用中国地图来表示不同省份的分布情况

1. 数量分布情况

查询代码如下:

const mongodb = require('./mongodb')

const fs = require('fs')

const regions = JSON.parse(fs.readFileSync('./region.json').toString())

const path = require('path')

const $conn = mongodb.$conn

;(async function()

{

const db = await $conn

const user = db.collection('user')

const result = {}

for(const gb in regions)

{

const num = await user.find

(

{

'profile.province': Number(gb)

}

).count()

console.log(`| ${regions[gb]} | ${num} |`)

result[regions[gb]] = num

}

// console.log(result)

fs.writeFileSync(path.resolve(\_\_dirname, 'province.json'), JSON.stringify(result))

})()

结果如下表所示:

|  |  |
| --- | --- |
| 省份 | 用户数量 |
| 北京 | 11766 |
| 天津 | 2781 |
| 河北 | 5923 |
| 山西 | 3889 |
| 内蒙古 | 3526 |
| 辽宁 | 5265 |
| 吉林 | 2901 |
| 黑龙江 | 3730 |
| 上海 | 7317 |
| 江苏 | 14033 |
| 浙江 | 11514 |
| 安徽 | 7044 |
| 福建 | 5835 |
| 江西 | 5772 |
| 山东 | 11301 |
| 河南 | 10174 |
| 湖北 | 8216 |
| 湖南 | 7733 |
| 广东 | 23985 |
| 广西 | 4246 |
| 海南 | 1146 |
| 重庆 | 4387 |
| 四川 | 10242 |
| 贵州 | 2656 |
| 云南 | 5013 |
| 西藏 | 516 |
| 陕西 | 6274 |
| 甘肃 | 2798 |
| 青海 | 917 |
| 宁夏 | 969 |
| 新疆 | 5598 |
| 台湾 | 635 |
| 香港 | 832 |
| 澳门 | 196 |

对每个省份的用户人数使用D3.js可视化之后结果如图4.2.1.1所示:

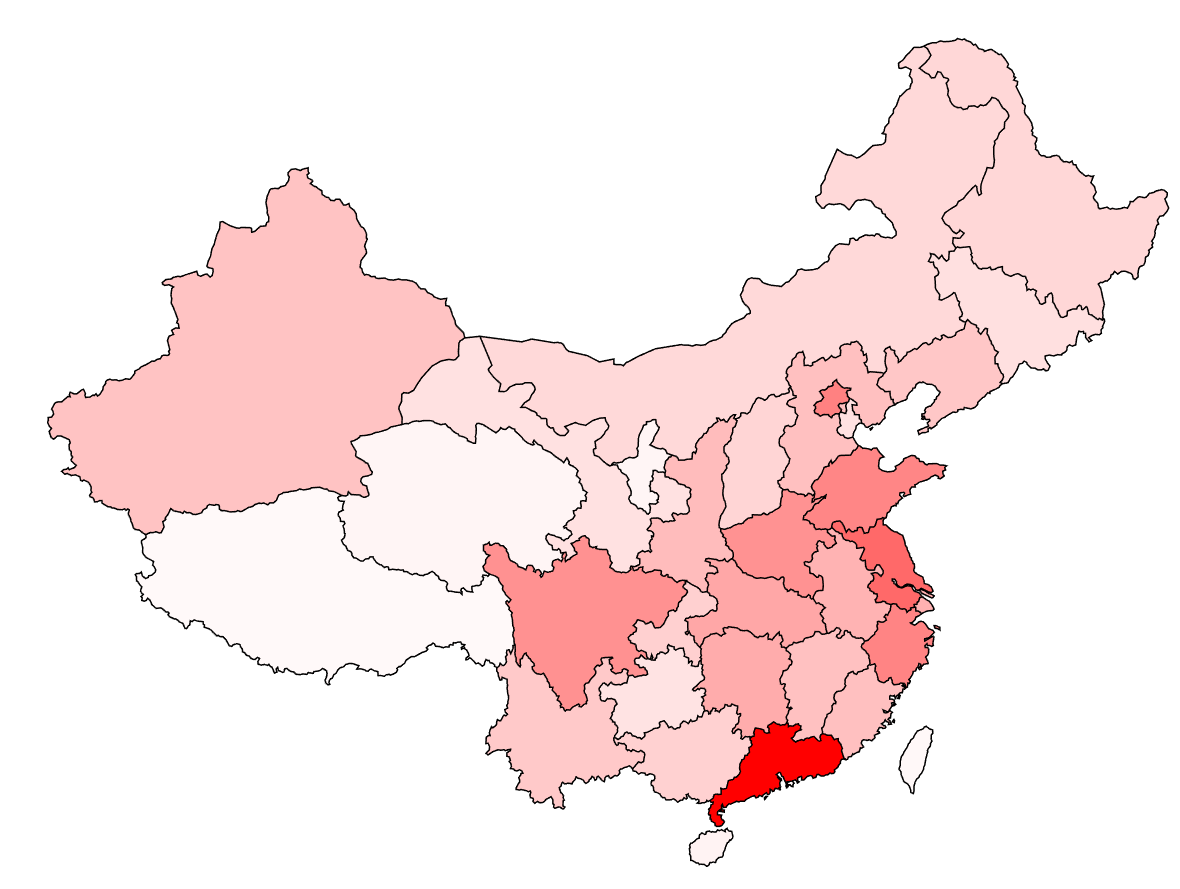
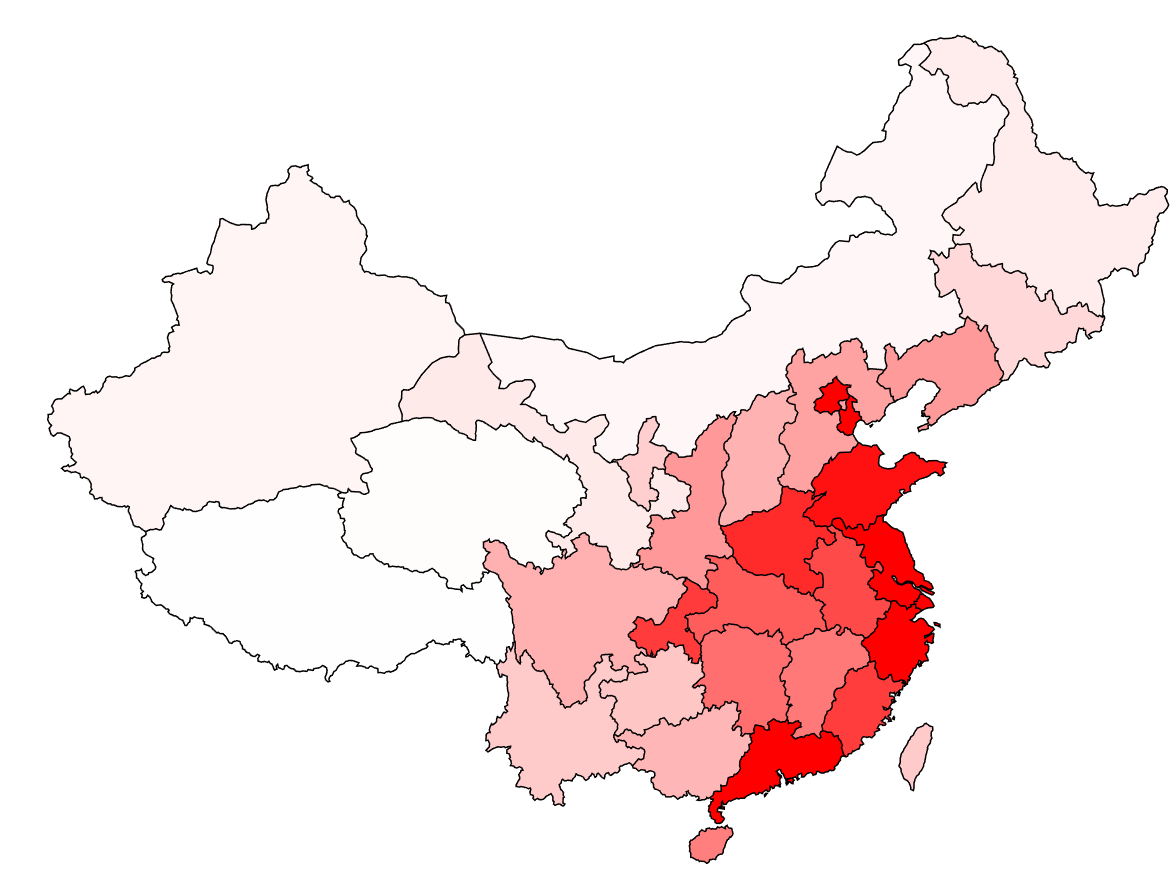


