

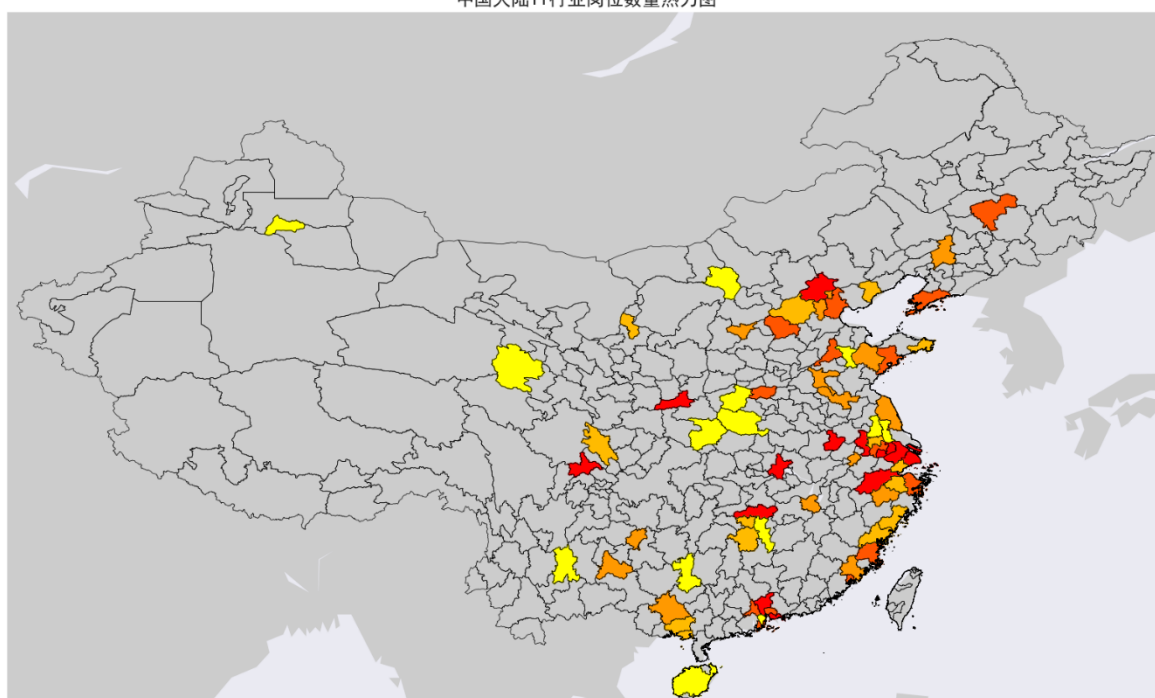
数据可视化代码及结果分析

1 it 工作机会分布情况

IT行业前15城市岗位分布情况



中国大陆IT行业岗位数量热力图

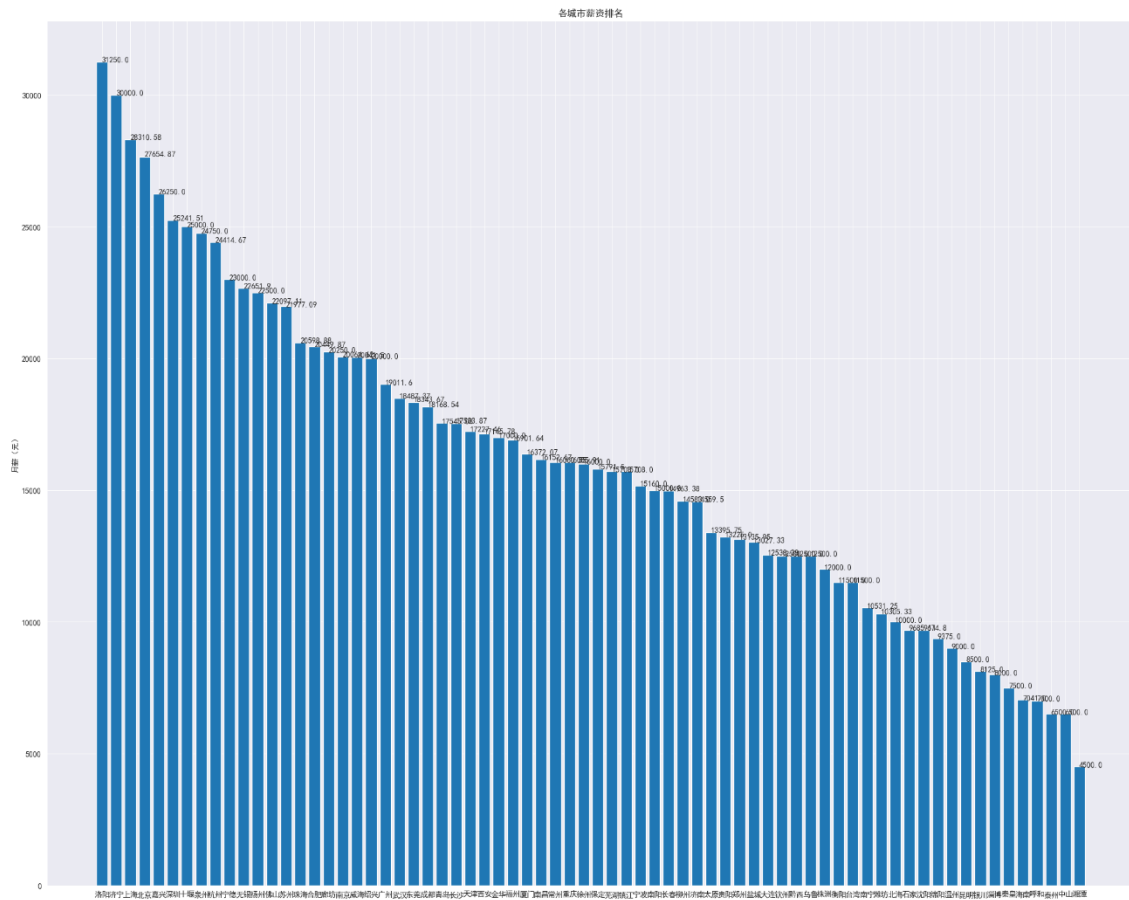


数据显示，IT 行业的工作机会主要聚集在中国的一线城市和新一线城市。这主要得益于一线城市的发达经济和良好的创新环境，以及新一线城市的快速崛起和投资吸引力。

