使用弦图来进行展示

1. 男女之间互相关注的数量的对比

查询男性用户对女性用户的关注数量: {'from.gender': 1, 'to.gender': 2}

查询男性用户对男性用户的关注数量: {'from.gender': 1, 'to.gender': 1}

查询女性用户对女性用户的关注数量: {'from.gender': 2, 'to.gender': 2}

查询女性用户对男性用户的关注数量: {'from.gender': 2, 'to.gender': 1}

具体代码如下:

const mongodb = require('./mongodb')

const $conn = mongodb.$conn

;(async function()

{

const $db = await $conn

const $follow\_sex = $db.collection('follow\_sex')

const male2male = await $follow\_sex.find

(

{

'from.gender': 1,

'to.gender': 1,

}

).count()

const male2female = await $follow\_sex.find

(

{

'from.gender': 1,

'to.gender': 2,

}

).count()

const female2male = await $follow\_sex.find

(

{

'from.gender': 2,

'to.gender': 1,

}

).count()

const female2female = await $follow\_sex.find

(

{

'from.gender': 2,

'to.gender': 2,

}

).count()

const matrix =

[

[male2male, male2female],

[female2male, female2female]

]

console.log(matrix)

})()

查询后, 结果如下表所示:

|  |  |  |
| --- | --- | --- |
| 关注 | 男性 | 女性 |
| 男性 | 1312487 | 927842 |
| 女性 | 633642 | 478432 |

使用d3.js可视化之后结果如图4.3.1所示:

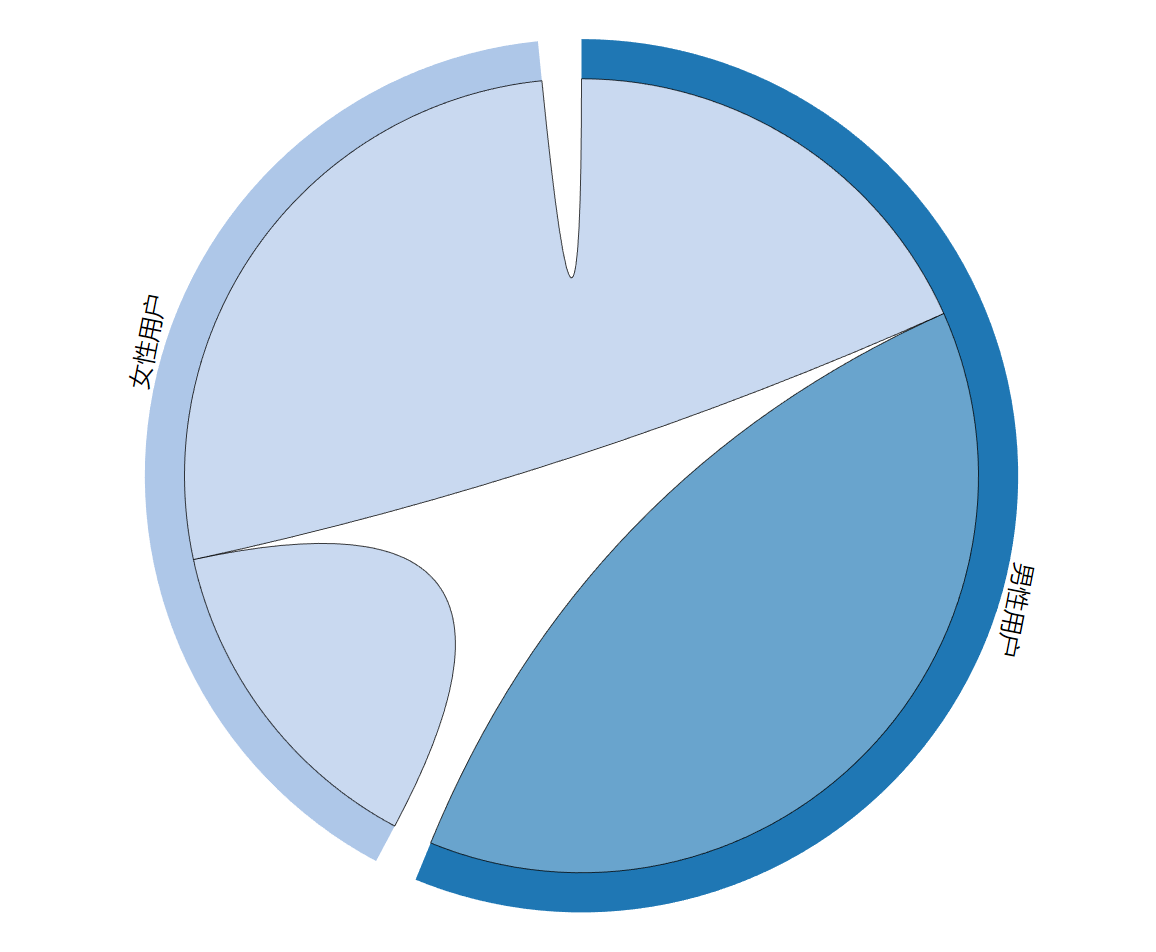


图4.3.1

1. 不同省份的用户之间相互关注数量的对比

不同省份的用户之间相互关注数量的对比的查询代码具体如下:

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 $follow\_sex = $db.collection('follow\_sex')

const rows = []

for(let name in regions)

{

const row = []

for(let name\_inner in regions)

{

const num = await $follow\_sex.find

(

{

'from.province': regions[name],

'to.province': regions[name\_inner],

}

).count()

row.push(num)

}

rows.push(row)

}

console.log(rows)

})()

查询结果较为复杂, 这里不列表展示.