图4.2.1.1

对每个省份的用户密度(用户数 / 省份面积)使用D3.js可视化之后结果如图4.2.1.2所示:

图4.2.1.2

1. 性别分布情况

在基于所有的样本进行分析之后, 得到的不同性别的用户在全国各个省份的分布情况.

代码如下:

const mongodb = require('./mongodb')

const fs = require('fs')

const regions = JSON.parse(fs.readFileSync('./region.json').toString())

const path = require('path')

const $conn = mongodb.$conn

;(async function()

{

const db = await $conn

const user = db.collection('user')

const result = {}

for(const gb in regions)

{

const female = await user.find

(

{

'profile.province': Number(gb),

'profile.gender': 2

}

).count()

const male = await user.find

(

{

'profile.province': Number(gb),

'profile.gender': 1

}

).count()

console.log(`| ${regions[gb]} | ${male} | ${female} |`)

result[regions[gb]] = [male, female]

}

fs.writeFileSync(path.resolve(\_\_dirname, 'sex.json'), JSON.stringify(result))

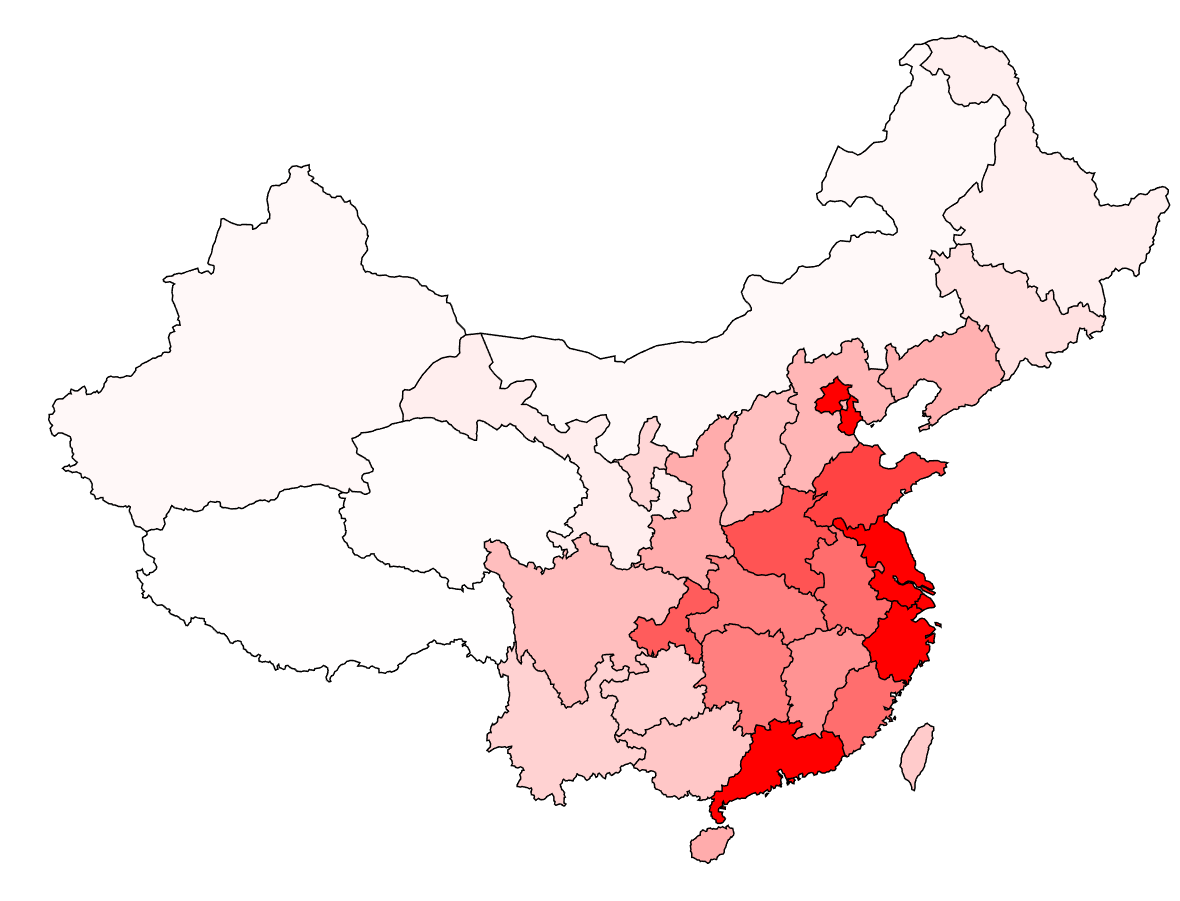
})()

结果如下:

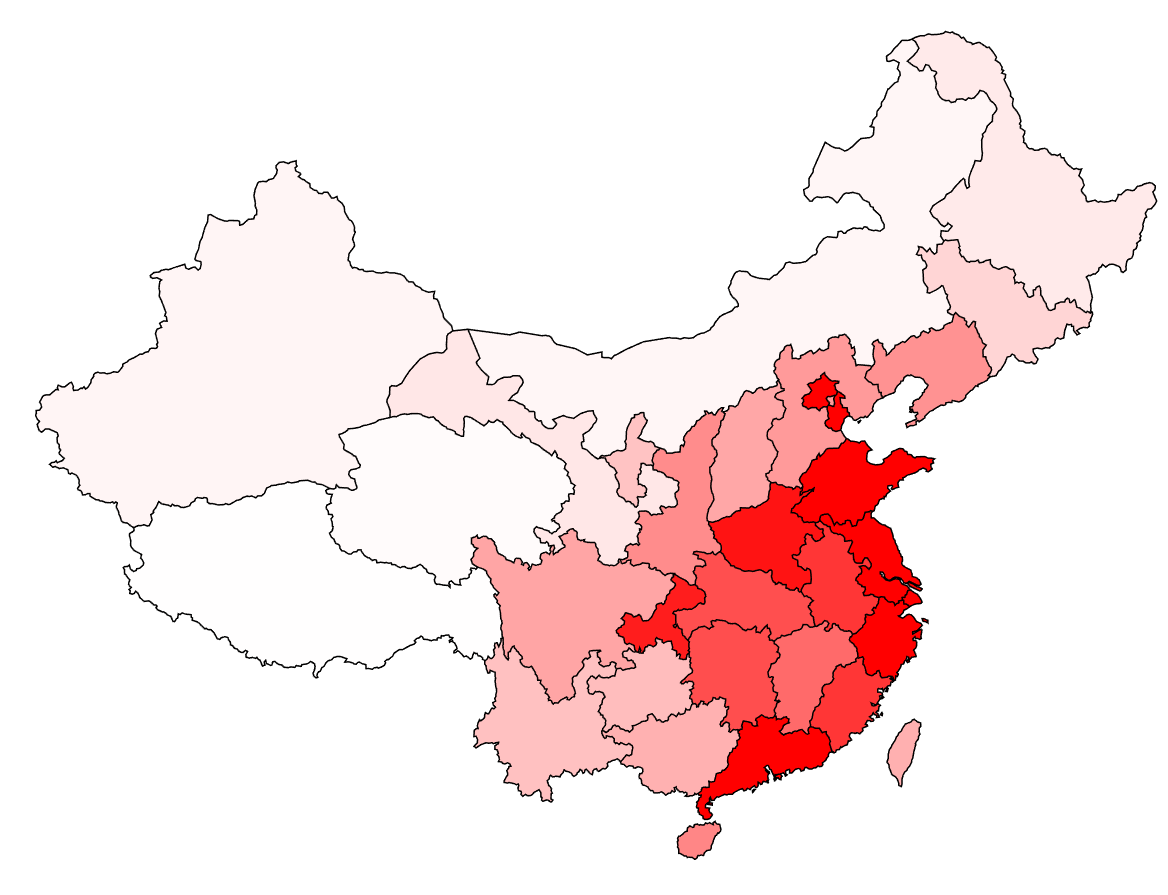
|  |  |  |
| --- | --- | --- |
| 省份 | 男性用户数量 | 女性用户数量 |
| 北京 | 5513 | 4058 |
| 天津 | 1249 | 1005 |
| 河北 | 2696 | 2197 |
| 山西 | 1688 | 1522 |
| 内蒙古 | 1581 | 1357 |
| 辽宁 | 2462 | 1951 |
| 吉林 | 1324 | 1092 |
| 黑龙江 | 1654 | 1467 |
| 上海 | 3223 | 2641 |
| 江苏 | 6319 | 5136 |
| 浙江 | 4848 | 4184 |
| 安徽 | 3233 | 2671 |
| 福建 | 2582 | 2065 |
| 江西 | 2547 | 2188 |
| 山东 | 5046 | 4204 |
| 河南 | 4523 | 3837 |
| 湖北 | 3729 | 3027 |
| 湖南 | 3314 | 3250 |
| 广东 | 11343 | 8195 |
| 广西 | 1956 | 1546 |
| 海南 | 574 | 383 |
| 重庆 | 1904 | 1725 |
| 四川 | 4558 | 3971 |
| 贵州 | 1195 | 1046 |
| 云南 | 2288 | 2005 |
| 西藏 | 232 | 200 |
| 陕西 | 2908 | 2433 |
| 甘肃 | 1274 | 1104 |
| 青海 | 438 | 357 |
| 宁夏 | 389 | 403 |
| 新疆 | 2696 | 2058 |
| 台湾 | 274 | 298 |
| 香港 | 382 | 358 |
| 澳门 | 96 | 76 |

