
Algorithm 1: 网格划分

Input: aunt.data, (r,c),aunt.num
Output: aunt.data['district_x'], aunt.data['district_y']

```
1 begin
2   将阿姨或订单的数据、网格参数、阿姨或订单的数量作为参数传入函数
3   eps  $\leftarrow$  10
4   在 X、Y 的最大坐标和最小坐标间生成与网格参数行、列数量相等的数组
5   X_range  $\leftarrow$  np.linspace(X_min, X_max, r+1) Y_range  $\leftarrow$  np.linspace(Y_min, Y_max, c+1)
6   扩大 X、Y 的网格边界
7   X_range[0]  $\leftarrow$  X_range[0] - eps X_range[-1]  $\leftarrow$  X_range[-1] + eps
8   Y_range[0]  $\leftarrow$  Y_range[0] - eps Y_range[-1]  $\leftarrow$  Y_range[-1] + eps
9   dis_x = [ ], dis_y = [ ]
10  判断所有阿姨或订单的位置，并将阿姨或订单的 X、Y 分别存在 dis_x,dis_y 列表中
11  for  $i$  in range(aunt.num) do
12    for  $j$  in range(r) do
13      if X_range[j]  $\leq$  aunt.data['X_od'].iloc[i] < X_range[j+1] then
14        | dis_x.append(j)
15      end if
16    end for
17    for  $k$  in range(c) do
18      if Y_range[k]  $\leq$  aunt.data['Y_od'].iloc[i] < Y_range[k+1] then
19        | dis_y.append(k)
20      end if
21    end for
22  end for
23  将阿姨或订单的数据写入阿姨、订单的区属性里面 aunt.data['district_x']  $\leftarrow$  dis_x
24  aunt.data['district_y']  $\leftarrow$  dis_y
25 end
```

Algorithm 2: 网格划分	
	Input: aunt.data, (r,c),aunt.num
	Output: aunt.data['district_x'], aunt.data['district_y']
1	begin
2	将阿姨或订单的数据、网格参数、阿姨或订单的数量作为参数传入函数
3	eps \leftarrow 10
4	在 X、Y 的最大坐标和最小坐标间生成与网格参数行、列数量相等的数组
5	X_range \leftarrow np.linspace(X_min, X_max, r+1) Y_range \leftarrow np.linspace(Y_min, Y_max, c+1)
6	扩大 X、Y 的网格边界
7	X_range[0] \leftarrow X_range[0] - eps X_range[-1] \leftarrow X_range[-1] + eps
8	Y_range[0] \leftarrow Y_range[0] - eps Y_range[-1] \leftarrow Y_range[-1] + eps
9	dis_x = [], dis_y = []
10	判断所有阿姨或订单的位置，并将阿姨或订单的 X、Y 分别存在 dis_x,dis_y 列表中
11	for 循环阿姨或订单的数量) do
12	for 循环所有行 do
13	if 阿姨或订单的 X 距离在第 i 行和第 i+1 行时 then
14	把阿姨、或订单划分在第 i 行
15	end if
16	end for
17	for 循环所有列 do
18	if 阿姨或订单的 X 距离在第 j 列和第 j+1 列时 then
19	把阿姨、或订单划分在第 j 列
20	end if
21	end for
22	end for
23	将阿姨或订单的数据写入阿姨、订单的区属性里面 aunt.data['district_x'] \leftarrow dis_x
24	aunt.data['district_y'] \leftarrow dis_y
25	end