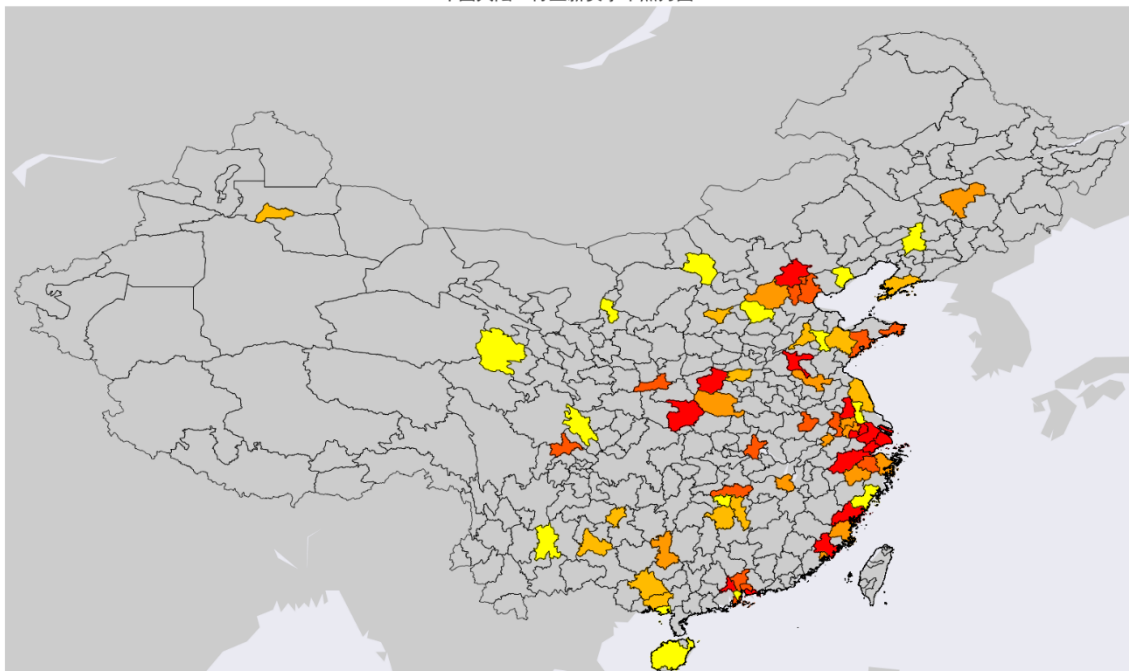
长三角地区（包括上海、杭州、苏州、宁波等城市）和珠三角地区（包括深圳、广州、珠海等城市）已经形成了繁荣的产业集群，成为 IT 行业的重要聚集地。这些地区吸引了大量的科技企业和初创公司，拥有丰富的人才资源、完善的产业链和发达的交通网络。这种集群效应使得这些地区相比其他城市提供了更广阔和多元化的工作机会。