使用d3.js进行可视化展示之后的结果如下:

1. 男性用户在全国的密度分布, 如图4.2.2.1所示

图4.2.2.1

1. 女性用户在全国的密度分布, 如图4.2.2.2所示

图4.2.2.2

3. 女性用户在所在的省份的用户中的比率在全国的分布, 如图4.2.2.3所示

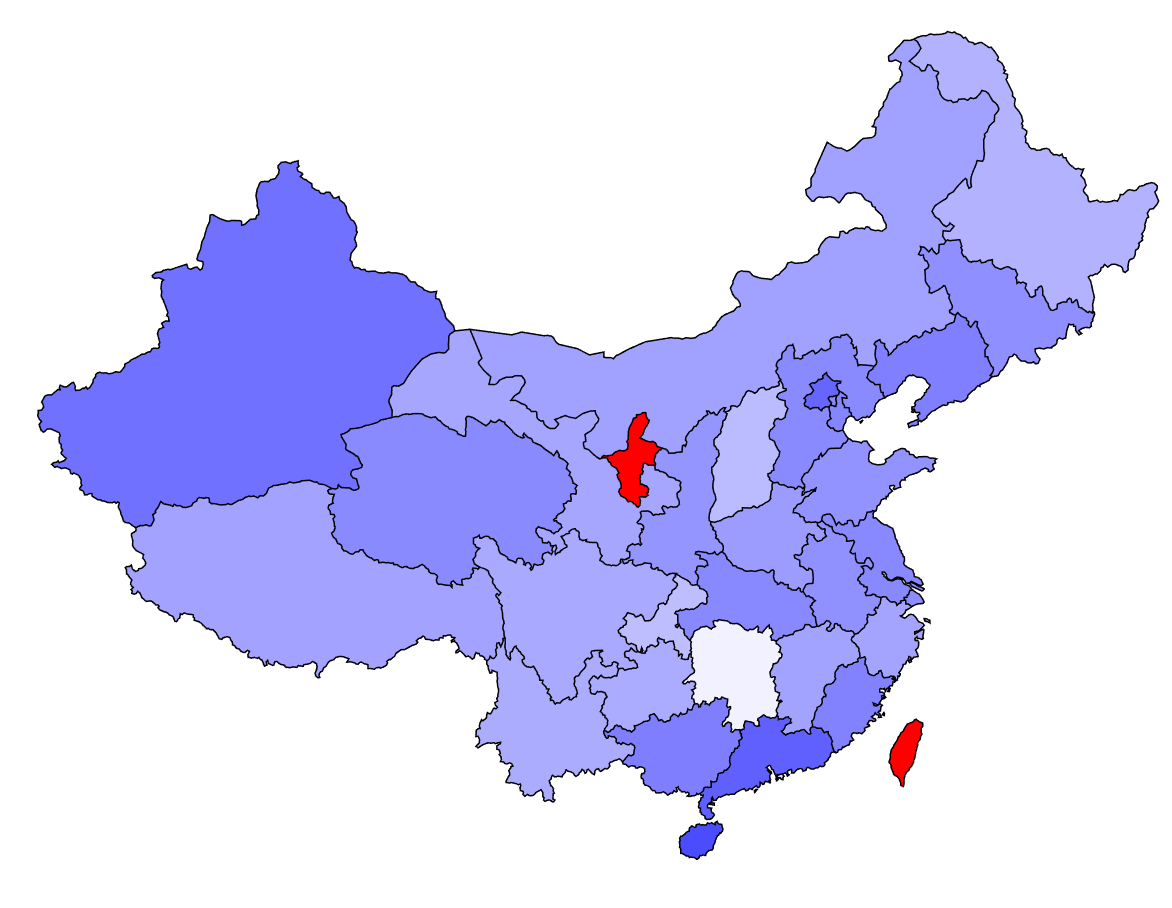


图4.2.2.3

1. 年龄分布情况

查询代码如下:

const mongodb = require('./mongodb')

const fs = require('fs')

const regions = JSON.parse(fs.readFileSync('./region.json').toString())

const path = require('path')

const $conn = mongodb.$conn

;(async function()

