data_map_Chinese_version

2021 年 11 月 25 日

本设计方案基于 Jupyter lab 平台，使用 Python 语言，对烟台市各区市范围内各停车场、充电站的充电桩数据进行整理分析。通过数据的分析和设计模型算法，对莱山区电动桩的未来发展提供设计方案。在考虑充电桩的使用效率基础上，方案详细给出了现今、2025、2030 年充电桩的安装数量及安装位置。
本报告直接由 Notebook ipynb 文件生成。

1 烟台地区各区县充电桩数据分析

烟台市是我国一个地级市，全市土地面积约 13745 平方公里，由 5 个市辖区（芝罘区、福山区、牟平区、莱山区、蓬莱区），6 个县级市组成（龙口市、莱阳市、莱州市、招远市、栖霞市、海阳市），截至 2020 年 11 月，常住人口约为 710 万人。

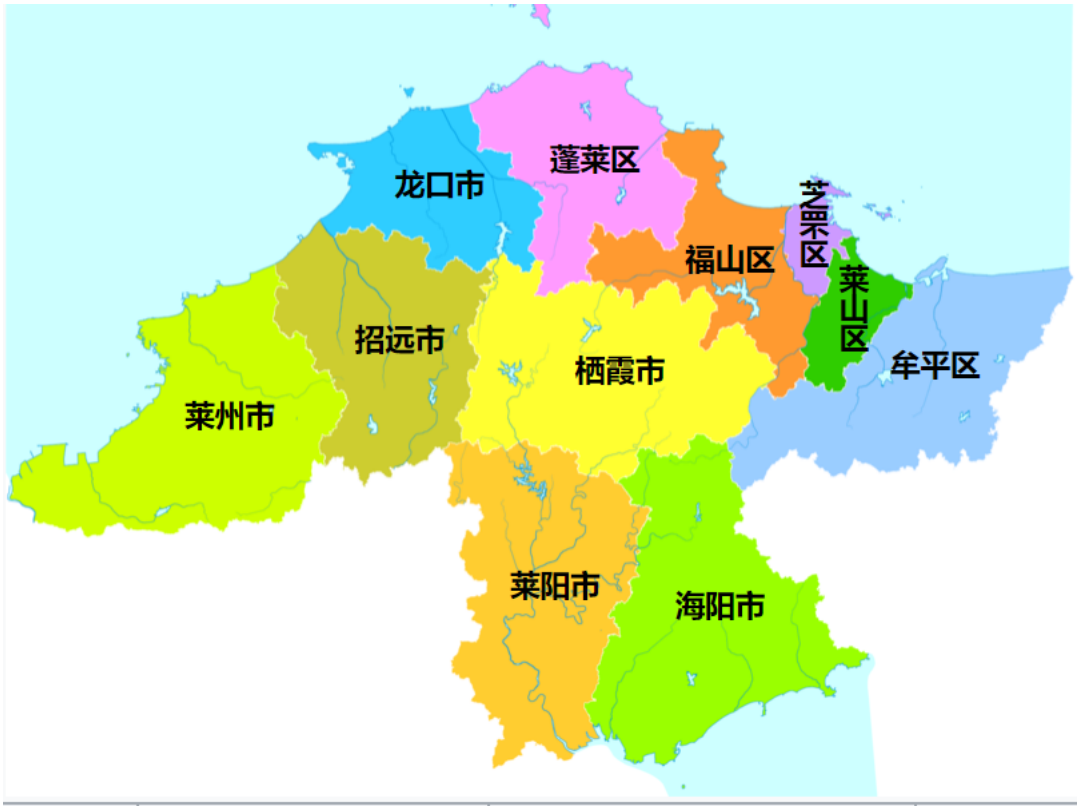


图 1.1 烟台市分区图

已有数据提供了各区市的充电站点经纬度、充电桩数量、充电次数、以及充电量的基础数据。借助 folium 地图包，展示烟台市各分区，并在地图上标记各区市的电动桩数据，非常方便实现充电站定位，各区域充电桩数量热图展示。地图包的导入和展示如下所示：

[2]:
```
# 导入地图模型包 folium、数据分析包 pandas
import folium
import pandas as pd
```

[3]:
```
# 定义世界地图
world_map = folium.Map()
# 烟台经纬度定位
latitude = 37.45
longitude = 121.43
# 创建烟台市地图
Yantai_map = folium.Map(location=[latitude, longitude], zoom_start=12)
# 显示烟台市地图
Yantai_map
```

[3]: <folium.folium.Map at 0x1cfdf834ee0>

图 1.2 烟台市 folium 地图显示

先加载原始数据表 1 和表 2（表 1 烟台地区现有充电桩信息，表 2 烟台地区充电站充电次数及充电量信息）。分析数据前，通过 pandas 包的 merge 功能来合并表 1 和表 2，并对数据进行简单处理，数据处理过程代码见 data-processing.ipynb 文件，并将合并后的文件存储为 data_emerge_all.csv。表 1 中的相同充电站被合并，并统计各充电站充电桩数量，从而得到了分析所需的基本数据格式需求。

[4]:
```
# 加载原始数据表 1 和表 2
data_1= pd.read_excel("D:
↪\\OneDrive\\02_Programs\\2021-Data-competition-Shangdong\\烟台 1-创意赛-城市新能源车充电桩安装位置规划\\表 1 烟台地区现有充电桩信息.xlsx")
data_2 = pd.read_excel("D:
↪\\OneDrive\\02_Programs\\2021-Data-competition-Shangdong\\烟台 1-创意赛-城市新能源车充电桩安装位置规划\\表 2 烟台地区充电站充电次数及充电量信息.xlsx")
# pd.merge(data_2, data_1).to_csv('data_emerge_all.csv', encoding='utf_8_sig')
```

加载 data_emerge_all.csv 数据文件

[5]:
```
data = pd.read_csv("D:
↪\\OneDrive\\02_Programs\\2021-Data-competition-Shangdong\\data_emerge_all.
↪csv")
```

通过各充电站的经纬度信息，利用地图信息，识别出各充电站所属的工作区，并添加到 data_emerge_all.csv 文件中，即为 data 中 district 列。

[6]:
```
# get the district name
import requests
import json
district_all = pd.Series()
for lat, lgn in zip(data.latitude, data.longitude):
    key = 'GjG3XAdmywz7CyETWqHwIuEC6ZExY6QT'
    r = requests.get(url='http://api.map.baidu.com/geocoder/v2/',␣
 ↪params={'location':f'{lat},{lgn}','ak':key,'output':'json'})
    result = r.json()
    # print(result)
    province = result['result']['addressComponent']['province']
    city = result['result']['addressComponent']['city']
    district = result['result']['addressComponent']['district']
    # print(district)
```

```
<ipython-input-6-939eb5b84485>:4: DeprecationWarning: The default dtype for
empty Series will be 'object' instead of 'float64' in a future version. Specify
a dtype explicitly to silence this warning.
  district_all = pd.Series()
```

下面显示了处理好后的数据结构。为了编程所需，英文说明如下:
Station_name: 充电站名称，district：所属区市，year_month: 数据记录月份，power_consum（kWh）: 充电量，times_charging：充电次数，Number_charging_piles：充电桩数量 latitude：维度，longitude：经度，Unit_per_Cha_time：单次充电量（kWh），Unit_per_Cha_pile：每个充电桩月均充电量（kWh）month_use_frequency_per_pile：每个充电桩月均使用次数。

[7]:
```
# 显示前五行数据
data.head()
```

[7]:

	Unnamed: 0	Station_name	district	year_month	power_consum（kWh）
0	0	烟台高新区公交场站	莱山区	202109	147737.44
1	1	烟台泓澳新能源充电站	蓬莱市	202109	135702.52
2	2	烟台莱山长宁路充电站	莱山区	202109	125802.78
3	3	烟台康和新城公交场站	芝罘区	202109	120385.80

```
4           4  山东省烟台市开发区天马相城公交充电站       福山区       202109
↪   117562.77
```

	times_charging	Number_charging_piles	latitude	longitude
0	1857	17	37.411116	121.530138
1	3621	12	37.784622	120.764465
2	5098	25	37.447031	121.438939
3	1948	9	37.508728	121.323249
4	1236	16	37.580740	121.175610

	Unit_per_Cha_time	Unit_per_Cha_pile	month_use_frequency_per_pile
0	79.56	8690.44	109
1	37.48	11308.54	302
2	24.68	5032.11	204
3	61.80	13376.20	216
4	95.12	7347.67	77

[8]:
```
# 删除数据存在空值的行
data.dropna(inplace=True)
```

利用 folium 的标记功能，比较不同各充电站在地图上所处的区域，标记上的数字大小代表所在区域拥有的充电站数量。

[9]:
```
# 导入 Plugins 模块
from folium import plugins
# 标记实例化
incidents =plugins.MarkerCluster().add_to(Yantai_map)
# 循环添加各充电站的标记
for lat, lng, label, in zip(data.latitude, data.longitude, data.
↪Number_charging_piles):
    folium.Marker(
        location=[lat, lng],
        icon=None,
        popup=label,
    ).add_to(incidents)

# 把标记加入到地图上
```

```
Yantai_map.add_child(incidents)
```

[9]: <folium.folium.Map at 0x1cfdf834ee0>

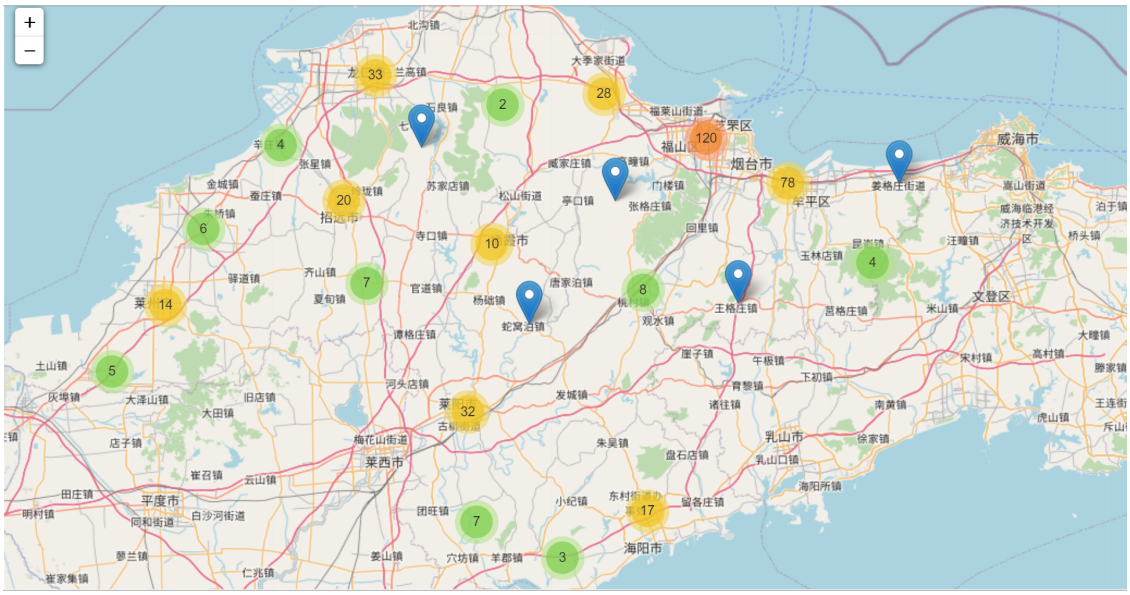


图 1.3 烟台市各充电站标记

由于各充电站拥有的充电桩数量不同，如果把充电桩数量多少显示在标记图中将更加直观。以下代码通过 folium 的 CircleMarker 功能可实现该目的。

[10]:
```
# Instantiate a feature group for the incidents in the dataframe
incidents = folium.map.FeatureGroup()

# Loop through the 200 crimes and add each to the incidents feature group
for lat, lng, nums in zip(data.latitude, data.longitude, data.
 ↪Number_charging_piles):
    incidents.add_child(
        folium.CircleMarker(
            [lat, lng],
            radius=nums, # define how big you want the circle markers to be
            color='yellow',
            fill=True,
            fill_color='red',
            fill_opacity=0.4,
            tooltip = nums
```

```
        )
    )

# Add incidents to map
Yantai_map_piles_num = folium.Map(location=[latitude, longitude], zoom_start=12)
Yantai_map_piles_num.add_child(incidents)
```

