

重庆科技大学

实验报告

课程名称: 数据可视化技术

学 院: 计算机科学与工程学院(人工智能学院)

专业班级: 智科 2023-02

学生姓名: 刘志贵

学 号: 2023440221

重庆科技大学实验报告

课程名称	数据可视化技术	实验项目名称	绘制 3D 图表和统计地图	
开课学院	计算机科学与工程学院		实验日期	2025. 10. 22
学生姓名	刘志贵	学号	2023440221	专业班级
指导教师	杨怡康		实验成绩	

一、实验目的

掌握利用 Matplotlib 进行高级图表绘制的方法

二、实验结果

- 终端操作: pip install pyproj geos basemap, 然后在 vscode 的 demo.ipynb 文件输入 from mpl_toolkits.basemap import Basemap 后, 显示结果如下:



```
from mpl_toolkits.basemap import Basemap
```

图 1 导入 basemap 结果

- 在三维空间中绘制不同颜色、不同样式的点, 随机指定点的颜色、样式、位置等参数。

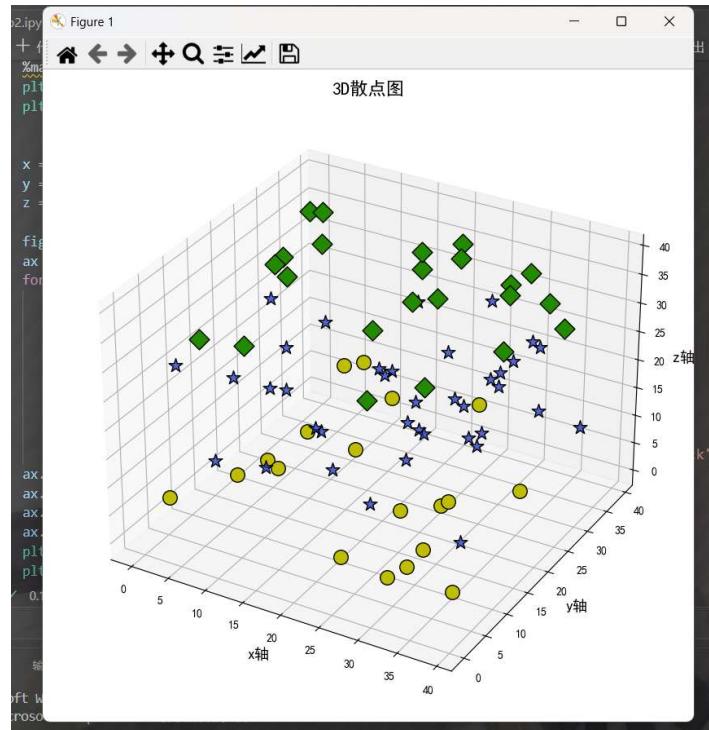


图 2 三维随机样式图

