")

.data(polygon)

.enter().append(“path”)

.attr(“d”: function(d,i) { return "M" + d.join("L") + "Z"; }) //path如何利用对应数据作图就是依靠这个d属性，这里画多边形，所以采用上述语句，其中d是被绑定的数据，即绑定的多边形的边界点集合，也可表示成polygon[i]

1. 对应点采用

svg.selectAll("circle")

.data(Point)

.enter().append("circle")

.attr({“r”:1.5，"transform":function(d) { return "translate(" + d + ")"; }}) //点的具体位置，这里通过设置平移位置(即绑定点的像素位置[x,y])，也可以通过设置cx，cy属性确定圆心。

**百度地图添加覆盖物步骤：**

1. new BMap.Overlay() 创建自定义覆盖物

2、实现 .initialize=function(){

<div>

<svg>

<g>

<path> //这两个在 .draw中创建

<circle>

创建上述一个结点，用d3.select()包装一下

map.getPanes().markerPane.appendChild(div); 把div结点添加到markerPane图层

return div；

}

3、实现 .draw=function(){ //当地图变化时，draw会重新调用

map.pointToOverlayPixel(Point(lng,lat)); //经纬度转换为像素坐标，需要把geojson中Point(x,y)经纬度先封装成BMap.Point再传入,转换为像素放入像素数组(大数组中包含多个Point[x,y]数组),然后交给d3.geom.voronoi()处理得到polygons，接着就是d3画出<path><circle>.

}

1. map.addOverlay(overlay); //最后添加覆盖物
2. 自定义覆盖物没有addEventListener,只能在创建出来的div元素上添加事件。

**只在屏幕范围内绘制泰森多边形：**

定义的svg画布偏移量是4000，那么初始化屏幕左上角是[4000,4000]，左移百度地图会出现4000-x的情况。借助d3.geom.voronoi().clipExtent([[x1,y1],[x2,y2]]) 其中[x1,y1]是实时的左上角像素坐标，[x2,y2]右下角，此函数可以让产生的多边形只在这个矩形内。

**锁定对应景点多边形区域：**

只需要把对应多边形getPath()的数据传给new BMap.Polygon()，这里注意数据库中获取的是数值，需要封装成new BMap.Point。

**动态加载景点按钮：**

采用for循环，动态append <li>结点，需要给<li>属性id name附上数据库中的值，这样事件函数中才能id name，不能在事件函数中用变量var 实现，否则总是触发时最后一个id name。

**动态添加景点区域：**

利用百度地图绘图工具，注意必须要有对应的css文件，利用模态框与用户交互，获得name数据，根据画出的overlay，获得相应数据，封装成json传给后台处理。

**显示特定泰森多边形内包含的多个基站信息：**

这个多边形是用D3.js画的， 和百度地图没关系(百度地图接受的只是一张画好的图)，所以只能在d3.draw(path)时添加事件机制。由于给的信息是聚合后的点，其实里面是几个基站信息，所以需要根据cellId数组在另一张表中找到里面的基站信息数组，在屏幕上显示出这个数组包含的3-4个基站信息。关于之前不能及时响应颜色变化，是因为BMap也响应click事件，而多边形没有着色的话作为覆盖物添加到BMap上，不能完全覆盖住BMap，使得click有可能只被BMap响应，解决方案为多边形着色，使得多边形完全处于BMap之上。

**给泰森多边形根据权值着色：**

这里利用d3的颜色渐变插值法：

var compute = d3.interpolateHsl(d3.rgb(0,255,0).hsl(),d3.rgb(255,0,0).hsl());

var linear = d3.scale.linear()

.domain([0,1000])

.range([0,1]);

只需把权值传入到d3.path.fill中，即可填充颜色。

**显示各个基站的热力图信息：**

利用heatmap.js，传入点的经纬度和对应的count权值形成json数据即可，即可像覆盖物一样添加到地图中。

**ECharts显示人口迁移图：**

下拉按钮/菜单格式：<div> <button> <ul> <li>共四个元素。

利用Echarts+BMap实现

1. 引入echarts.js+BMap,整个body放置一个div用来加载BMap
2. 配置require.config，path：//echarts.js的路径，package：//需要用到的扩展包路径，给其取一个别名name。配置require，添加用到的所有name(echarts,BMap,config,map.js…)。
3. require的回调函数参数就是上述用到的js文件对象,也可以通过require(‘name’)获取到对应的js对象，下述功能都是此函数中执行。
4. 利用main.js中函数BMapExtension初始化地图，和普通地图一样设置显示参数，这里用到setMapstyle改变原来的地图显示。
5. 这里最重点在于数据option对象设置，数据驱动，这个后期也可以通过setOption()修改，但这里注意setOption默认行为是合并，而非覆盖。
6. 获取echarts的dom容器，BMapExt.getEchartsContainer();
7. 通过initEcharts初始化echarts并获取实例var myChart = BMapExt.initECharts(container);
8. BMapExt.setOption(option);把option添加到BMap中。

**人口迁移图能够改变迁入迁出效果：**

1、只要setOption(series，{notManger:true}),series设置不同表现数据格式，图标就能改变样式。实现对应的排名也随着迁入迁出改变，这需要根据图例中选择按钮的返回值确定id，再结合IO值，使用range(id,IO)。

**人口迁移图实现点击图例即排名：**

1. 图例添加点击事件要config.js支持，charts.on(config.EVENT.LEGEND\_SELECTED,callback())。在回调函数中需要调用排名函数，传入id和IO值，依靠两者参数确定获取到迁入还是迁出的points县市数组，排序后写入table。

**人口迁移图添加日历选择：**

1. 这里引入jquery-datetimepicker.js,需要注意和echarts.js冲突，必须在此之前引入。利用日期选择的回调函数，根据时间找到所需的迁移二维数组。

**基站搜索功能：**

1. 利用select2.js插件，编写ajax与后台交互即可。

**点击景区有相应的图片和介绍：**

1. 利用百度地图的infowindow和画多边形api即可完成。

**利用JTS套件完成Voronoi图的制作：**

1. JTS是java开发的一套用于地理数据计算的组件，这里需要利用VoronoiDiagramBuilder类
2. 通过传入一系列经纬度坐标Coordinate，获得生成的一系列泰森多边形Geometry
3. 这里有一个问题需要注意，只能通过多边形获得其包含的点((Coordinate) Geometry.getUserData())，而不是像D3那样点和多边形一一对应，所以这里还需要一个sql语句通过经纬度来找到数据库中对应的那个基站，更改其Polygon\_p信息。

**Echarts折线图实现人口预测：**

人口预测算法利用一次指数平滑，结合图标制作，这里时间轴问题没解决。