[10]: <folium.folium.Map at 0x1cfe050a2e0>

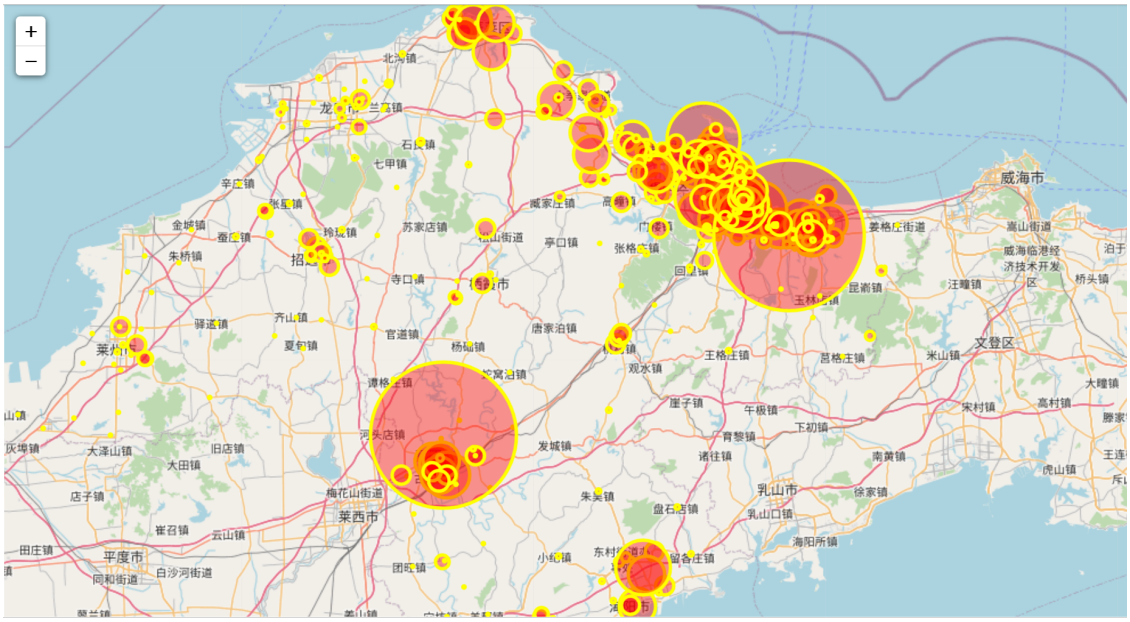


图 1.4 烟台市各充电站充电桩数量标记

接下来对烟台各区市进行边界标记，实行代码如下。

[11]:
```
import json
import requests
def basic_map():
    Yantai_geo = 'city_geojson-master/city_geojson-master/geojsons/Yantai.json'
    basic_map = folium.Map(location = [latitude, longitude], zoom_start = 12)
    folium.GeoJson(
        data = (open(Yantai_geo, "r", encoding="utf-8-sig")).read(),
        style_function=lambda feature: {
            'fillColor': '#ffff00',
            'color': 'black',
```

```
            'weight': 2,
            'dashArray': '5, 5'
            }
    ).add_to(basic_map)
    return basic_map
#display map
basic_map()
```

[11]: <folium.folium.Map at 0x1cfe04a12b0>

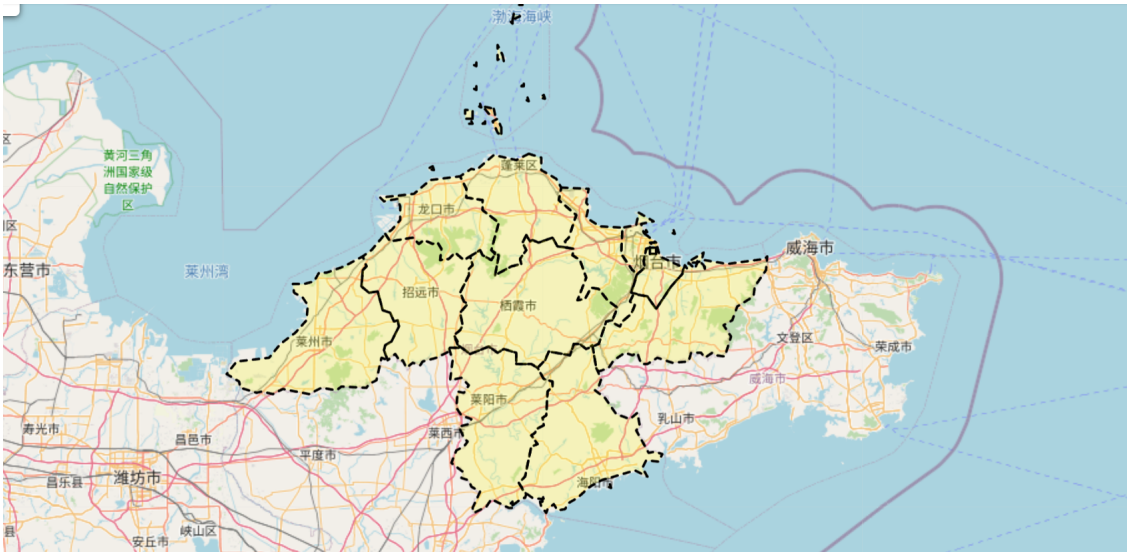


图 1.5 烟台市区市边界线

生成烟台市充电桩数量热图，红色颜色越深代表该区域充电桩数量越多。实现代码如下：

[12]:
```
from folium.plugins import HeatMap

# 准备充电桩数量数据
heatdata = data[['latitude','longitude','Number_charging_piles']].values.
 ↪tolist()

# 把数量信息添加到地图
basic_heatMap = basic_map()
HeatMap(heatdata).add_to(basic_heatMap)
```

```
basic_heatMap
```

[12]: <folium.folium.Map at 0x1cfe1807a90>

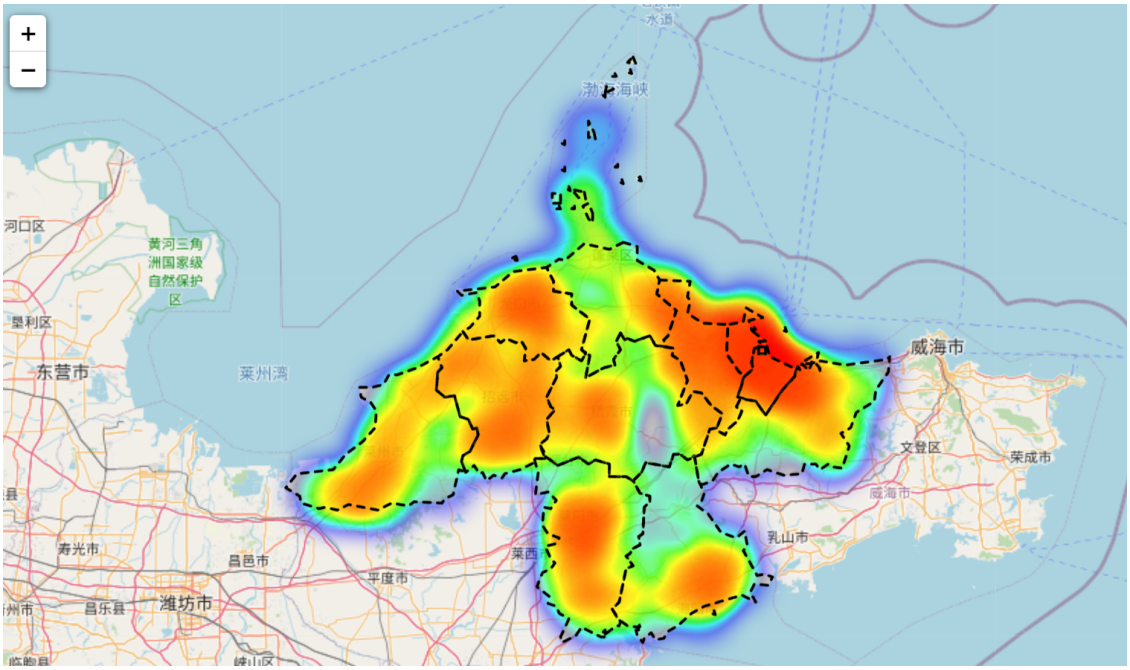


图 1.6 烟台市充电桩数量热图

在地图上标记烟台市所有充电桩位置信息，可以看出不同区域充电站密集度，实现代码如下：

[13]:
```
# 构建特征图
incidents = folium.map.FeatureGroup()
# 添加经纬度定位信息
for lat, lng, name in zip(data.latitude, data.longitude, data.Station_name):
    incidents.add_child(
        folium.CircleMarker(
            [lat, lng],
            radius=7, # define how big you want the circle markers to be
            color='yellow',
            fill=True,
            fill_color='red',
            fill_opacity=0.4,
            popup=name
        )
```

```
    )
# 展示定位地图
basic_map().add_child(incidents)
```

[13]: <folium.folium.Map at 0x1cfe185b940>

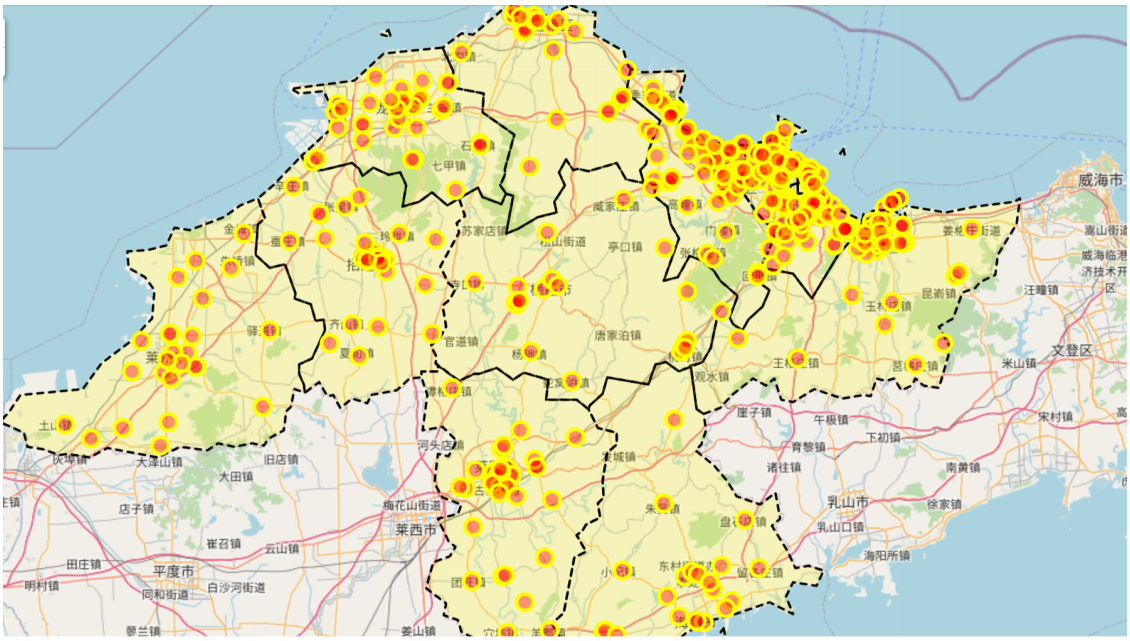


图 1.7 烟台市充电桩站位置信息

通过 head 函数，展示烟台市部分充电站汇总信息，如下：

[14]:
```
data.head()
```

[14]:

	Unnamed: 0	Station_name	district	year_month	power_consum（kWh）
0	0	烟台高新区公交场站	莱山区	202109	147737.44
1	1	烟台泓澳新能源充电站	蓬莱市	202109	135702.52
2	2	烟台莱山长宁路充电站	莱山区	202109	125802.78
3	3	烟台康和新城公交场站	芝罘区	202109	120385.80
4	4	山东省烟台市开发区天马相城公交充电站	福山区	202109	117562.77

```
   times_charging  Number_charging_piles   latitude   longitude  \
0            1857                     17  37.411116  121.530138
1            3621                     12  37.784622  120.764465
2            5098                     25  37.447031  121.438939
3            1948                      9  37.508728  121.323249
4            1236                     16  37.580740  121.175610

   Unit_per_Cha_time  Unit_per_Cha_pile  month_use_frequency_per_pile
0              79.56            8690.44                           109
1              37.48           11308.54                           302
2              24.68            5032.11                           204
3              61.80           13376.20                           216
4              95.12            7347.67                            77
```