说明：这里我使用的 qt5 绘制图像，每个点的位置是随机的，颜色和样式是伪随机，指定了在一定坐标下显示固定的颜色和样式，代码见附录。

3. 在每帧动画中更新点的颜色，形成动画。下面是动画连续 5 帧的图片。

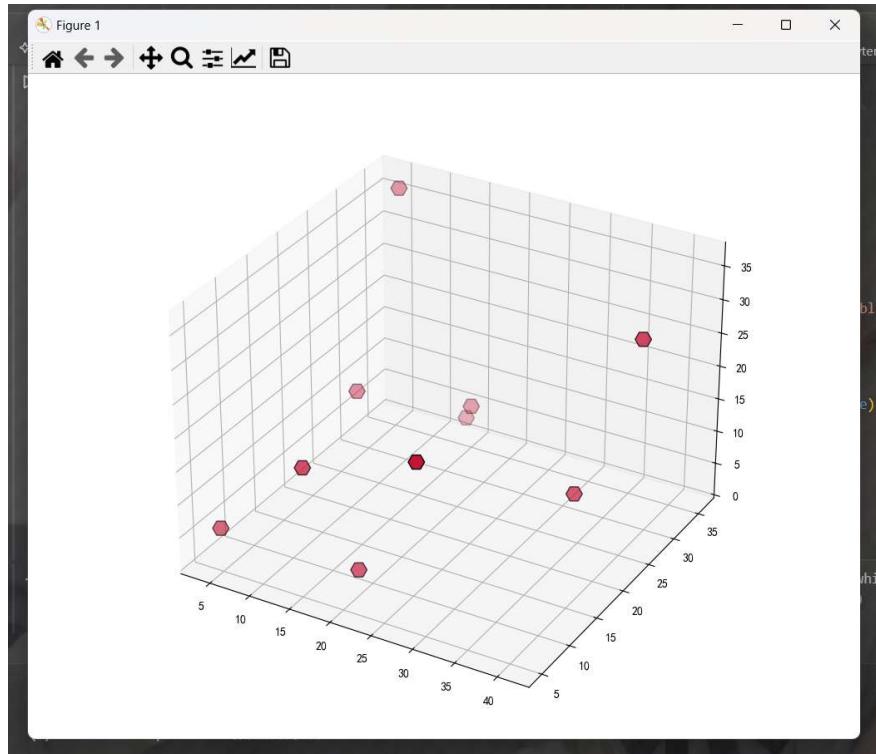


图 3 第一帧动画

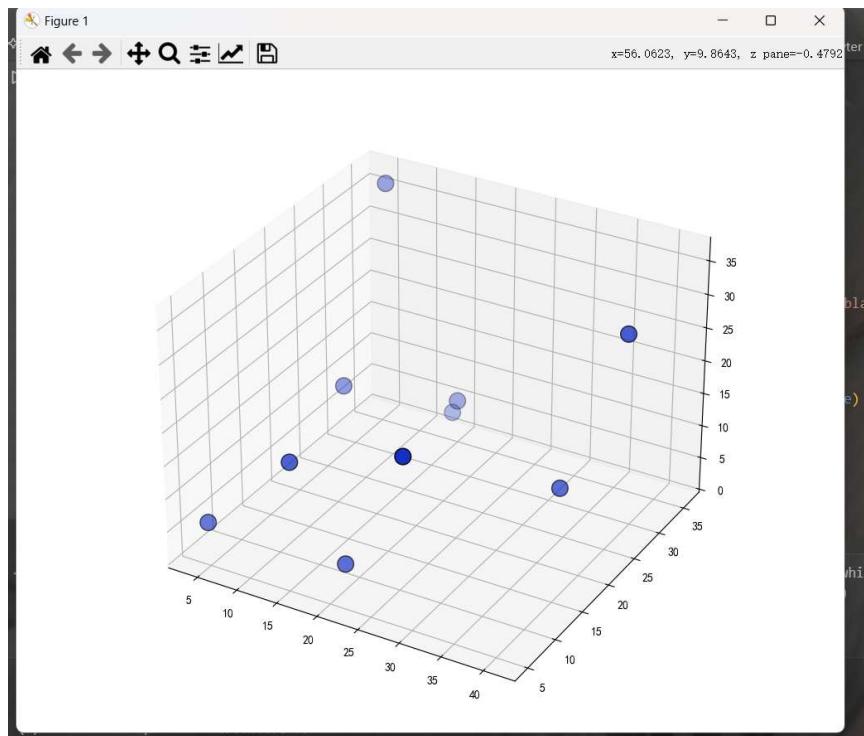


图 4 第二帧动画

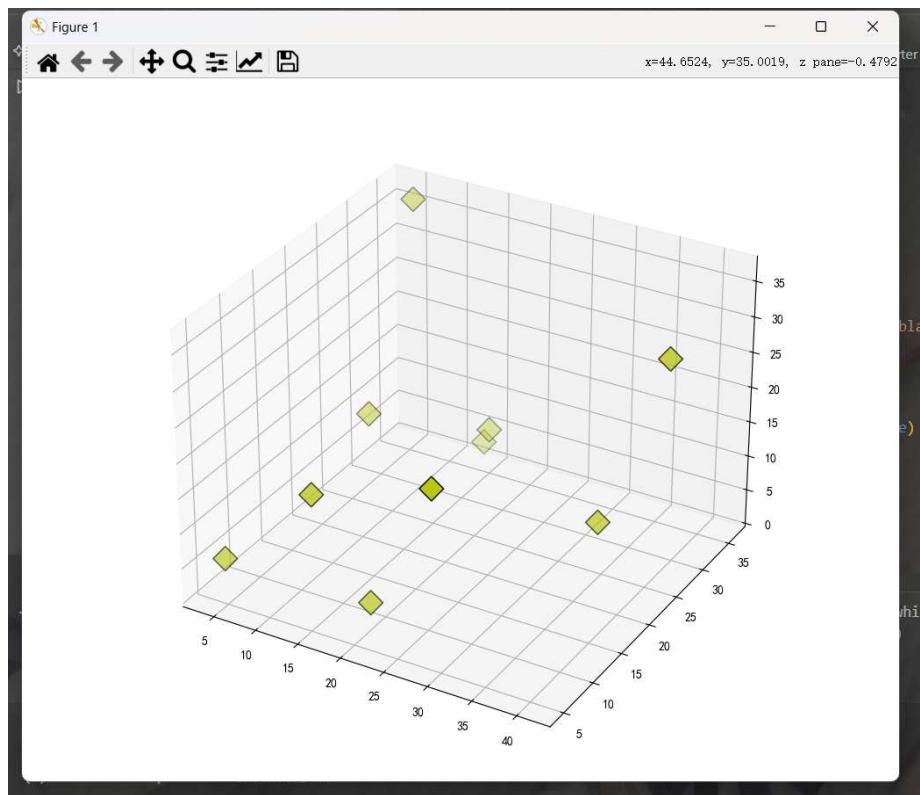


图 5 第三帧动画

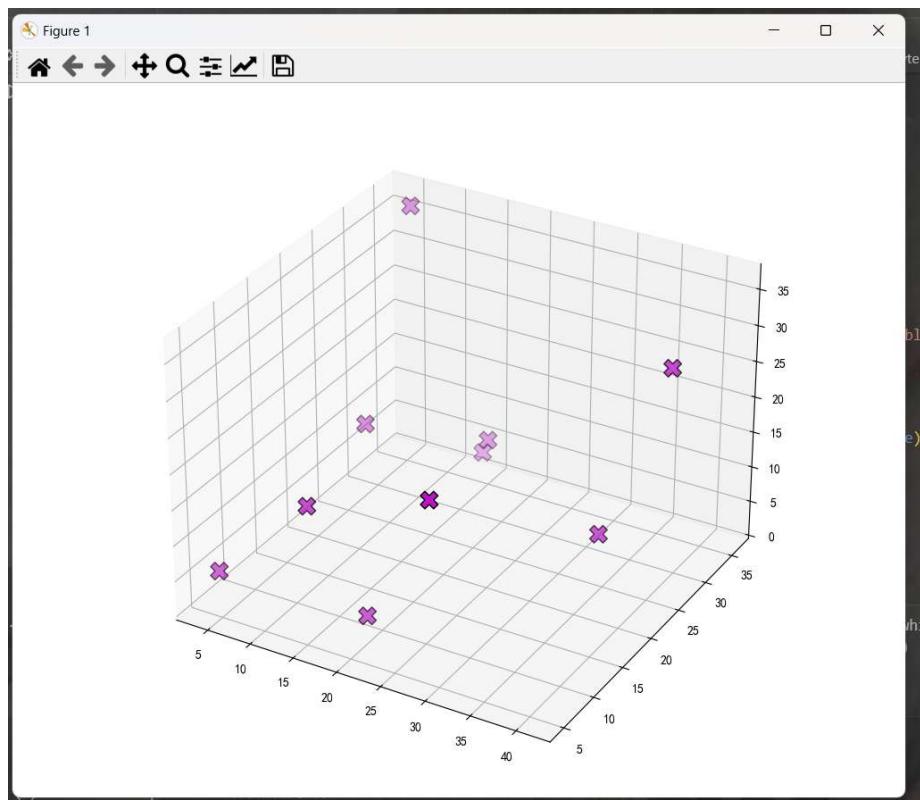