{

const db = await $conn

const user = db.collection('user')

const result = {}

for(const gb in regions)

{

const age70 = await user.find

(

{

'profile.province': Number(gb),

'profile.birthday': {$gte: 0, $lt: 315532800000}

}

).count()

const age80 = await user.find

(

{

'profile.province': Number(gb),

'profile.birthday': {$gte: 315532800000, $lt: 631152000000}

}

).count()

const age90 = await user.find

(

{

'profile.province': Number(gb),

'profile.birthday': {$gte: 631152000000, $lt: 946684800000}

}

).count()

const age00 = await user.find

(

{

'profile.province': Number(gb),

'profile.birthday': {$gte: 946684800000, $lt: 1262304000000}

}

).count()

const age10 = await user.find

(

{

'profile.province': Number(gb),

'profile.birthday': {$gte: 1262304000000}

}

).count()

console.log(`| ${regions[gb]} | ${age70} | ${age80} | ${age90} | ${age90} | ${age10} |`)

result[regions[gb]] = [age70, age80, age90, age00, age10]

}

fs.writeFileSync(path.resolve(\_\_dirname, 'age.json'), JSON.stringify(result))

})()

结果如下:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 省份 | 70后 | 80后 | 90后 | 00后 | 10后 |
| 北京 | 132 | 1167 | 2849 | 2849 | 271 |
| 天津 | 11 | 153 | 756 | 756 | 26 |
| 河北 | 26 | 308 | 1388 | 1388 | 50 |
| 山西 | 12 | 174 | 948 | 948 | 23 |
| 内蒙古 | 7 | 153 | 891 | 891 | 21 |
| 辽宁 | 22 | 324 | 1392 | 1392 | 53 |
| 吉林 | 14 | 146 | 746 | 746 | 23 |
| 黑龙江 | 23 | 175 | 969 | 969 | 30 |
| 上海 | 58 | 553 | 1981 | 1981 | 127 |
| 江苏 | 57 | 656 | 3820 | 3820 | 139 |
| 浙江 | 42 | 578 | 2749 | 2749 | 145 |
| 安徽 | 15 | 245 | 1919 | 1919 | 65 |
| 福建 | 21 | 306 | 1266 | 1266 | 78 |
| 江西 | 9 | 199 | 1342 | 1342 | 64 |
| 山东 | 29 | 568 | 3034 | 3034 | 109 |
| 河南 | 22 | 409 | 2383 | 2383 | 93 |
| 湖北 | 24 | 391 | 2206 | 2206 | 117 |
| 湖南 | 24 | 301 | 1830 | 1830 | 98 |
| 广东 | 75 | 1246 | 5387 | 5387 | 301 |
| 广西 | 14 | 211 | 891 | 891 | 53 |
| 海南 | 3 | 54 | 254 | 254 | 17 |
| 重庆 | 18 | 202 | 1211 | 1211 | 54 |
| 四川 | 34 | 451 | 2686 | 2686 | 137 |
| 贵州 | 8 | 94 | 625 | 625 | 28 |
| 云南 | 11 | 189 | 1293 | 1293 | 61 |
| 西藏 | 2 | 28 | 152 | 152 | 7 |
| 陕西 | 16 | 275 | 1759 | 1759 | 39 |
| 甘肃 | 6 | 111 | 746 | 746 | 27 |
| 青海 | 3 | 28 | 199 | 199 | 10 |
| 宁夏 | 2 | 33 | 224 | 224 | 11 |
| 新疆 | 21 | 266 | 1582 | 1582 | 83 |
| 台湾 | 13 | 47 | 152 | 152 | 13 |
| 香港 | 5 | 99 | 286 | 286 | 38 |
| 澳门 | 1 | 35 | 65 | 65 | 9 |

使用d3.js进行可视化展示之后的结果如下:

1. 70后在全国的分布, 如图4.2.3.1所示

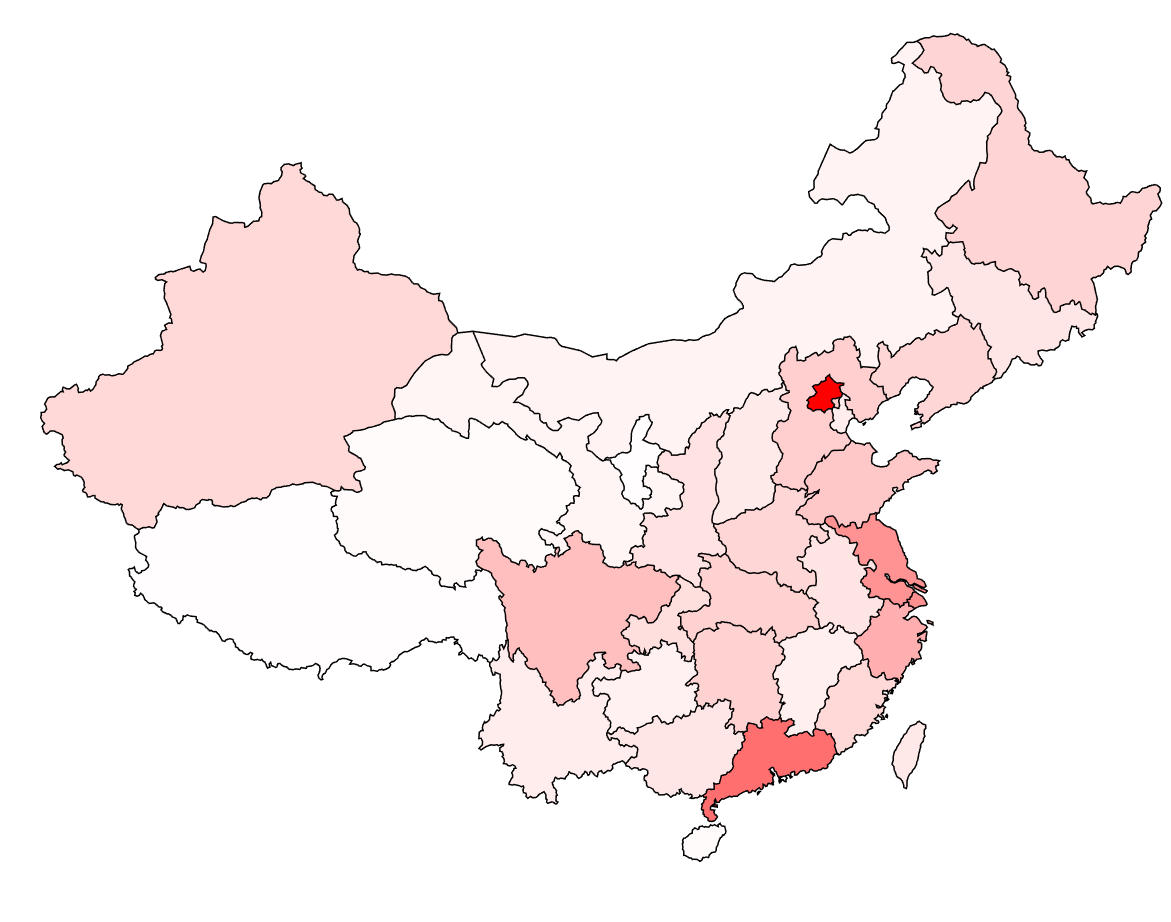


图4.2.3.1

2. 80后在全国的分布, 如图4.2.3.2所示

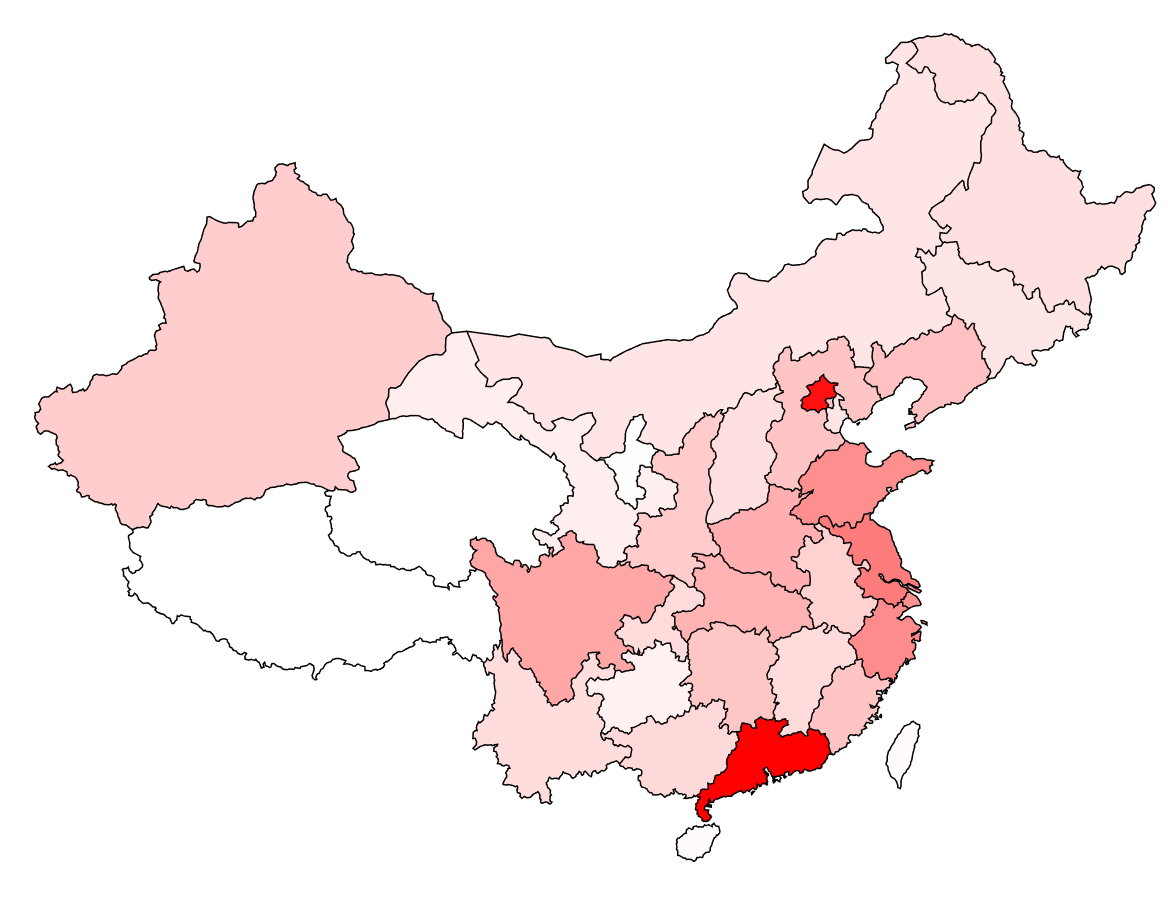