本项目是为莱山区充电桩再规划进行设计，所以对各区市充电站数据进行归类及汇总，实现代码如下。

[15]:
```
import numpy as np
data_laishan = data[data.district=='莱山区']
data_muping = data[data.district=='牟平区']
data_zhifu = data[data.district=='芝罘区']
data_fushan = data[data.district=='福山区']
data_penglai = data[data.district=='蓬莱市']
data_longkou = data[data.district=='龙口市']
data_zhaoyuan = data[data.district=='招远市']
data_qixia = data[data.district=='栖霞市']
data_haiyang = data[data.district=='海阳市']
data_laiyang = data[data.district=='莱阳市']
data_laizhou = data[data.district=='莱州市']
data_laishan.head()
```

[15]:
```
   Unnamed: 0   Station_name district  year_month  power_consum（kWh）  \
0           0     烟台高新区公交场站      莱山区      202109          147737.44
2           2    烟台莱山长宁路充电站       莱山区      202109          125802.
↪78
```

```
13          13  山东省烟台市莱山光华充电站      莱山区      202109          ␣
↪64413.98
14          14      烟台朱堵堡公交场站      莱山区      202109          58316.32
16          16    烟台上海滩花园公交场站      莱山区      202109          51615.
↪61
```

	times_charging	Number_charging_piles	latitude	longitude
0	1857	17	37.411116	121.530138
2	5098	25	37.447031	121.438939
13	3201	8	37.467384	121.443810
14	2191	7	37.351410	121.350403
16	968	8	37.440314	121.462538

	Unit_per_Cha_time	Unit_per_Cha_pile	month_use_frequency_per_pile
0	79.56	8690.44	109
2	24.68	5032.11	204
13	20.12	8051.75	400
14	26.62	8330.90	313
16	53.32	6451.95	121

充电桩的使用频率体现了充电桩利用情况，根据原始数据提供的 2021 年 9 月份的充电数据，可绘制各区市的充电桩平均使用次数 (month_use_frequency_per_pile) 的箱线图。实现代码如下。中间橙色线为中位数线，橙色线上、下分别为上、下四分位线，上、下四分位线上方为上限，下方为下限，圆圈为异常值点。从箱线图对比其它区市可以看出，莱山区中位数线较低，而四分位线不低，说明莱山区存在较多充电桩利用率较低，而部分充电桩存在使用拥挤的情形，即存在充电站位置及数量可优化的空间。

[16]:
```
# 保存各区数据
data_all_districts = [data_laishan, data_muping, data_zhifu, data_fushan,␣
↪data_penglai, data_longkou, data_zhaoyuan, data_qixia, data_haiyang,␣
↪data_laiyang,data_laizhou]
data_boxplot = [0]*np.size(data_all_districts)
for i in range(np.size(data_all_districts)):
    data_boxplot[i] = data_all_districts[i].
↪drop(data_all_districts[i][data_all_districts[i]['Unit_per_Cha_time']==0].
↪index).month_use_frequency_per_pile
```

```
C:\Users\admin\anaconda3\lib\site-packages\numpy\core\fromnumeric.py:3208:
VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences
(which is a list-or-tuple of lists-or-tuples-or ndarrays with different lengths
or shapes) is deprecated. If you meant to do this, you must specify
'dtype=object' when creating the ndarray.
  return asarray(a).size
```

[59]:
```
# 绘制箱线图
import matplotlib.pyplot as plt
district_name =␣
 ↪['laishan','muping','zhifu','fushan','penglai','longkou','zhaoyuan','qixia','haiyang','laiy
plt.figure(figsize=(12,6))
plt.boxplot(data_boxplot, labels=district_name)
plt.xlabel('区市名')
plt.ylabel('充电桩月平均使用次数（次）')
```

[59]: Text(0, 0.5, '充电桩月平均使用次数（次）')

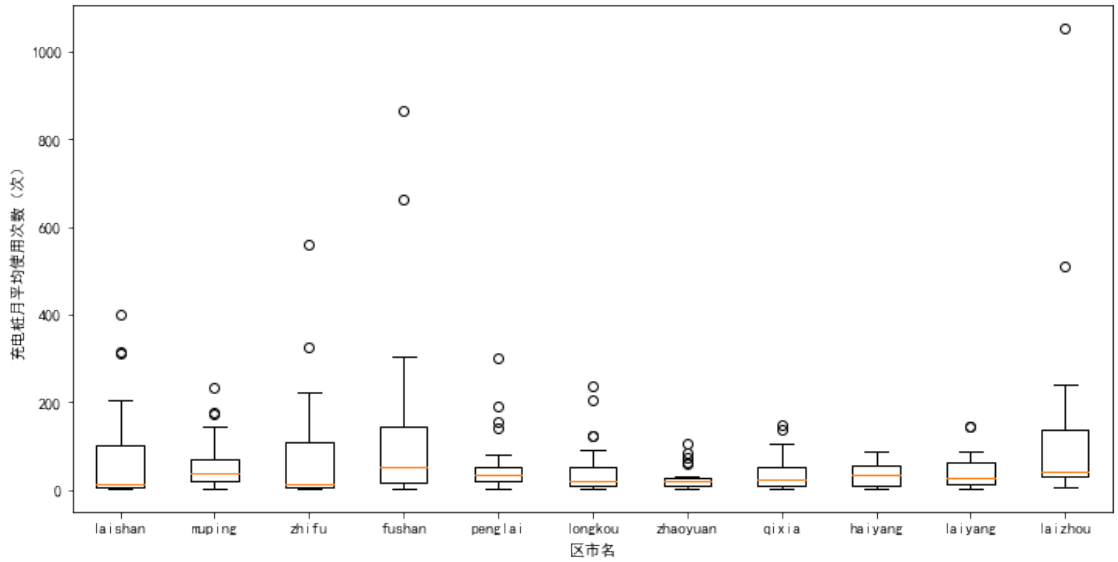


图 1.8 各区市充电桩月使用频率箱线图

同样，用箱线图绘制各区市每次充电量情况。实现代码如下。

[18]:
```
# 删除充电量为零的充电站
data_power_per_char_time = [0]*np.size(data_all_districts)
for i in range(np.size(data_all_districts)):
    data_power_per_char_time[i] = data_all_districts[i].
 ↪drop(data_all_districts[i][data_all_districts[i]['Unit_per_Cha_time']==0].
 ↪index).Unit_per_Cha_time
```

```
C:\Users\admin\anaconda3\lib\site-packages\numpy\core\fromnumeric.py:3208:
VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences
(which is a list-or-tuple of lists-or-tuples-or ndarrays with different lengths
or shapes) is deprecated. If you meant to do this, you must specify
'dtype=object' when creating the ndarray.
  return asarray(a).size
```

[60]:
```
# 绘制单次充电量
plt.figure(figsize=(12,6))
plt.boxplot(data_power_per_char_time, labels=district_name)
plt.xlabel('区市名')
plt.ylabel('单次充电量 (kWh)')
```

[60]:
```
Text(0, 0.5, ' 单次充电量 (kWh)')
```

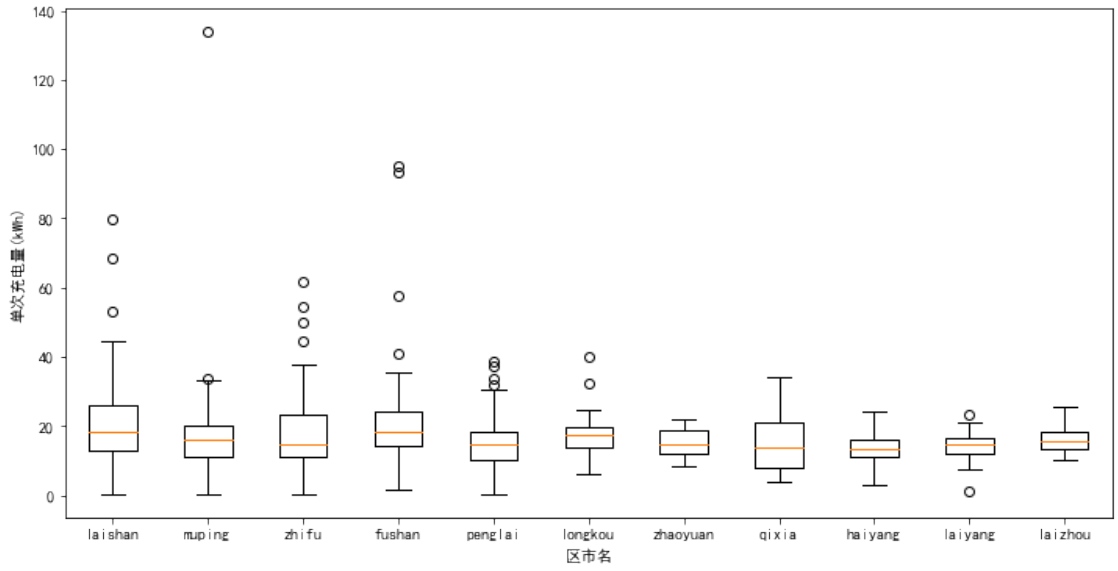


图 1.9 各区市单次充电充电量

接着，绘制各充电桩月充电量。月充电量的大小反映了各桩的使用率信息。实现代码如下。

[20]:
```
data_power_per_char_pile = [0]*np.size(data_all_districts)
for i in range(np.size(data_all_districts)):
    data_power_per_char_pile[i] = data_all_districts[i].
↪drop(data_all_districts[i][data_all_districts[i]['Unit_per_Cha_time']==0].
↪index).Unit_per_Cha_pile
```

```
C:\Users\admin\anaconda3\lib\site-packages\numpy\core\fromnumeric.py:3208:
VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences
(which is a list-or-tuple of lists-or-tuples-or ndarrays with different lengths
or shapes) is deprecated. If you meant to do this, you must specify
'dtype=object' when creating the ndarray.
  return asarray(a).size
```

[62]:
```
plt.figure(figsize=(12,6))
plt.boxplot(data_power_per_char_pile, labels=district_name)
plt.xlabel('区市名')
plt.ylabel('各桩月均充电量 (kWh)')
```

[62]:
```
Text(0, 0.5, ' 各桩月均充电量 (kWh)')
```

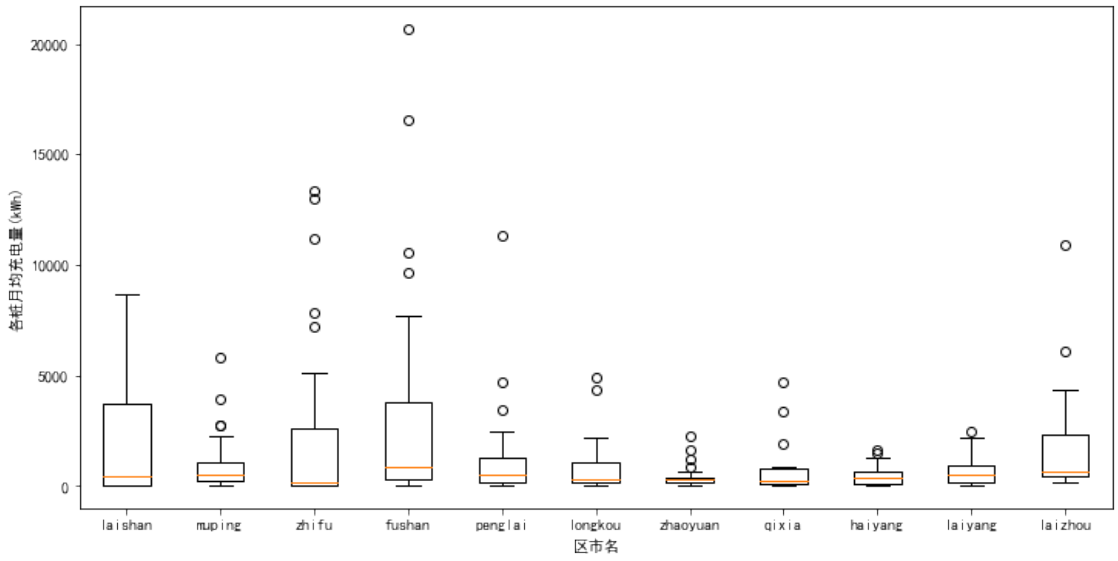


图 1.10 各区市各桩月充电量箱线图

各区市充电站数量和总充电桩数量可以体现区域发展情况，由以下代码实现。

[22]:
```
# 统计各区市充电桩总数量
total_num_piles = [0]*np.size(data_all_districts)
for i in range(np.size(data_all_districts)):
    total_num_piles[i] = data_all_districts[i]['Number_charging_piles'].sum()
```

```
C:\Users\admin\anaconda3\lib\site-packages\numpy\core\fromnumeric.py:3208:
VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences
(which is a list-or-tuple of lists-or-tuples-or ndarrays with different lengths
or shapes) is deprecated. If you meant to do this, you must specify
'dtype=object' when creating the ndarray.
  return asarray(a).size
```