图 6 第四帧动画

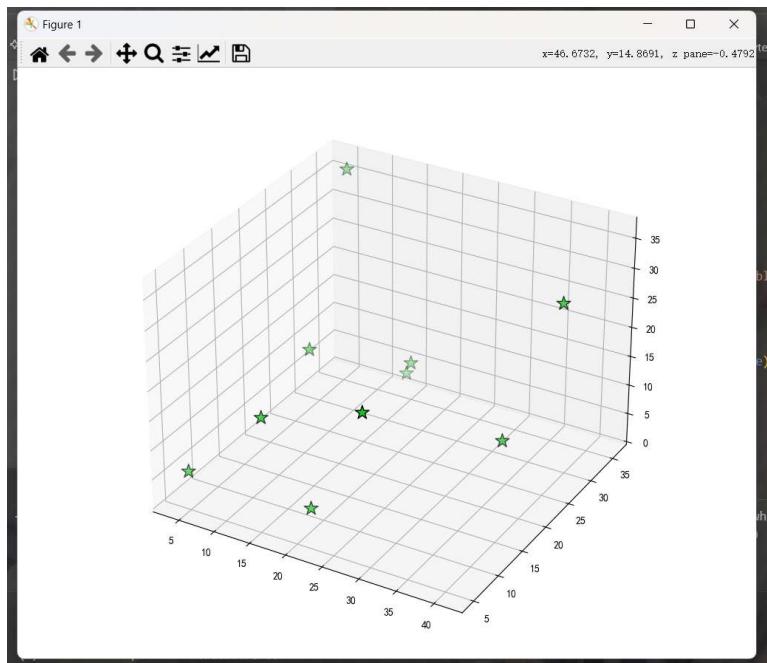


图 7 第五帧动画

说明：一共有 5 帧动画，设置了 10 个点的 xyz 坐标，同样使用 qt5 显示。

4. 利用所给的美国城镇人口分布情况数据绘制所有美国城镇的人口分布情况。要求将人口大于 200 万的城市绘制为红色，大于 50 万但小于 200 万的城市绘制为蓝色，小于 50 万的城市绘制为黄色。

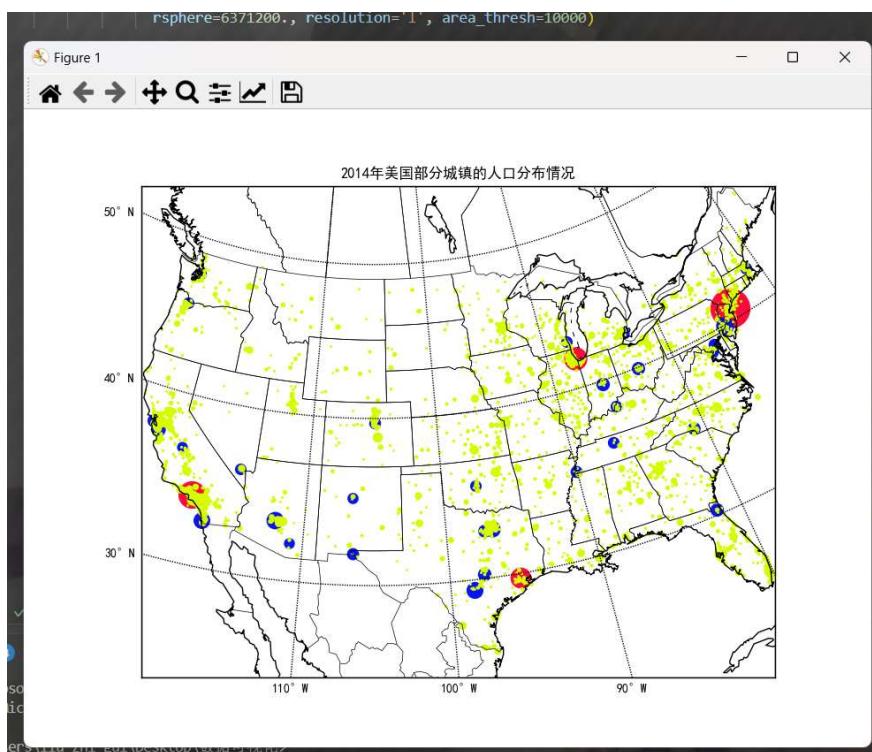
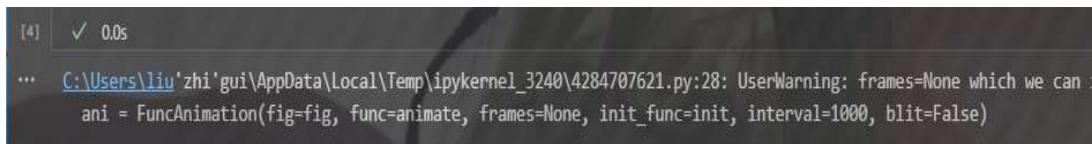