此外，这些城市还受益于优质的教育资源、创新创业氛围以及政府的支持政策。这些因素为 IT 从业者提供了更好的发展平台和职业机会。

2 it 月薪高低分布情况



中国大陆IT行业薪资水平热力图



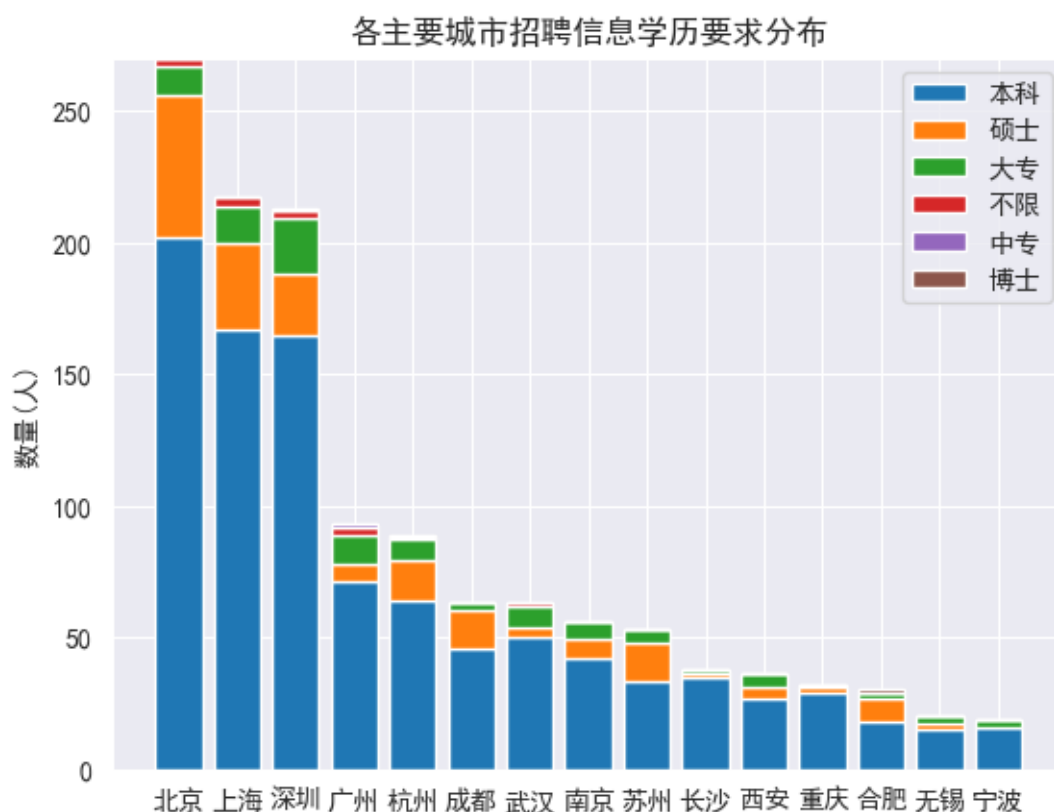
根据数据显示，除了武汉都市圈外，IT 行业的薪资普遍较高的主要集中在沿海城市，特别是长三角和珠三角地区。这可以归因于沿海地区的经济发达、产业集群形成和外资企业的进驻。在长三角和珠三角地区，由于集中了大量的科技企业和创新型公司，对技术人才的需求较高，因此薪资

水平相对较高。这些地区拥有较高的生活成本和竞争激烈的就业市场，因此公司为了吸引和留住优秀的人才，通常提供更具竞争力的薪资待遇。

相比之下，内陆城市的工资水平一般较低。这可能与内陆地区的经济发展相对滞后、产业集中度较低以及相对较少的外资企业投资有关。尽管内陆城市的工作机会可能较多，但由于市场竞争相对较小，公司对薪资的投入也相应较少。