[63]:
```
# 绘制柱状图
plt.figure(figsize=(12,6))
plt.bar(district_name, total_num_piles, tick_label=district_name)
plt.xlabel('区市名')
plt.ylabel('充电桩总数量')
for a,b in zip(district_name, total_num_piles):
    plt.text(a,b,
             b,
             ha='center',
             va='bottom',
            )
```

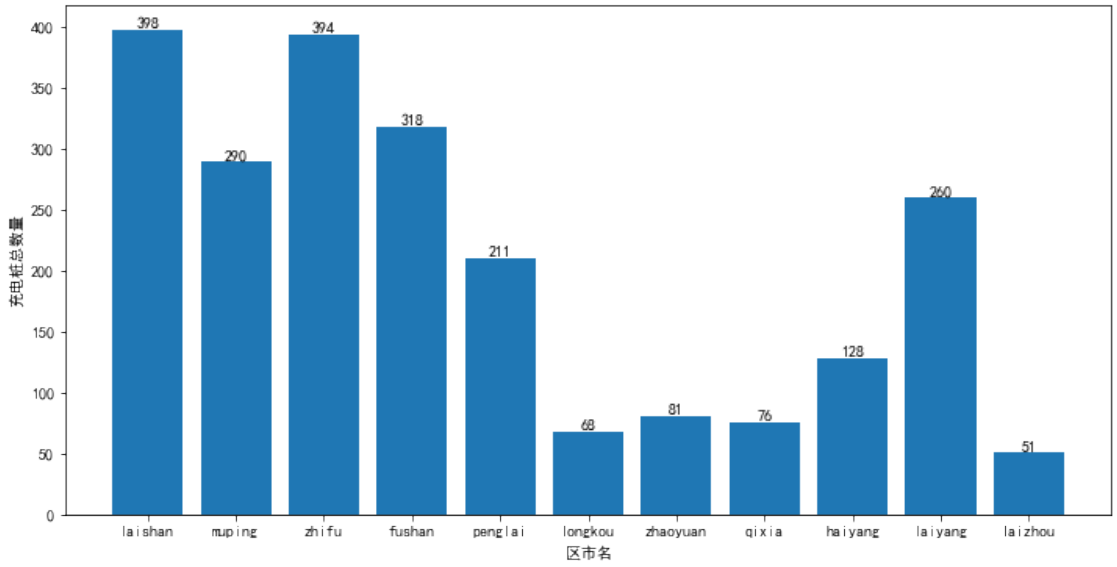


图 1.11 各区市总充电桩数量

[24]:
```
# 汇总各区市总充电站数量
total_num_stations = [0]*np.size(data_all_districts)
for i in range(np.size(data_all_districts)):
    total_num_stations[i] = data_all_districts[i]['Station_name'].count()
```

[64]:
```
# 汇总各区市总充电站数量柱状图
plt.figure(figsize=(12,6))
plt.bar(district_name, total_num_stations, tick_label=district_name)
plt.xlabel('区市名')
plt.ylabel('充电站总数量')
for a,b in zip(district_name, total_num_stations):
    plt.text(a,b,
             b,
             ha='center',
             va='bottom',
            )
```

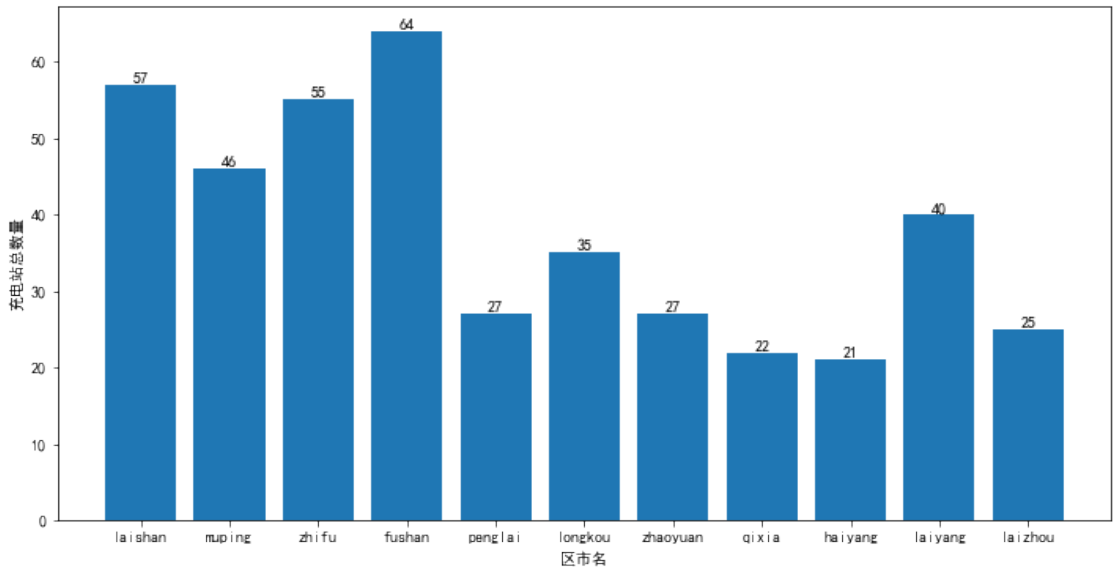


图 1.12 各区市总充电站数量

2 莱山区各充电站数据分析

[26]:
```
# 展示莱山区充电站数据表
num_stations_laishan = total_num_stations[0]
data_laishan.head()
```

[26]:

	Unnamed: 0	Station_name	district	year_month	power_consum（kWh）	times_charging	Number_charging_piles	latitude	longitude
0	0	烟台高新区公交场站	莱山区	202109	147737.44	1857	17	37.411116	121.530138
2	2	烟台莱山长宁路充电站	莱山区	202109	125802.78	5098	25	37.447031	121.438939
13	13	山东省烟台市莱山光华充电站	莱山区	202109	64413.98				
14	14	烟台朱堵堡公交场站	莱山区	202109	58316.32				
16	16	烟台上海滩花园公交场站	莱山区	202109	51615.61				

```
13            3201                     8  37.467384  121.443810
14            2191                     7  37.351410  121.350403
16             968                     8  37.440314  121.462538
```

```
    Unit_per_Cha_time  Unit_per_Cha_pile  month_use_frequency_per_pile
0               79.56            8690.44                           109
2               24.68            5032.11                           204
13              20.12            8051.75                           400
14              26.62            8330.90                           313
16              53.32            6451.95                           121
```

莱山区各充电站数量展示有下列代码实现。

[65]:
```
# charging piles number
from pylab import *
mpl.rcParams['font.sans-serif'] = ['SimHei']
plt.figure(figsize=(40,8))
plt.bar(data_laishan['Station_name'], data_laishan['Number_charging_piles'],␣
 ↪tick_label=data_laishan['Station_name'])
plt.xlabel('莱山区充电站名')
plt.ylabel('充电桩数量')
for a,b in zip(data_laishan['Station_name'],␣
 ↪data_laishan['Number_charging_piles']):
    plt.text(a,b,
             b,
             ha='center',
             va='bottom',
            )
plt.xticks(rotation = 90)
import warnings
warnings.filterwarnings('ignore')
```

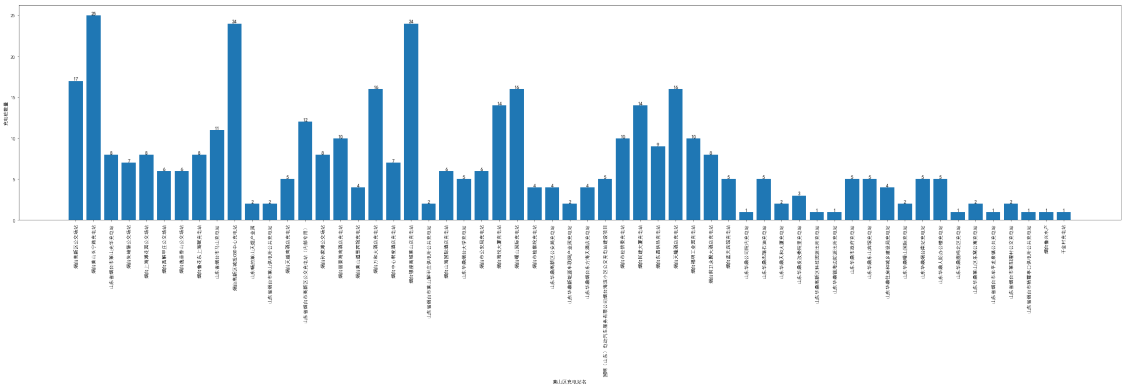


图 1.13 莱山区各充电站充电桩数量

各充电站月使用频率由以下代码实现。

[66]:
```
# 绘制月使用频率图
plt.figure(figsize=(40,8))
plt.bar(data_laishan['Station_name'],␣
 ↪data_laishan['month_use_frequency_per_pile'],␣
 ↪tick_label=data_laishan['Station_name'])
plt.xlabel('莱山区充电站名')
plt.ylabel('每个充电桩月均使用次数')
for a,b in zip(data_laishan['Station_name'],␣
 ↪data_laishan['month_use_frequency_per_pile']):
    plt.text(a,b,
             b,
             ha='center',
             va='bottom',
            )
plt.xticks(rotation = 90)
import warnings
warnings.filterwarnings('ignore')
```

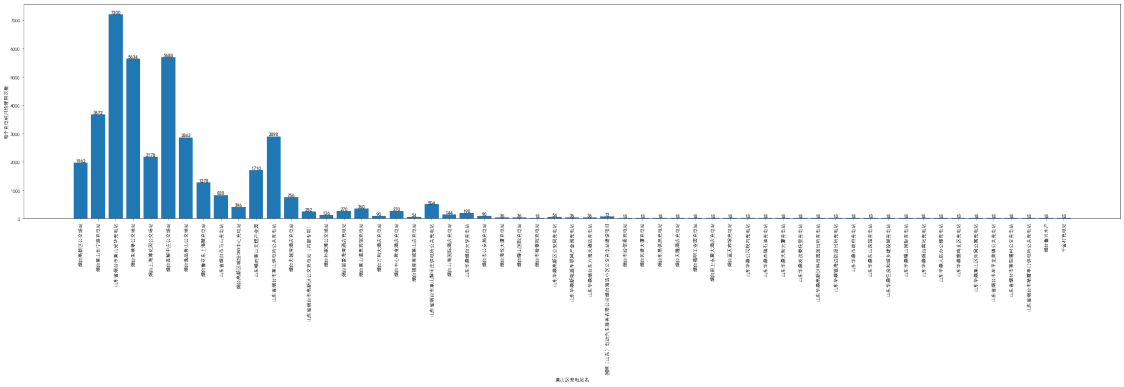


图 1.14 莱山区充电桩月均使用次数

同样，绘制各充电站地理位置。

[29]:
```
# 绘制各充电站地理位置
laishan_map = folium.Map(location = [latitude, longitude], zoom_start = 12)
incidents_laishan =plugins.MarkerCluster().add_to(laishan_map)
for lat, lng, name in zip(data_laishan.latitude, data_laishan.longitude,␣
 ↪data_laishan.Station_name):
    incidents_laishan.add_child(
        folium.CircleMarker(
            [lat, lng],
            radius=7, # define how big you want the circle markers to be
            color='yellow',
            fill=True,
            fill_color='red',
            fill_opacity=0.4,
            popup=name
        )
    )
circleMarker_laishan = basic_map()
circleMarker_laishan.add_child(incidents_laishan)
```