图 8 2014 年美国人口分布

三、思考体会

这次实验主要是进行了 basemap 工具的使用，绘制了 3D 散点图，用循环来实现随机的效果。还用 FuncAnimation 函数实现了动画，把静态散点图转换成了动态的效果。最后绘制了美国 2014 年的人口分布图，这几个图像都采用的 qt5 框架实现，这样可以更加轻松的实现交互效果。

在进行动画代码运行后遇到一个警告：



```
[4]: 0.0s
...
C:\Users\liu'zhi'gui\AppData\Local\Temp\ipykernel_3240\4284707621.py:28: UserWarning: frames=None which we can
ani = FuncAnimation(fig=fig, func=animate, frames=None, init_func=init, interval=1000, blit=False)
```

图 9 警告图

查询得知是因为 Matplotlib 会为了防止缓存无限制增长自动禁用缓存，然后提示警告。只需要在 FuncAnimation 中添加 cache_frame_data=False 参数，明确禁用缓存就行了，如下成功的显示了图像且没有弹出警告，问题解决：

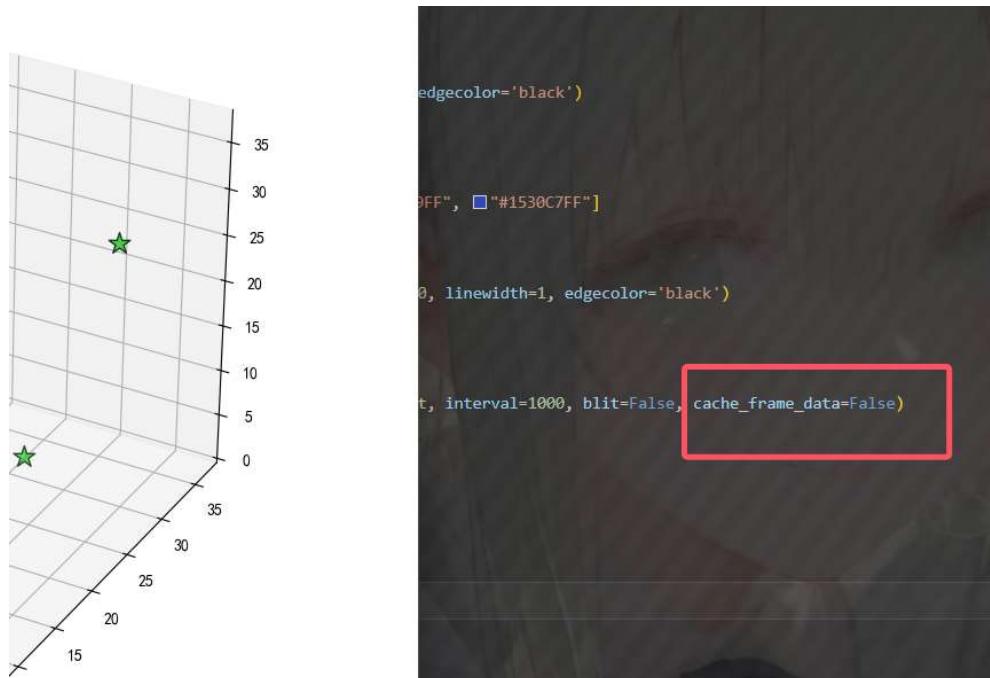


图 10 解决问题图

附录：

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

%matplotlib qt5
plt.rcParams["font.sans-serif"] = ["SimHei"]
plt.rcParams["axes.unicode_minus"] = False

x = np.random.randint(0, 40, 80)
y = np.random.randint(0, 40, 80)
z = np.random.randint(0, 40, 80)

fig = plt.figure(figsize=(8,8))
ax = fig.add_subplot(111, projection='3d')
for xx, yy, zz in zip(x, y, z):
    color = 'y'
    marker='o'
    if 10 < zz < 30:
        color = "#1530C7B9"
        marker='*'
    elif zz >= 30:
        color = "#238800"
        marker='D'
    ax.scatter(xx, yy, zz, c=color, marker=marker, s=160, linewidth=1, edgecolor='black')
ax.set_xlabel('x轴', fontsize=13)
ax.set_ylabel('y轴', fontsize=13)
ax.set_zlabel('z轴', fontsize=13)
ax.set_title('3D散点图', fontproperties='simhei', fontsize=16)
plt.tight_layout()
plt.show()
```

图 11 第二小题代码

```
> >
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from matplotlib.animation import FuncAnimation