图4.2.3.2

1. 90后在全国的分布, 如图4.2.3.3所示

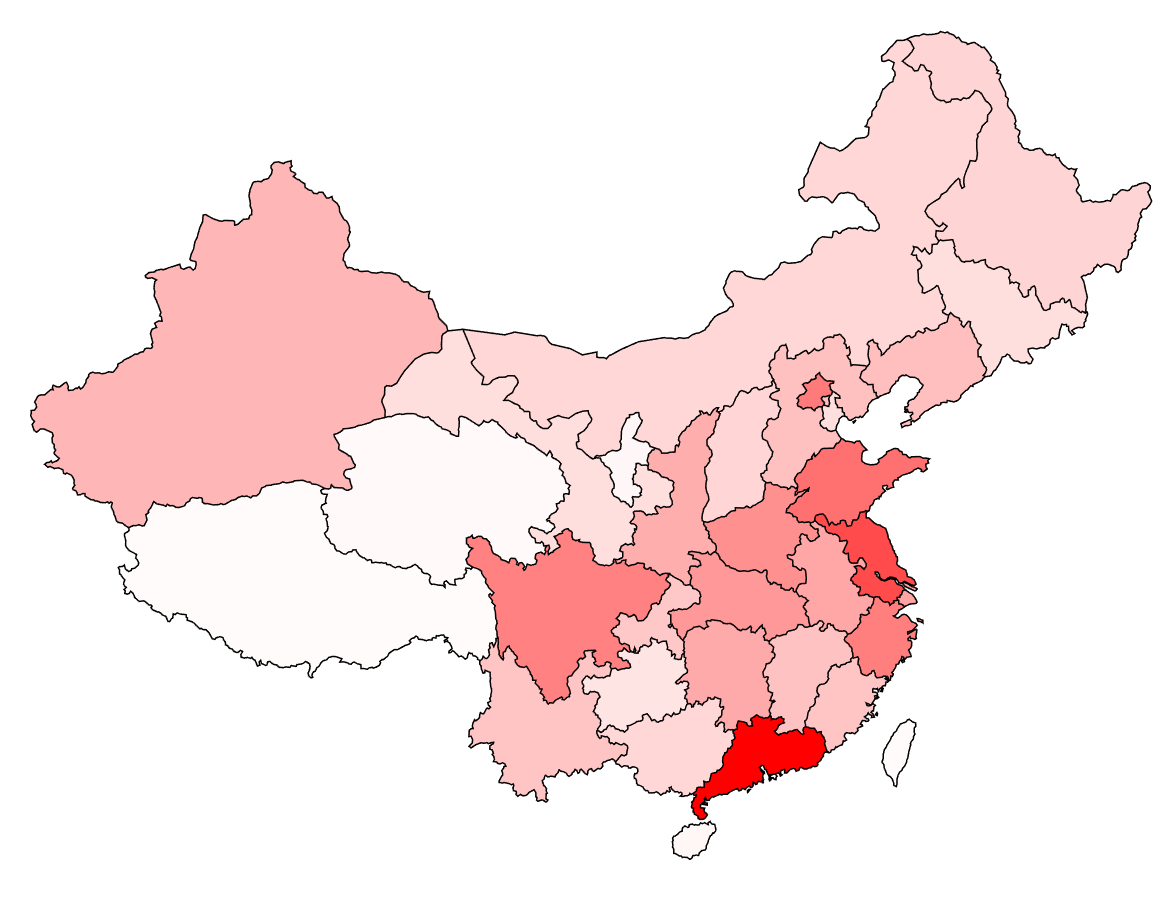


图4.2.3.3

1. 00后在全国的分布, 如图4.2.3.4所示