[29]:
```
<folium.folium.Map at 0x1cfe5203940>
```

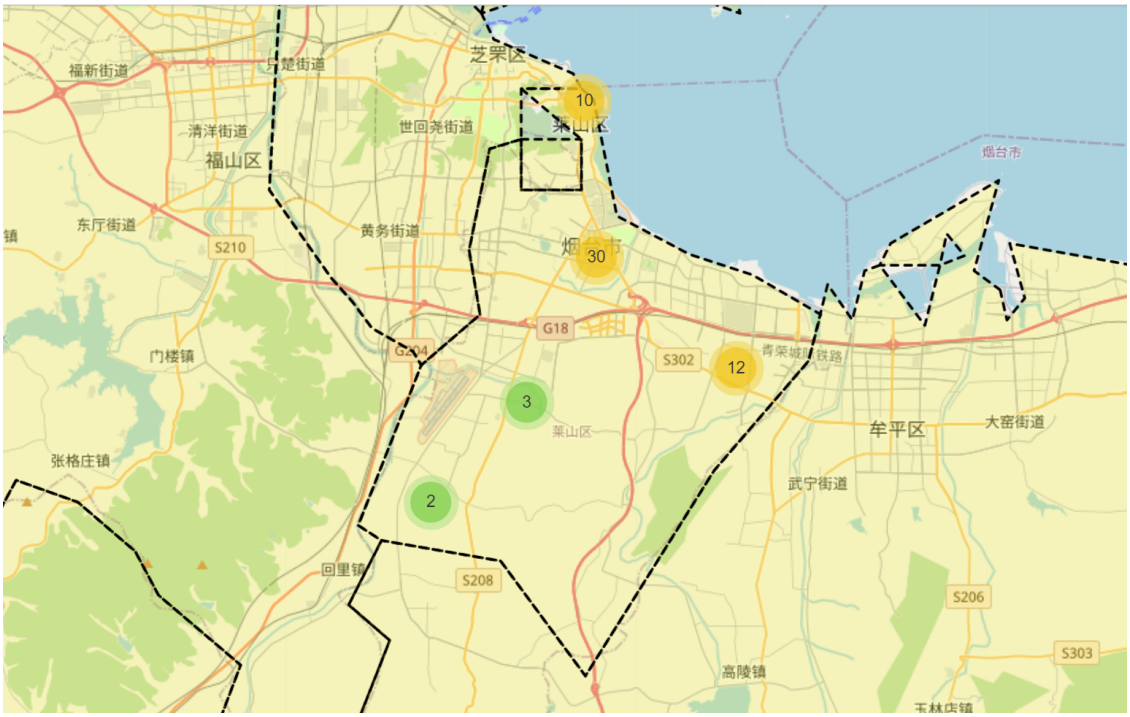


图 1.15 莱山区充电站位置

绘制莱山区充电桩数量热图。

[30]:
```
# 绘制热图
heatMap_laishan = circleMarker_laishan
heatdata = data_laishan[['latitude','longitude','Number_charging_piles']].
 ↪values.tolist()
HeatMap(heatdata).add_to(heatMap_laishan)
heatMap_laishan
```

[30]: <folium.folium.Map at 0x1cfe5203940>

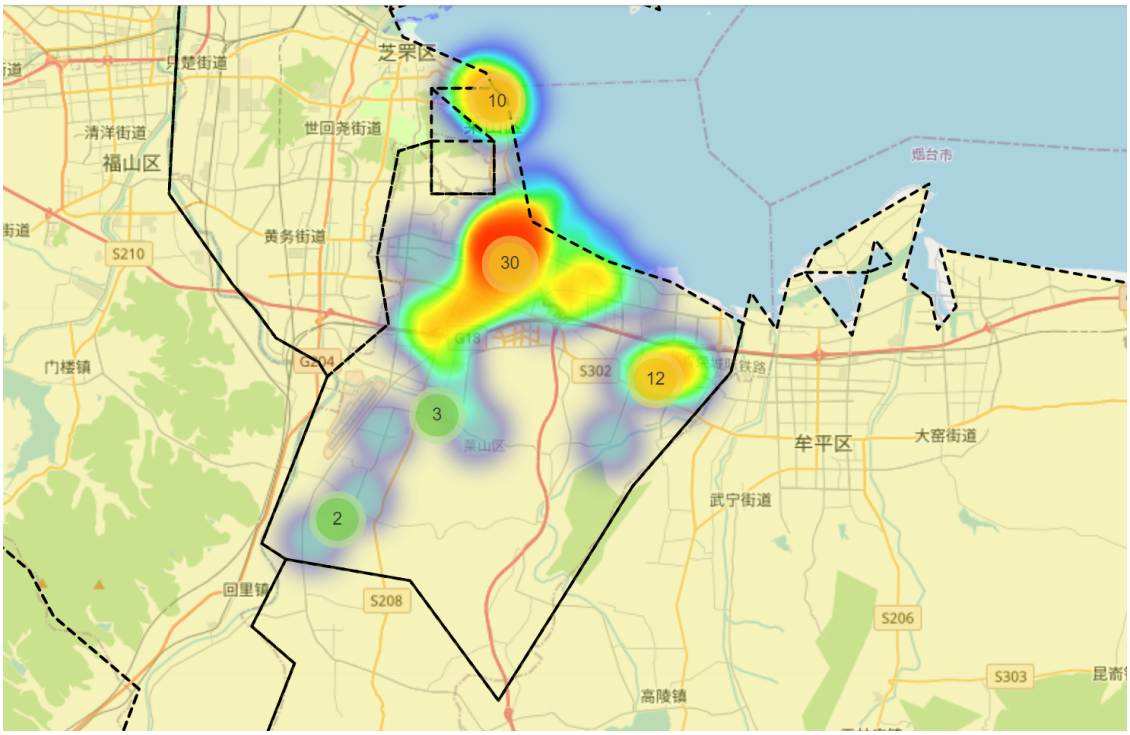


图 1.16 莱山区充电桩数量热图

[31]:
```
# 充电桩月使用频率热图
heatMap_use_frequncy_laishan = circleMarker_laishan
heatdata =␣
↪data_laishan[['latitude','longitude','month_use_frequency_per_pile']].values.
↪tolist()
HeatMap(heatdata).add_to(heatMap_use_frequncy_laishan)
heatMap_use_frequncy_laishan
```

[31]:
```
<folium.folium.Map at 0x1cfe5203940>
```

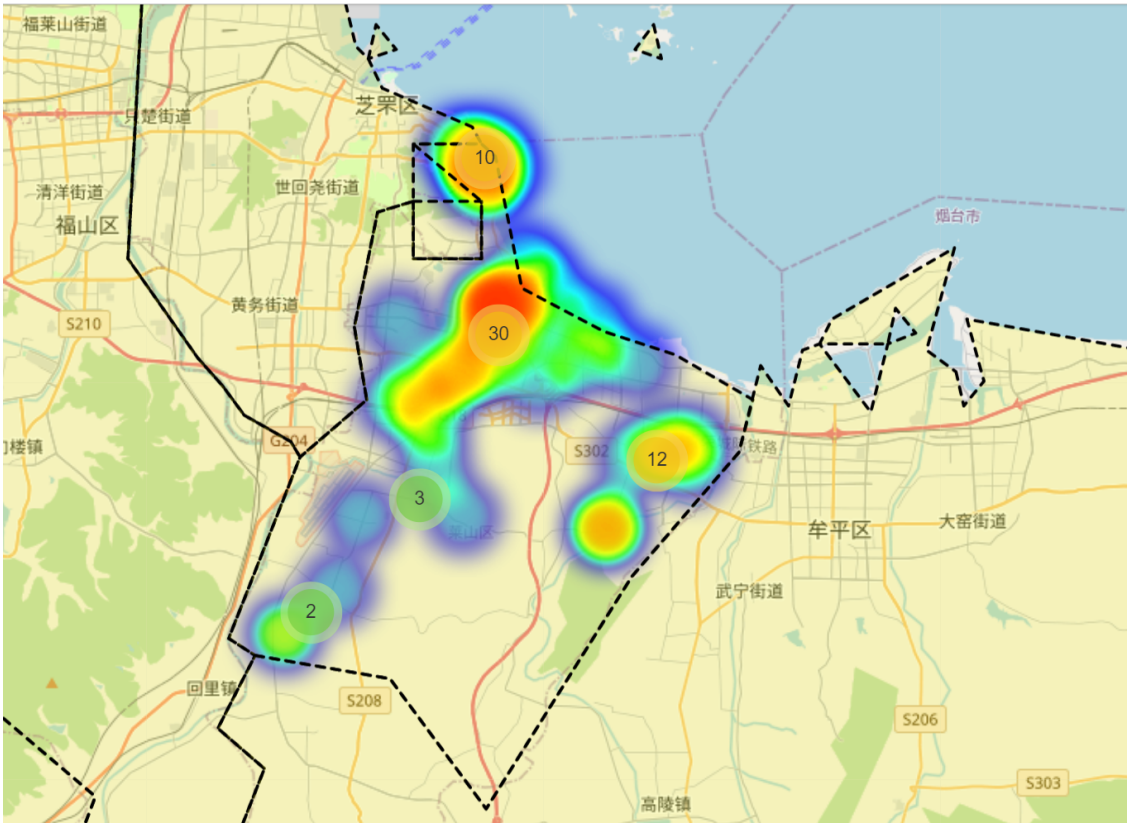


图 1.17 莱山区充电桩月使用频率热图

[32]:
```
# heatmap of power use of each charging time
heatMap_power_use_times_laishan = circleMarker_laishan
heatdata = data_laishan[['latitude','longitude','Unit_per_Cha_time']].values.
 ↪tolist()
HeatMap(heatdata).add_to(heatMap_power_use_times_laishan)
heatMap_power_use_times_laishan
```

[32]:
```
<folium.folium.Map at 0x1cfe5203940>
```

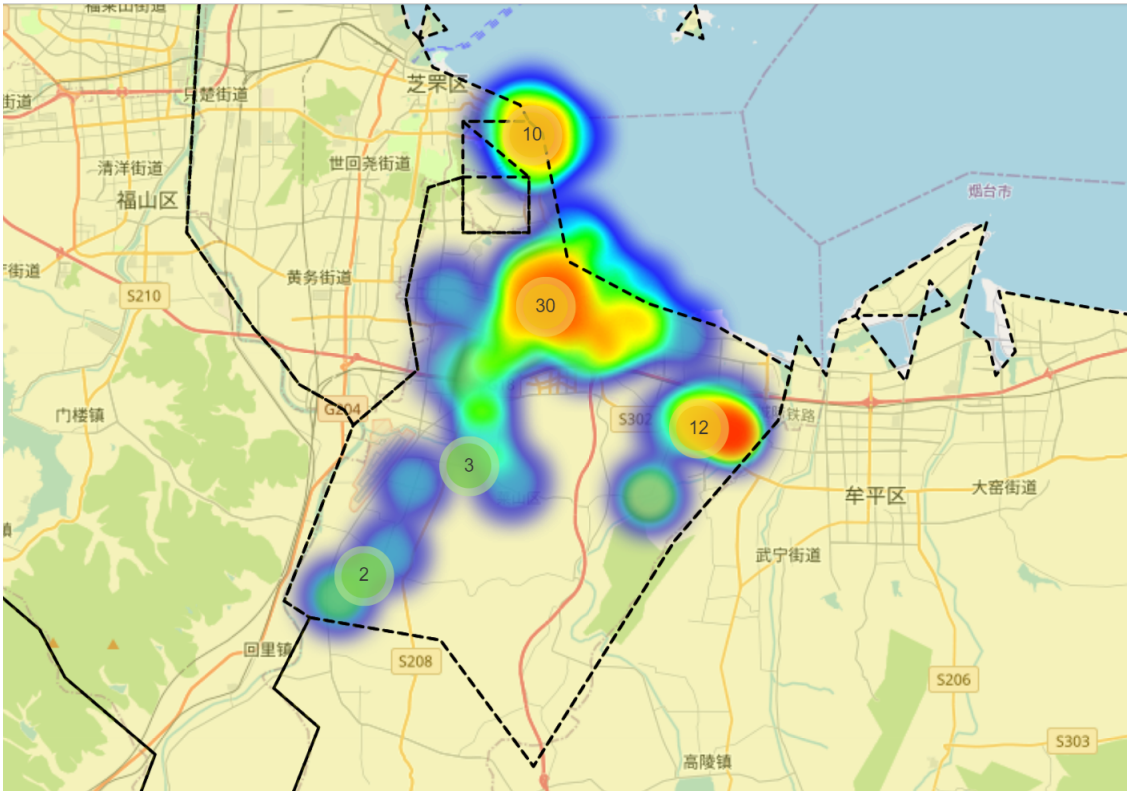


图 1.18 莱山区每次充电量热图

[33]:
```
# heatmap of monthly power use of each pile
heatMap_power_use_piles_laishan = circleMarker_laishan
heatdata = data_laishan[['latitude','longitude','Unit_per_Cha_pile']].values.
 ↪tolist()
HeatMap(heatdata).add_to(heatMap_power_use_piles_laishan)
heatMap_power_use_piles_laishan
```