使用d3.js可视化之后结果如图4.3.2所示:

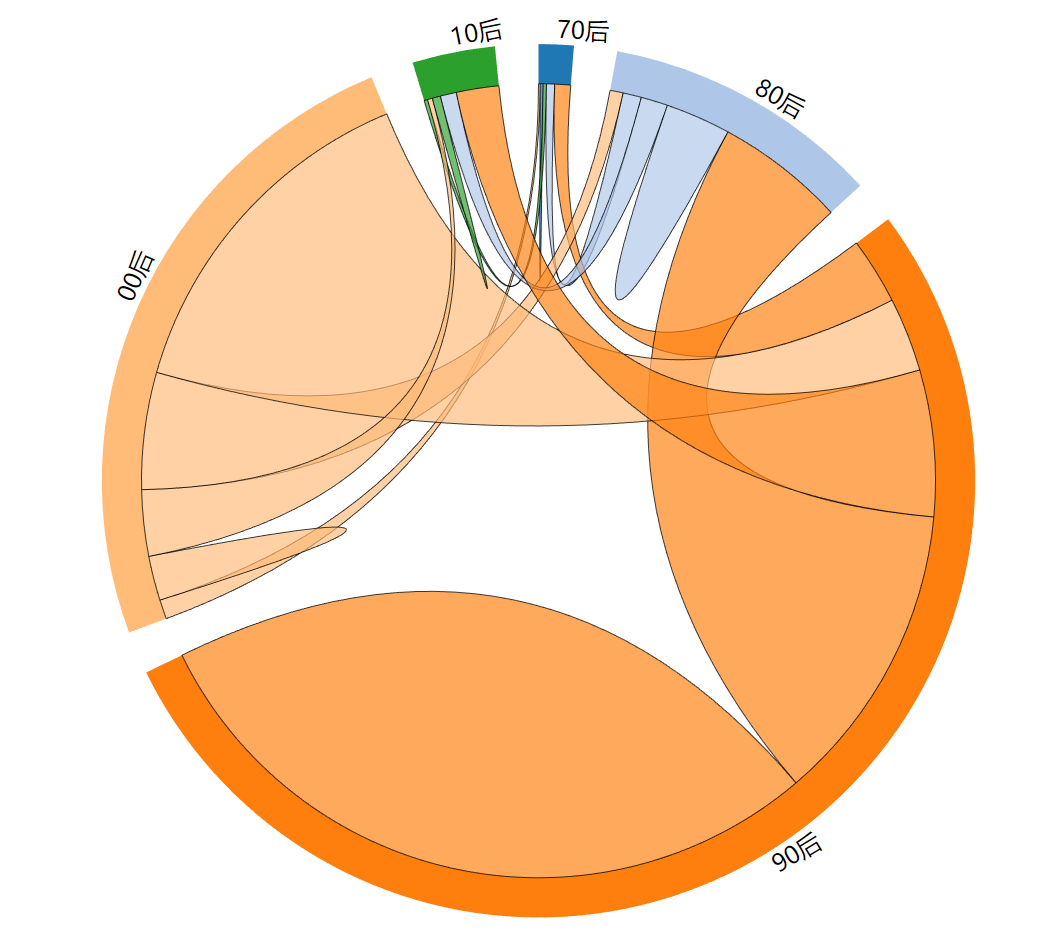


图4.3.2

1. 不同年龄的用户的互相关注数量对比

查询不同年龄的用户互相关注数量对比的具体代码如下:

const mongodb = require('./mongodb')

const year70 = 0

const year80 = 315532800000

const year90 = 631152000000

const year00 = 946684800000

const year10 = 1262304000000

const year20 = 1562304000000

const years = [year70, year80, year90, year00, year10, year20]

const $conn = mongodb.$conn

;(async function()

{

const $db = await $conn

const $user = $db.collection('user')

const $follow\_sex = $db.collection('follow\_sex')

const rows = []

for(let index = 0; index < years.length - 1; index ++)

{

const row = []

for(let index\_inner = 0; index\_inner < years.length - 1; index\_inner ++)

{

const num = await $follow\_sex.find

(

{

'from.birthday':

{

$gte: years[index],

$lt: years[index + 1]

},

'to.birthday':

{

$gte: years[index\_inner],

$lt: years[index\_inner + 1]

},

}

).count()

row.push(num)

}

rows.push(row)

}

console.log(rows)

})()

查询结果较为复杂, 这里不列表展示.

使用d3.js可视化之后结果如图4.3.3所示:

图4.3.3