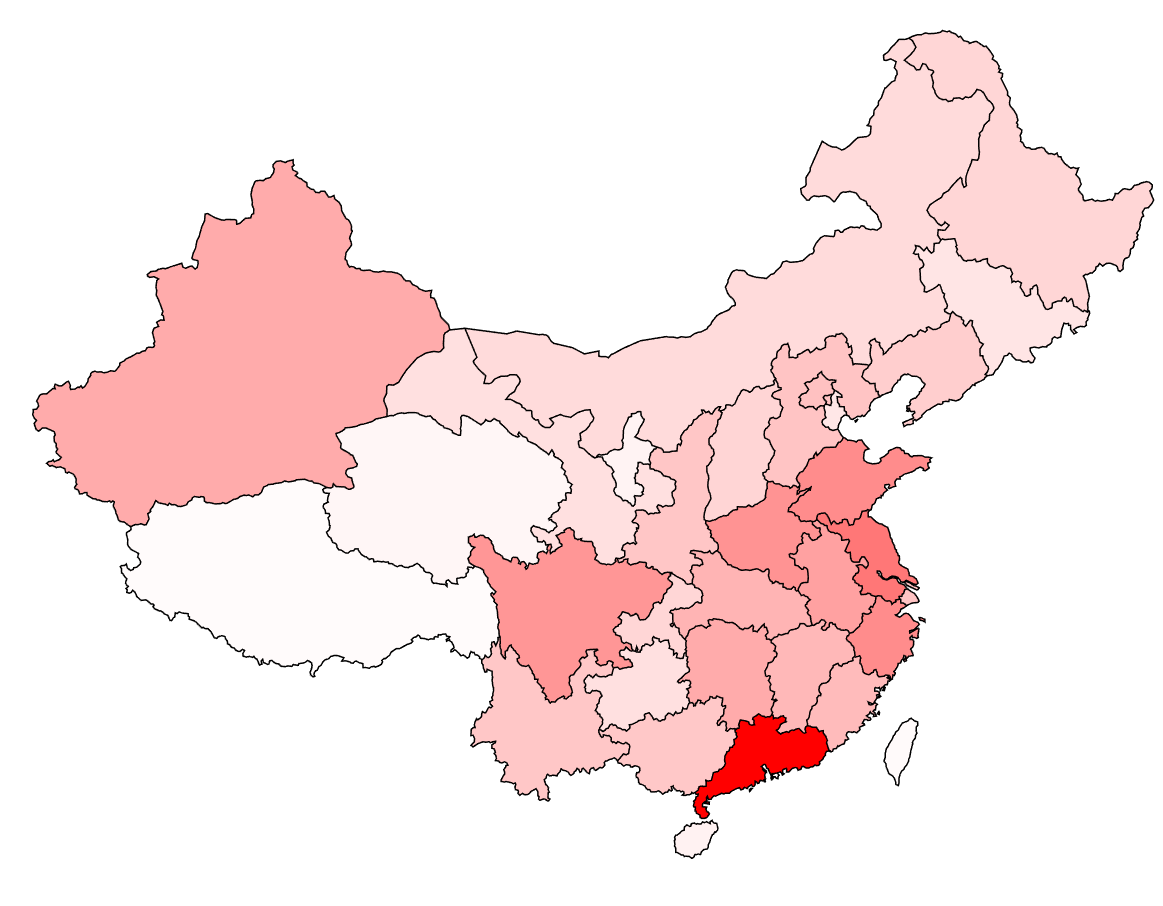


图4.2.3.4

5. 10后在全国的分布, 如图4.2.3.5所示

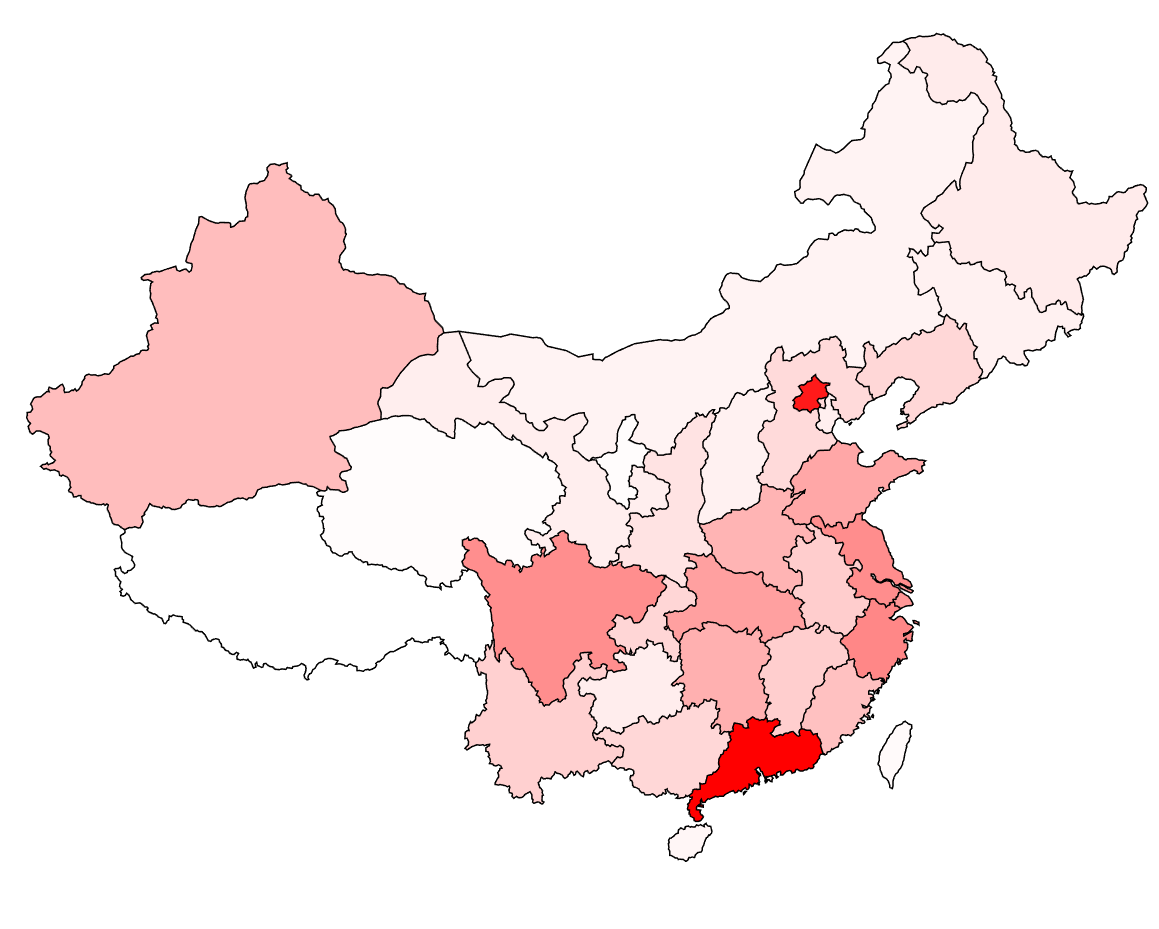


图4.2.3.5