然而，需要指出的是，这仅仅是总体趋势，并不意味着所有沿海城市都具有较高的薪资水平，或者所有内陆城市都具有较低的薪资水平。在具体就业选择时，个人的技能、经验、所在行业和公司的规模等因素也会对薪资产生影响。

3 招聘信息学历要求分布情况与薪资水平的相关性

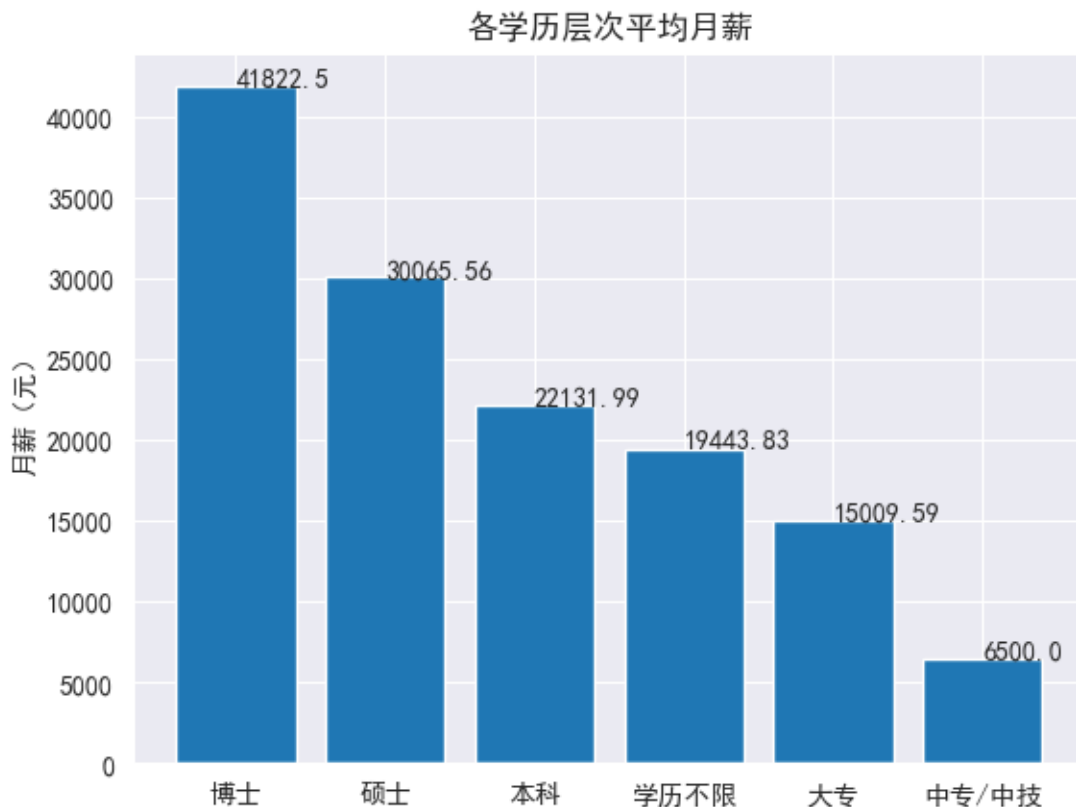


对于 IT 从业者来说，大部分岗位的学历要求为本科，其次是硕士学历和大专学历。本科学历通常是企业招聘计算机相关职位的基本要求，因为它提供了扎实的计算机基础知识和技能。本科课程涵盖了编程、数据结构、算法、数据库管理等核心领域，为学生提供了全面的计算机教育。

硕士学历在 IT 行业也很受欢迎。硕士研究生的学习深入了解计算机科学的前沿技术和研究领域，同时培养了解决复杂问题和开展研究的能力。硕士学历对于一些高级技术职位或研发职位来说具有竞争优势。大专学历在 IT 行业也有一定的需求。大专学历通常注重实际操作技能的培养，例如应用开发、网络管理和数据库操作等。对于一些应用型岗位，大专学历的候选人具备更实际的经验和技能，更容易适应工作环境。