[33]:
```
<folium.folium.Map at 0x1cfe5203940>
```

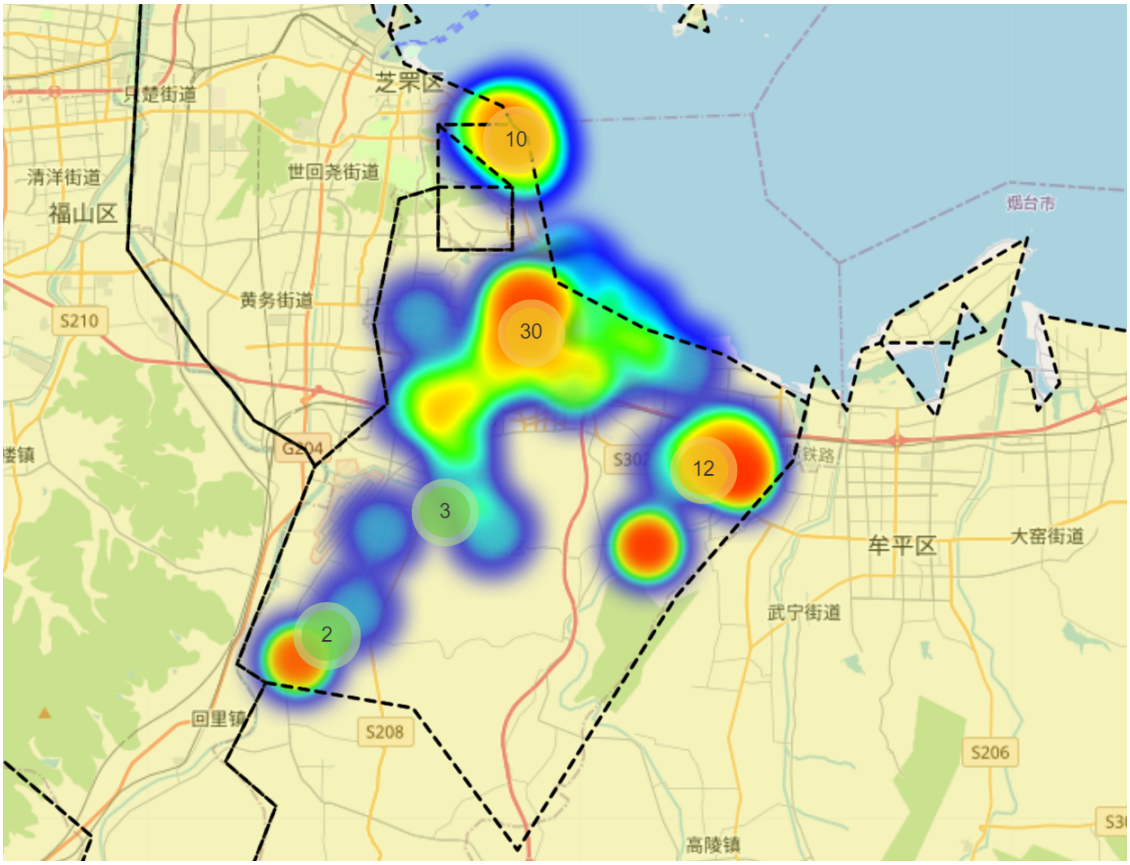


图 1.19 莱山区单桩月约充电量

3 莱山区充电桩未来发展规划（2021，2025，2030），计划 A, 在已有充电站基础上规划充电桩

[68]:
```
'''incidents_demand_point = folium.map.FeatureGroup()

# Loop through the 200 crimes and add each to the incidents feature group
for lat, lng, name in zip(data_laishan.latitude, data_laishan.longitude,␣
 ↪data_laishan.Station_name):
    incidents_demand_point.add_child(
        folium.CircleMarker(
            [lat, lng],
            radius=7, # define how big you want the circle markers to be
            color='yellow',
            fill=True,
```

```
            fill_color='red',
            fill_opacity=0.4,
            popup=name

        )
    )

# Add incidents to map
basic_map().add_child(incidents_demand_point)
'''
```

[68]:
```
"incidents_demand_point = folium.map.FeatureGroup()\n\n# Loop through the 200
crimes and add each to the incidents feature group\nfor lat, lng, name in
zip(data_laishan.latitude, data_laishan.longitude, data_laishan.Station_name):\n
incidents_demand_point.add_child(\n        folium.CircleMarker(\n
[lat, lng],\n            radius=7, # define how big you want the circle markers
to be\n            color='yellow',\n            fill=True,\n
fill_color='red',\n            fill_opacity=0.4,\n            popup=name\n
\n        )\n    )\n\n# Add incidents to
map\nbasic_map().add_child(incidents_demand_point)\n"
```

[69]:
```
# 显示部分莱山区数据信息情况
data_laishan.head()
```

[69]:

	Unnamed: 0	Station_name	district	year_month	power_consum（kWh）	times_charging	Number_charging_piles	latitude	longitude
0	0	烟台高新区公交场站	莱山区	202109	147737.44	1857	17	37.411116	121.530138
2	2	烟台莱山长宁路充电站	莱山区	202109	125802.78	5098	25	37.447031	121.438939
13	13	山东省烟台市莱山光华充电站	莱山区	202109	64413.98				
14	14	烟台朱堵堡公交场站	莱山区	202109	58316.32				
16	16	烟台上海滩花园公交场站	莱山区	202109	51615.61				

```
13            3201                      8  37.467384  121.443810
14            2191                      7  37.351410  121.350403
16             968                      8  37.440314  121.462538

    Unit_per_Cha_time  Unit_per_Cha_pile  month_use_frequency_per_pile  \
0               79.56            8690.44                          1962
2               24.68            5032.11                          3672
13              20.12            8051.75                          7200
14              26.62            8330.90                          5634
16              53.32            6451.95                          2178

    time_use_efficiency
0                13.625
2                25.500
13               50.000
14               39.125
16               15.125
```

3.1 2021 充电桩规划

对莱山区充电站进行筛选，筛选出充电桩月均使用频率大于 10 的充电桩的定位，实现代码如下。

[37]:
```
# 实例化地图
incidents_demand_point_high_use_frequecy = folium.map.FeatureGroup()
# 指定月均使用频率
month_use_frequency = 10
data_high_use_frequency =␣
 ↪data_laishan[data_laishan['month_use_frequency_per_pile']>month_use_frequency]
# 定位
for lat, lng, name in zip(data_high_use_frequency.latitude,␣
 ↪data_high_use_frequency.longitude, data_high_use_frequency.
 ↪Number_charging_piles):
    incidents_demand_point_high_use_frequecy.add_child(
        folium.CircleMarker(
            [lat, lng],
            radius=7, # define how big you want the circle markers to be
            color='yellow',
```

```
            fill=True,
            fill_color='red',
            fill_opacity=0.4,
            popup=name

        )
    )

# Add incidents to map
basic_map().add_child(incidents_demand_point_high_use_frequecy)
```

[37]: <folium.folium.Map at 0x1cfe553ce50>

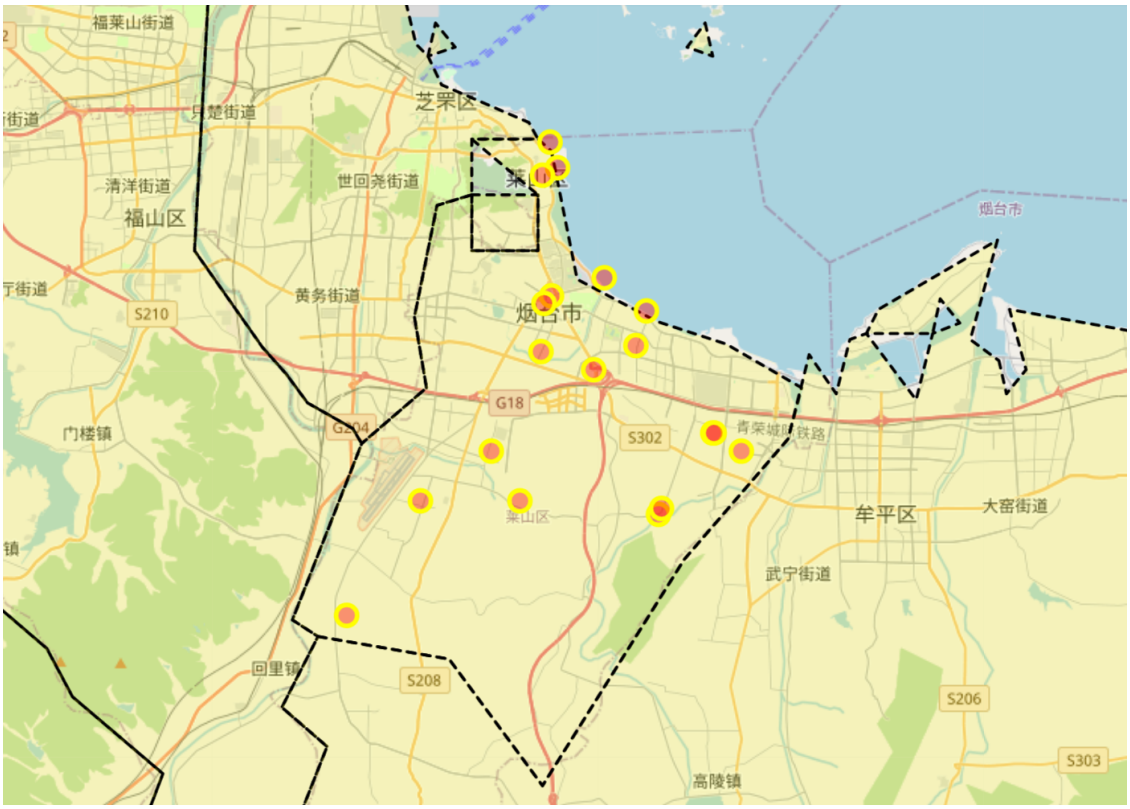


图 1.20 莱山区使用频率较高充电站

[70]:
```
# 数据展示
data_high_use_frequency.head()
```

[70]:

	Unnamed: 0	Station_name	district	year_month	power_consum（kWh）	times_charging	Number_charging_piles	latitude	longitude	Unit_per_Cha_time	Unit_per_Cha_pile	month_use_frequency_per_pile	demand factor	time_use_efficiency
0	0	烟台高新区公交场站	莱山区	202109	147737.44	1857	17	37.411116	121.530138	79.56	8690.44	109	36	0.756944
2	2	烟台莱山长宁路充电站	莱山区	202109	125802.78	5098	25	37.447031	121.438939	24.68	5032.11	204	68	1.416667
13	13	山东省烟台市莱山光华充电站	莱山区	202109	64413.98	3201	8	37.467384	121.443810	20.12	8051.75	400	133	2.777778
14	14	烟台朱堵堡公交场站	莱山区	202109	58316.32	2191	7	37.351410	121.350403	26.62	8330.90	313	104	2.173611
16	16	烟台上海滩花园公交场站	莱山区	202109	51615.61	968	8	37.440314	121.462538	53.32	6451.95	121	40	0.840278

为莱山区充电站情况定义相关指标，如下：
需求因素指标（demand factor），使用效率（time_use_efficiency），以及需增加的电动桩数量（piles_add_number）。