%matplotlib qt5
plt.rcParams["font.sans-serif"] = ["SimHei"]
plt.rcParams["axes.unicode_minus"] = False

xx = np.array([15, 8, 5, 19, 4, 17, 13, 29, 37, 41])
yy = np.array([34, 24, 36, 9, 6, 32, 8, 5, 21, 27])
zz = np.array([5, 14, 36, 2, 5, 9, 16, 25, 11, 31])
fig = plt.figure(figsize=(10, 8))
ax = fig.add_subplot(111, projection='3d')

star = ax.scatter(xx, yy, zz, c='k', marker='*', s=200, linewidth=1, edgecolor='black')

@ | 解释 | 添加注释
def animate(i):
    ax.clear()
    colors = [ "#BBC715FF", "#BB15C7FF", "#15C71EFF", "#C71539FF", "#1530C7FF" ]
    markers = [ 'D', 'X', '*', 'H', 'o' ]
    color = colors[i % len(colors)]
    marker = markers[i % len(markers)]
    next_star = ax.scatter(xx, yy, zz, c=color, marker=marker, s = 200, linewidth=1, edgecolor='black')
    return next_star

@ | 解释 | 添加注释
def init():
    return star

ani = FuncAnimation(fig=fig, func=animate, frames=None, init_func=init, interval=1000, blit=False)
ax.set_xlabel('x轴')
ax.set_ylabel('y轴')
ax.set_zlabel('z轴')
ax.set_title('3D散点图', fontproperties='simhei', fontsize=16)
plt.tight_layout()
plt.show()
```

图 12 第三小题代码

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from mpl_toolkits.basemap import Basemap
%matplotlib qt5
plt.rcParams["font.sans-serif"] = ["SimHei"]
plt.rcParams["axes.unicode_minus"] = False

plt.figure(figsize=(10, 8))
map = Basemap(projection='stere', lat_0=90, lon_0=-105, llcrnrlat=23.41,
              urcrnrlat=45.44, llcrnrlon=-118.67, urcrnrlon=-64.52,
              rsphere=6371200., resolution='l', area_thresh=10000)
map.drawmapboundary()
map.drawstates()
map.drawcoastlines()
map.drawcountries()

parallels = np.arange(0., 90, 10.)
map.drawparallels(parallels, labels=[1, 0, 0, 0], fontsize=10)
meridians = np.arange(-110., -60., 10.)
map.drawmeridians(meridians, labels=[0, 0, 0, 1], fontsize=10)
posi = pd.read_csv(r"Data\2014_us_cities.csv")

lat = np.array(posi["lat"])
lon = np.array(posi["lon"])
pop = np.array(posi["pop"], dtype=float)
for i in range(len(lat)):
    size = (pop[i] / np.max(pop)) * 1000
    x, y = map(lon[i], lat[i])
    if pop[i] >=2000000:
        color = "#F0B3BFF"
    elif pop[i] >=500000:
        color = "#0616F5FF"
    else:
        color = "#D7FD00FF"
    map.scatter(x, y, s=size, c=color)
plt.title('2014年美国部分城镇的人口分布情况')
plt.show()

```

图 13 第四小题代码图

3212	Highland Heights ,7153,41.5519954,-81.4784522
3213	Sioux Center ,7148,43.0796915,-96.1756717
3214	Glenolden borough,7143,39.9001126,-75.2890745
3215	Pinson ,7142,33.6889908,-86.6833229
3216	Independence ,7137,37.2242358,-95.7083131
3217	Clewiston ,7134,26.7542312,-80.9336753
3218	Delphos ,7128,40.8433831,-84.341618
3219	University Park village,7127,34.18269725,-117.3339093
3220	Huron ,7122,44.7923065,-82.3311296
3221	Delafield ,7122,43.0610275,-88.4040692
3222	La Junta ,7116,37.9850091,-103.5438321
3223	New Square village,7109,41.1670394,-74.043197
3224	Sullivan ,7104,36.4966673,-82.286903
3225	Spanish Fort ,7102,30.6749127,-87.9152724
3226	Plaquemine ,7102,30.2890833,-91.2342744
3227	Milton-Freewater ,7102,45.9326346,-118.3877435
3228	Benton ,7096,36.0345286,-88.101285
3229	Ocean City ,7094,39.2776156,-74.5746001

图 14 2014 年美国城市人口分布数据