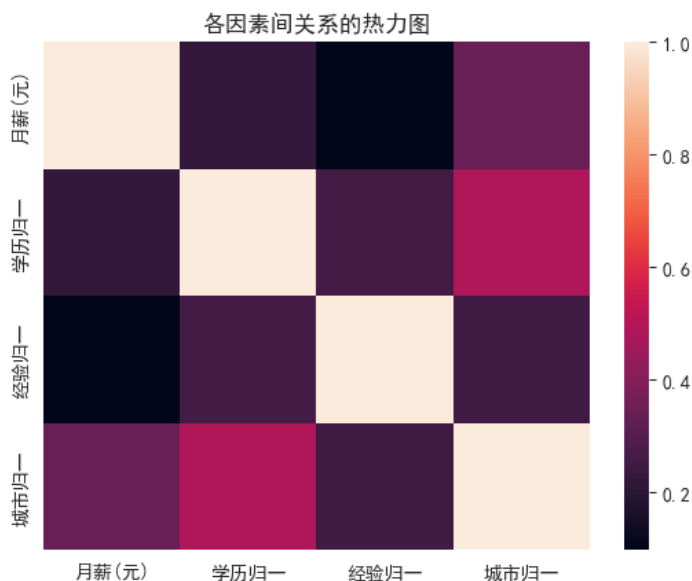
少部分高级技术岗位或研究型岗位对博士学历有要求。博士学位对于从事科学研究、创新和领导层职位的人才培养具有重要作用。

然而，需要注意的是，学历并不是唯一的决定因素，实际工作经验和技能同样重要。在计算机专业中，技术能力、项目经验、解决问题的能力以及不断学习和更新的意愿也是被雇主重视的因素。因此，IT 从业者在提升自身竞争力时，除了学历外，还应注重技术能力的提升和实践经验的积累。



从岗位的分工角度来看，学历对薪资有一定的影响。通常情况下，较高学历的从业者在职位分工上更倾向于担任高级技术职位、管理职位或研发领导职位。这些职位通常要求更深入的技术理解、解决复杂问题的能力以及对行业的前沿知识有更深入的了解。因此，博士学历往往能够获得更高的薪资回报，约为 4 万元左右。硕士学历在这方面也相对较高，约为 3 万元左右。本科学历的从业者通常担任开发、测试、实施等基础岗位，薪资水平约为 2 万元左右。而其他学历的从业者由于学历较低，可能在技术能力和职位层级上有一定限制，因此薪资水平较低，约为 6500 元左右。然而，需要注意的是，薪资也受其他因素的影响，如工作经验、技能水平、行业需求和地域差异等。因此，学历虽然是一个重要因素，但不是决定薪资的唯一因素。

4.2.4 薪资水平与各因素之间的相关性分析



根据各因素间的关系系数分析，我们可以发现薪资与不同属性之间存在一定的相关性。其中，城市是与薪资最相关的属性，表明选择合适的城市对薪资水平有着重要影响。一线城市和新一线城市往往拥有更多的高薪工作机会和发展机会，因为这些地区集聚了更多的科技企业和创新型公司。

其次，学历也与薪资密切相关。学历的提升通常伴随着知识和技能的增长，使个人具备更高级的专业能力和解决问题的能力。因此，博士和硕士学历的从业者在求职过程中往往能够获得更高的薪资回报，相比于本科及以下学历的从业者。

最后，经验对薪资的影响较为次要。虽然经验可以带来实际工作经历和技能积累，但在面对城市选择和学历提升这两个因素时，其影响力相对较小。然而，随着经验的积累，从业者在工作中展现的能力和 value 会得到认可，这有望带来薪资的进一步提升。