[40]:
```
# 计算需求因素指标
```

```
data_high_use_frequency['demand factor'] =␣
↪(data_high_use_frequency['month_use_frequency_per_pile']/3).astype(int)
# difine the indexes to judge the stations efficiency/overload or not, 5 means␣
↪charging 5 hours per time, usually, 2 hours can be full charged with public␣
↪charging pile
# but custermer won't remove the EV cars imediately, so we assume 5 hours per␣
↪charging time, if time_use_efficiency>1 means that more charging piles␣
↪should be installed
# otherwise, no more charging piles are needed
duration = 5
data_high_use_frequency['time_use_efficiency'] =␣
↪(data_high_use_frequency['month_use_frequency_per_pile']/30)*duration/24
# which stations should add new piles, and how many should be added?
station_add_piles =␣
↪data_high_use_frequency[data_high_use_frequency['time_use_efficiency']>1.0]
station_add_piles['piles_add_number'] = np.
↪ceil(station_add_piles['times_charging']*duration/
↪(30*24)-station_add_piles['Number_charging_piles']).astype(int)
```

[41]:
```
station_add_piles.head(10)
```

[41]:
```
    Unnamed: 0      Station_name district  year_month  power_consum（kWh）  \
2            2        烟台莱山长宁路充电站       莱山区      202109          ␣
↪125802.78
13          13     山东省烟台市莱山光华充电站       莱山区      202109          ␣
↪64413.98
14          14         烟台朱堵堡公交场站       莱山区      202109          ␣
↪58316.32
18          18        烟台西解甲庄公交场站       莱山区      202109          ␣
↪49071.64
30          30        烟台逸品香山公交场站       莱山区      202109          ␣
↪29516.10
65          65  山东省烟台市莱山供电所公共充电站       莱山区      202109          ␣
↪  6546.01

    times_charging  Number_charging_piles   latitude   longitude  \
```

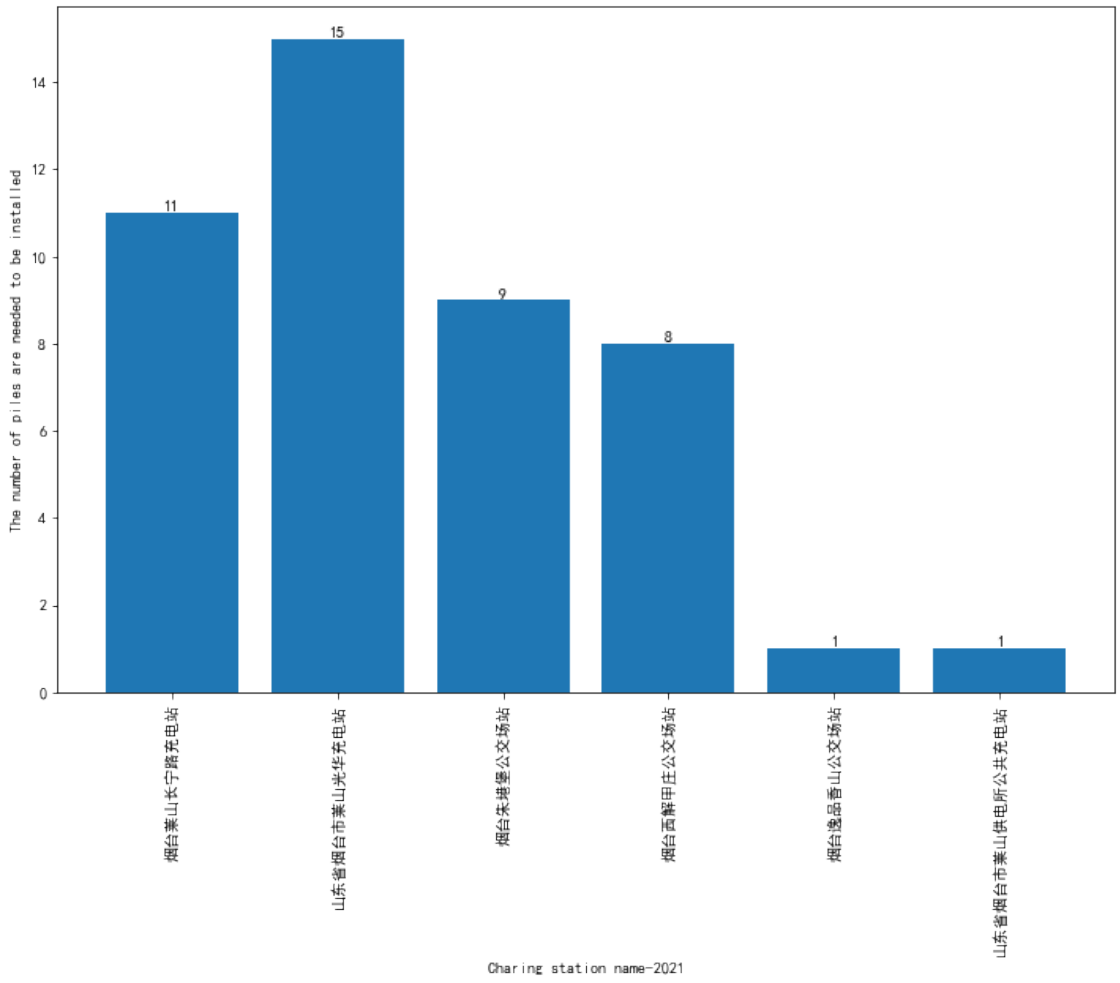
```
2              5098                     25  37.447031  121.438939
13             3201                      8  37.467384  121.443810
14             2191                      7  37.351410  121.350403
18             1898                      6  37.388390  121.492616
30              956                      6  37.393285  121.429032
65              322                      2  37.393078  121.383545

    Unit_per_Cha_time  Unit_per_Cha_pile  month_use_frequency_per_pile  \
2               24.68            5032.11                           204
13              20.12            8051.75                           400
14              26.62            8330.90                           313
18              25.85            8178.61                           316
30              30.87            4919.35                           159
65              20.33            3273.01                           161

    demand factor  time_use_efficiency  piles_add_number
2              68             1.416667                11
13            133             2.777778                15
14            104             2.173611                 9
18            105             2.194444                 8
30             53             1.104167                 1
65             53             1.118056                 1
```

[42]:
```
# figure out the added piles in the charging stations
plt.figure(figsize=(12,8))
plt.bar(station_add_piles['Station_name'],␣
 ↪station_add_piles['piles_add_number'],␣
 ↪tick_label=station_add_piles['Station_name'])
plt.xlabel('Charing station name-2021')
plt.ylabel('The number of piles are needed to be installed')
plt.xticks(rotation = 90)
for a,b in zip(station_add_piles['Station_name'],␣
 ↪station_add_piles['piles_add_number']):
    plt.text(a,b,
             b,
             ha='center',
```

```
            va='bottom',
           )
```



3.2 2025 充电桩规划

[43]:
```
# define indexes service factor Z_distance and E_service
# ✓ building new piles for specific stations
# E_service = W*d/(TD), W-demand factor, d-distance of demand and station, T-3␣
 ↪days in this model, D-2~3km to the charging station
# ✓ define a factor/efficiency to check if the piles of stations are enought/
 ↪overworked
# how close can be considered as the same charging station?
```

```
# if the charging station needs to be removed, which one should be removed?
# ✓ the future development of the EVs,now, 2025,2030
# if new charging stations are required, where should we locate it?
```

[44]:
```
# the future development of the EVs,now, 2025,2030
# according to the development of China, https://www.idc.com/getdoc.jsp?
 ↪containerId=prCHC47071920,https://www.canalys.com/newsroom/
 ↪canalys-%E7%BB%8F%E5%8E%86%E9%80%82%E5%BA%A6%E5%A2%9E%E9%95%BF%E7%9A%842020%E5%B9%B4%E5%90%
# 2025, it's 3* of 2021
```

[45]:
```
data_laishan['time_use_efficiency'] =␣
 ↪(data_laishan['month_use_frequency_per_pile']/30)*duration/24
```

[46]:
```
# EV cars increasing rate from 2021 to 2025
increase_rate_2025 = 3
data_laishan_2025 = data_laishan
data_laishan_2025['month_use_frequency_per_pile'] =␣
 ↪data_laishan_2025['month_use_frequency_per_pile']*increase_rate_2025
data_laishan_2025['time_use_efficiency'] =␣
 ↪(data_laishan_2025['month_use_frequency_per_pile']/30)*duration/24
```

[47]:
```
station_add_piles_2025 =␣
 ↪data_laishan_2025[data_laishan_2025['time_use_efficiency']>1.0]
station_add_piles_2025['piles_add_number'] = np.
 ↪ceil(station_add_piles_2025['times_charging']*increase_rate_2025*duration/
 ↪(30*24)-station_add_piles_2025['Number_charging_piles']).astype(int)
```

[48]:
```
station_add_piles_2025.head(57)
```

[48]:

	Unnamed: 0	Station_name	district	year_month	power_consum（kWh）
0	0	烟台高新区公交场站	莱山区	202109	147737.44
2	2	烟台莱山长宁路充电站	莱山区	202109	125802.78
13	13	山东省烟台市莱山光华充电站	莱山区	202109	64413.98
14	14	烟台朱堵堡公交场站	莱山区	202109	58316.32

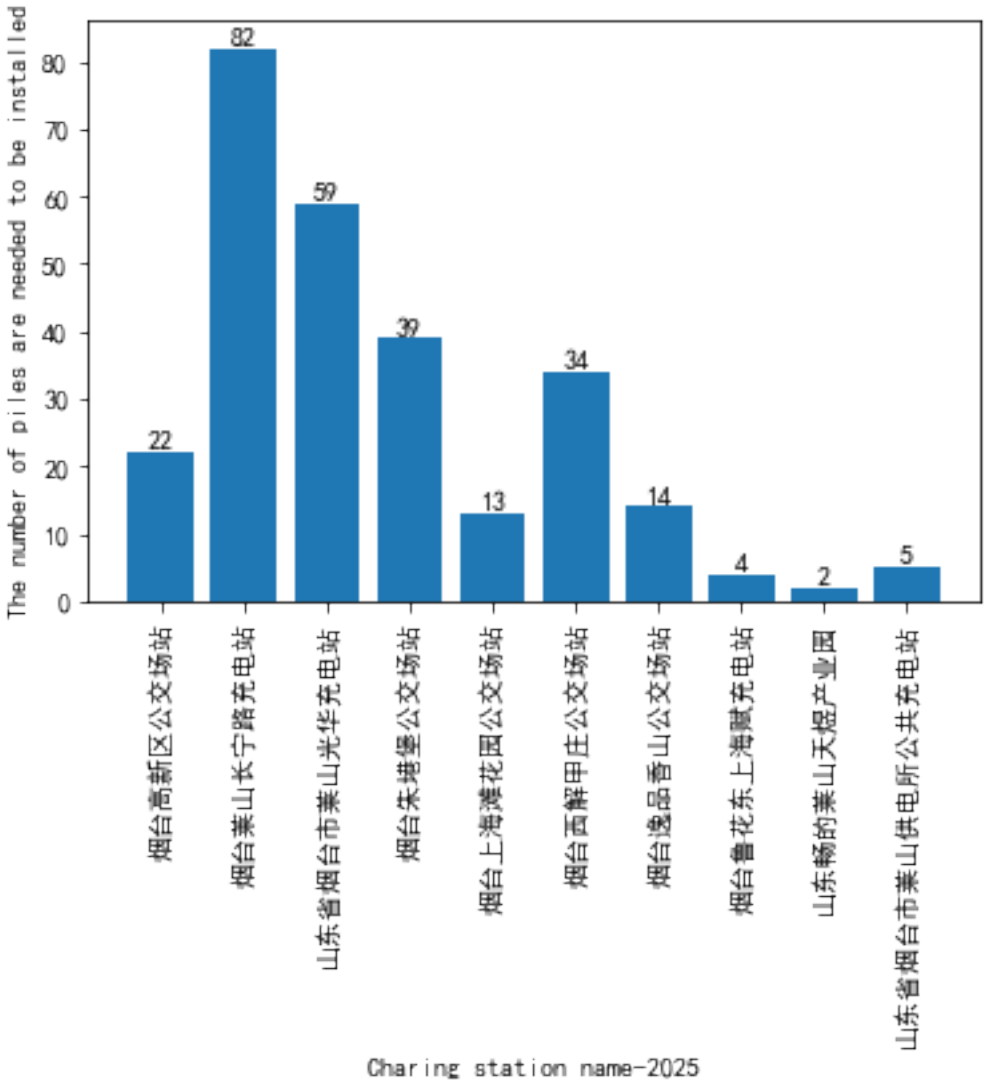
```
16          16        烟台上海滩花园公交场站       莱山区       202109          ␣
↪51615.61
18          18         烟台西解甲庄公交场站       莱山区       202109          ␣
↪49071.64
30          30         烟台逸品香山公交场站       莱山区       202109          ␣
↪29516.10
53          53        烟台鲁花东上海赋充电站       莱山区       202109          ␣
↪9792.35
61          61        山东畅的莱山天煜产业园       莱山区       202109          ␣
↪8409.60
65          65  山东省烟台市莱山供电所公共充电站       莱山区       202109          ␣
↪  6546.01
```

	times_charging	Number_charging_piles	latitude	longitude	\
0	1857	17	37.411116	121.530138	
2	5098	25	37.447031	121.438939	
13	3201	8	37.467384	121.443810	
14	2191	7	37.351410	121.350403	
16	968	8	37.440314	121.462538	
18	1898	6	37.388390	121.492616	
30	956	6	37.393285	121.429032	
53	564	8	37.522659	121.442859	
61	189	2	37.411027	121.416223	
65	322	2	37.393078	121.383545	