数据可视化部分的代码：

```
import pandas as pd
import regex
import seaborn as sns
from matplotlib import pyplot as plt
from mpl_toolkits.basemap import Basemap
from matplotlib.colors import rgb2hex
from matplotlib.patches import Polygon
#载入所需库
```

```
data = pd.read_csv('jobs.csv')
#读取数据
sns.set_style('darkgrid')
plt.rcParams['font.sans-serif']=['SimHei'] #指定中文字体
```

```
data['工作地点'] = data['工作地点'].str[:2]
data.loc[data['学历要求'] == '统招本科', '学历要求'] = '本科'
#数据归一化
```

```
gb_city = data.groupby('工作地点').count()['信息ID']
gb_city
#岗位分布的前置工作
```

```
whole_city_list = gb_city
whole_city_list = whole_city_list.sort_values(ascending = False)
whole_city_list = whole_city_list.index
gb_city = gb_city.sort_values(ascending = False)
gb_city = gb_city[:15]
plt.pie(gb_city, labels = gb_city.index, autopct = '%0.2f%',
        explode = [j*0.01 for j in range(0, len(gb_city))],
        shadow = True
        )
plt.title('IT行业前15城市岗位分布情况')
#IT行业前15城市岗位分布情况
```

```
data['月薪(元)'] = 0
for i in range(0, len(data)):
    j = data['薪资区间'][i]
    lst = regex.findall(r'(\d+)-(\d+).*', j)
    # print(j)
    if(len(lst) >= 1):
        salary = (int(lst[0][0]) + int(lst[0][1])) / 2
    else:
        salary = 0
    if('薪' in j and salary > 0):
        months = int(regex.findall(r'.+(\d\d)薪', j)[0])
        salary = salary * months / 12
    salary = int(salary * 1000)
    data['月薪(元)'][i] = salary
#月薪的归一化
```

```
data[data['月薪(元)' == 0] = data['月薪(元)'].mean()
#把面谈工资的月薪设定为均值
```

```
city_salary = data.groupby('工作地点').mean()
#统计各地平均月薪
```

```
city_salary = city_salary['月薪(元)']
city_salary = city_salary.sort_values(ascending = False)
```

```
for i in range(0, len(city_salary)):
    if(type(city_salary.index[i]) != type('test')):
        city_salary = city_salary.drop(city_salary.index[i])
#清洗异常值
```

```
def add_value_label(x_list, y_list):
    for i in range(0, len(x_list)):
        plt.text(i, round(y_list[i], 2), round(y_list[i], 2))
#这是给每个条柱赋值的字段
```

```
plt.figure(figsize=(25, 20)) #画布大小
plt.bar(city_salary.index, city_salary)
plt.ylabel('月薪 (元)')
add_value_label(city_salary.index, city_salary)
plt.title('各城市薪资排名')
#绘制各城市薪资排名
```

```
degree_salary = data.groupby('学历要求').mean()
#统计各学历层次月薪
```

```
for i in range(0, len(degree_salary)):
    if(type(degree_salary.index[i]) != type('test')):
        degree_salary = degree_salary.drop(degree_salary.index[i])
```

```
degree_salary = degree_salary['月薪(元)'].sort_values(ascending = False)
# for i in range(0, len(degree_salary)):
#     if(type(degree_salary.index[i]) != type('test')):
#         degree_salary = degree_salary.drop(degree_salary.index[i])
```

```
degree_salary = round(degree_salary, 2)
degree_salary
```

```
city_list = gb_city.index
data_city = data[data['工作地点'] == '北京']
for i in range(1, len(city_list)):
    data_city = pd.concat([data_city, data[data['工作地点'] == city_list[i]]])
```

```
data_degree = []
data_degree.append(data_city[data_city['学历要求'] == '本科'])
data_degree.append(data_city[data_city['学历要求'] == '硕士'])
data_degree.append(data_city[data_city['学历要求'] == '大专'])
data_degree.append(data_city[data_city['学历要求'] == '学历不限'])
data_degree.append(data_city[data_city['学历要求'] == '中专/中技'])
data_degree.append(data_city[data_city['学历要求'] == '博士'])
```

```
for i in range(0, len(data_degree)):
    data_degree[i] = data_degree[i].groupby('工作地点').count()['信息ID']
    data_degree[i] = data_degree[i].reindex(city_list, fill_value = 0)
```

```
bottom = data_degree[0]
plt.bar(data_degree[0].index, data_degree[0], label = ['本科', '硕士', '大专', '不限', '中专', '博士'][0])
plt.ylabel('数量(人)')
for i in range(1, 6):
    plt.bar(data_degree[i].index, data_degree[i], bottom = bottom, label = ['本科', '硕士', '大专', '不限', '中专', '博士'][i])
    bottom = bottom + data_degree[i]
plt.legend()
plt.title('各主要城市招聘信息学历要求分布')
#绘制各主要城市招聘信息学历要求分布
```

```
whole_city_list
colors_hash = dict()
for i in range(0, len(whole_city_list)):
    j = int(i / (len(whole_city_list) / 5))
    hash_ = ['#ff0000', '#ff5500', '#ff9900', '#ffbb00', '#ffff00']
    colors_hash.update({whole_city_list[i]:hash_[j]})
#颜色的前置准备
```

```

plt.figure(figsize=(20, 15))
map = Basemap(llcrnrlon=73, llcrnrlat=18, urcrnrlon=135, urcrnrlat=55)
map.fillcontinents()
map.readshapefile('gadm36_TWN_1', 'states', drawbounds = True)
map.readshapefile('gadm36_CHN_2', 'states', drawbounds = True)

map_city_list = map.states_info
map_city_list_2 = []
already = []
count = 0

colors_2 = []
for i in map_city_list:
    if i['NL_NAME_2'] in whole_city_list:
        map_city_list_2.append(count)
        colors_2.append(colors_hash[i['NL_NAME_2']])
    if i['NL_NAME_2'][:2] in whole_city_list:
        map_city_list_2.append(count)
        k = i['NL_NAME_2'][:2]
        colors_2.append(colors_hash[k])
    count = count + 1

ax = plt.gca()
for i in range(0, len(map_city_list_2)):
    j = map_city_list_2[i]
    color = rgb2hex(colors_2[i])
    poly = Polygon(map.states[j], facecolor=color, edgecolor=color)
    ax.add_patch(poly)

```

```

ax.set_title(label='中国大陆IT行业岗位数量热力图',
             fontdict={'fontsize':18})
plt.show()
#绘制中国大陆IT行业岗位数量热力图

```

```

city_salary_list
colors_hash = dict()
for i in range(0, len(city_salary_list)):
    j = int(i / (len(city_salary_list) / 5))
    hash_ = ['#ff0000', '#ff5500', '#ff9900', '#ffbb00', '#ffff00']
    colors_hash.update({city_salary_list[i]:hash_[j]})

```

#颜色的前置准备

```

plt.figure(figsize=(20, 15))
map = Basemap(llcrnrlon=73, llcrnrlat=18, urcrnrlon=135, urcrnrlat=55)
map.fillcontinents()
map.readshapefile('gadm36_TWN_1', 'states', drawbounds = True)
map.readshapefile('gadm36_CHN_2', 'states', drawbounds = True)

map_city_list = map.states_info
map_city_list_2 = []
already = []
count = 0

# colors = {s:cmap(np.sqrt((GDP[s] - GDP.min()) / (GDP.max() - GDP.min()))[:3])
#           for s in GDP.index}

colors_2 = []
for i in map_city_list:
    if i['NL_NAME_2'] in whole_city_list:
        map_city_list_2.append(count)
        colors_2.append(colors_hash[i['NL_NAME_2']])
    if i['NL_NAME_2'][:2] in whole_city_list:
        map_city_list_2.append(count)
        k = i['NL_NAME_2'][:2]

```



```

        colors_2.append(colors_hash[KJ])
        count = count + 1

ax = plt.gca()
for i in range(0, len(map_city_list_2)):
    j = map_city_list_2[i]
    color = rgb2hex(colors_2[i])
    poly = Polygon(map.states[j], facecolor=color, edgecolor=color)
    ax.add_patch(poly)

ax.set_title(label='中国大陆IT行业薪资水平热力图',
             fontdict={'fontsize':18}
             )

#导入地图文件
plt.show()

#地图热力图绘制

```

```

data_corr = data
data_corr['学历归一'] = 0
data_corr['经验归一'] = 0
data_corr['城市归一'] = 0

data_corr.loc[data['学历要求'] == '学历不限', '学历归一'] = 1
data_corr.loc[data['学历要求'] == '中专/中技', '学历归一'] = 2.5
data_corr.loc[data['学历要求'] == '大专', '学历归一'] = 4
data_corr.loc[data['学历要求'] == '本科', '学历归一'] = 6
data_corr.loc[data['学历要求'] == '硕士', '学历归一'] = 8
data_corr.loc[data['学历要求'] == '博士', '学历归一'] = 10

data_corr.loc[data['工作经验'] == '经验不限', '经验归一'] = 2.5
data_corr.loc[data['工作经验'] == '1-3年', '经验归一'] = 5
data_corr.loc[data['工作经验'] == '3-5年', '经验归一'] = 7.5
data_corr.loc[data['工作经验'] == '5-10年', '经验归一'] = 10

for i in range(0, len(city_salary_list)):
    j = 10 - (int(i / (len(city_salary_list) / 10)))
    data_corr.loc[data['工作地点'] == city_salary_list[i], '城市归一'] = j

```

```
data_corr = data_corr.drop(axis=1, columns='信息ID')
```

```
aaa = data_corr.corr()
```

```
aaa =aaa.drop(axis=1, columns='信息ID')
```

```

sns.heatmap(aaa)
plt.title('各因素间关系的热力图')

```