	Unit_per_Cha_time	Unit_per_Cha_pile	month_use_frequency_per_pile	\
0	79.56	8690.44	327	
2	24.68	5032.11	612	
13	20.12	8051.75	1200	
14	26.62	8330.90	939	
16	53.32	6451.95	363	
18	25.85	8178.61	948	
30	30.87	4919.35	477	
53	17.36	1224.04	213	
61	44.50	4204.80	285	
65	20.33	3273.01	483	

	time_use_efficiency	piles_add_number
0	2.270833	22
2	4.250000	82
13	8.333333	59
14	6.520833	39
16	2.520833	13
18	6.583333	34
30	3.312500	14
53	1.479167	4
61	1.979167	2
65	3.354167	5

[49]:
```
# figure out the added piles in the charging stations of 2025
plt.bar(station_add_piles_2025['Station_name'],␣
 ↪station_add_piles_2025['piles_add_number'],␣
 ↪tick_label=station_add_piles_2025['Station_name'])
plt.xlabel('Charing station name-2025')
plt.ylabel('The number of piles are needed to be installed')
plt.xticks(rotation = 90)
for a,b in zip(station_add_piles_2025['Station_name'],␣
 ↪station_add_piles_2025['piles_add_number']):
    plt.text(a,b,
             b,
             ha='center',
             va='bottom',
            )
```



3.3 2030 充电桩规划

[50]:
```
# EV cars increasing rate from 2021 to 2025, 6*
increase_rate_2030 = 6
data_laishan_2030 = data_laishan
data_laishan_2030['month_use_frequency_per_pile'] =␣
 ↪data_laishan_2025['month_use_frequency_per_pile']*increase_rate_2030
data_laishan_2030['time_use_efficiency'] =␣
 ↪(data_laishan_2030['month_use_frequency_per_pile']/30)*duration/24
```

[51]:
```
station_add_piles_2030 =␣
↪data_laishan_2030[data_laishan_2030['time_use_efficiency']>1.0]
station_add_piles_2030['piles_add_number'] = np.
↪ceil(station_add_piles_2030['times_charging']*increase_rate_2030*duration/
↪(30*24)-station_add_piles_2030['Number_charging_piles']).astype(int)
```

[52]:
```
station_add_piles_2030.head(10)
```

[52]:

	Unnamed: 0	Station_name	district	year_month	power_consum（kWh）	times_charging	Number_charging_piles	latitude	longitude
0	0	烟台高新区公交场站	莱山区	202109	147737.44	1857	17	37.411116	121.530138
2	2	烟台莱山长宁路充电站	莱山区	202109	125802.78	5098	25	37.447031	121.438939
13	13	山东省烟台市莱山光华充电站	莱山区	202109	64413.98	3201	8	37.467384	121.443810
14	14	烟台朱堵堡公交场站	莱山区	202109	58316.32	2191	7	37.351410	121.350403
16	16	烟台上海滩花园公交场站	莱山区	202109	51615.61	968	8	37.440314	121.462538
18	18	烟台西解甲庄公交场站	莱山区	202109	49071.64				
30	30	烟台逸品香山公交场站	莱山区	202109	29516.10				
53	53	烟台鲁花东上海赋充电站	莱山区	202109	9792.35				
57	57	山东省烟台市马山充电站	莱山区	202109	9203.06				
60	60	烟台高新区城投 CBD 中心充电站	莱山区	202109	8828.70				

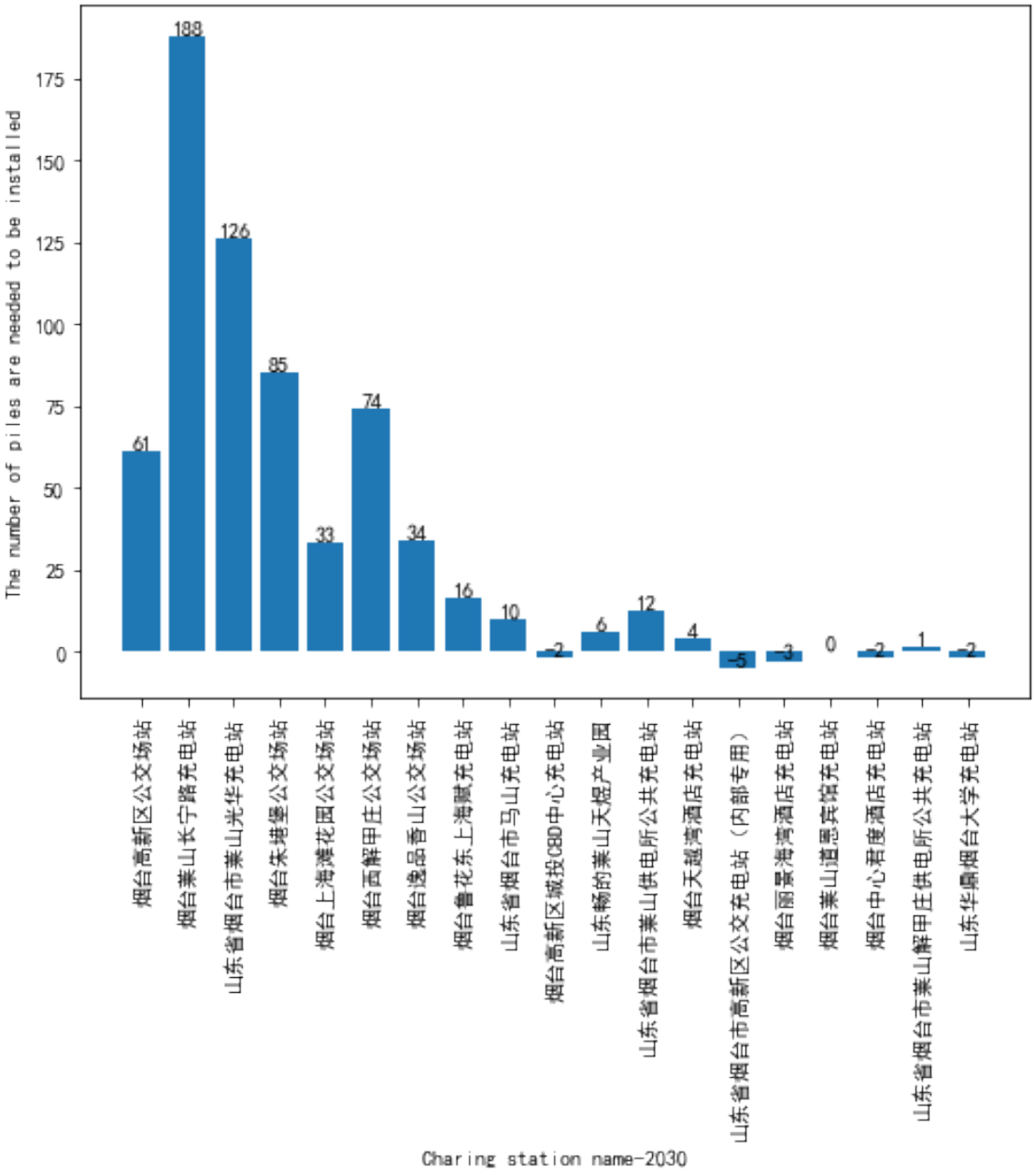
```
18            1898                     6  37.388390  121.492616
30             956                     6  37.393285  121.429032
53             564                     8  37.522659  121.442859
57             502                    11  37.417534  121.517010
60             518                    24  37.449213  121.482192

    Unit_per_Cha_time  Unit_per_Cha_pile  month_use_frequency_per_pile  \
0               79.56            8690.44                          1962
2               24.68            5032.11                          3672
13              20.12            8051.75                          7200
14              26.62            8330.90                          5634
16              53.32            6451.95                          2178
18              25.85            8178.61                          5688
30              30.87            4919.35                          2862
53              17.36            1224.04                          1278
57              18.33             836.64                           828
60              17.04             367.86                           396

    time_use_efficiency  piles_add_number
0                13.625                61
2                25.500               188
13               50.000               126
14               39.125                85
16               15.125                33
18               39.500                74
30               19.875                34
53                8.875                16
57                5.750                10
60                2.750                -2
```

[53]:
```
# figure out the added piles in the charging stations of 2025, https://finance.
 ↪sina.com.cn/chanjing/cyxw/2019-05-19/doc-ihvhiews2890186.shtml
plt.figure(figsize=(8,6))
plt.bar(station_add_piles_2030['Station_name'],␣
 ↪station_add_piles_2030['piles_add_number'],␣
 ↪tick_label=station_add_piles_2030['Station_name'])
```

```
plt.xlabel('Charing station name-2030')
plt.ylabel('The number of piles are needed to be installed')
plt.xticks(rotation = 90)
for a,b in zip(station_add_piles_2030['Station_name'],␣
 ↪station_add_piles_2030['piles_add_number']):
    plt.text(a,b,
             b,
             ha='center',
             va='bottom',
            )
```


The number of piles are needed to be installed
0
25
50
75
100
125
150
175
61
188
126
85
33
74
34
16
10
-2
6
12
4
-5
-3
0
-2
1
-2
烟台高新区公交场站
烟台莱山长宁路充电站
山东省烟台市莱山光华充电站
烟台朱塔堡公交场站
烟台上海滨花园公交场站
烟台西解甲庄公交场站
烟台逸品香山公交场站
烟台鲁花东上海赋充电站
山东省烟台市马山充电站
烟台高新区城投CBD中心充电站
山东鞠的莱山天煜产业园
山东省烟台市莱山供电所公共充电站
烟台天越湾酒店充电站
山东省烟台市高新区公交充电站（内部专用）
烟台丽景海湾酒店充电站
烟台莱山道恩宾馆充电站
烟台中心君度酒店充电站
山东省烟台市莱山解甲庄供电所公共充电站
山东华鼎烟台大学充电站
Charing station name-2030

4 莱山区充电桩未来发展规划（2025，2030），计划 B, 可选定新的地址

[54]:
```
# building new stations should be more better in the future, select the␣
↪candidate position first
# for now, i.e., year of 2021
# for the year of 2025
# for the year of 2030
```

[55]:
```
# for now, i.e., year of 2021
# the traffic information should be provided to analyze the new charging␣
↪stations, and P-median methond can be used to find out the optimal charging␣
↪place/station
```

[56]:
```
# calculate the matrix distance of each charging station（unit:km）
# from geopy.distance import geodesic
# print(geodesic((34, -81), (32, -87)).km)
import math

def get_distances(locs_1, locs_2):
    n_rows_1 = locs_1.shape[0]
    n_rows_2 = locs_2.shape[0]
    dists = np.empty((n_rows_1, n_rows_2))
    # The loops here are inefficient
    for i in range(n_rows_1):
        for j in range(n_rows_2):
            dists[i, j] = get_distance_from_lat_long(locs_1[i], locs_2[j])
    return dists*1.609344


def get_distance_from_lat_long(loc_1, loc_2):

    earth_radius = 3958.75

    lat_dif = math.radians(loc_1[0] - loc_2[0])
    long_dif = math.radians(loc_1[1] - loc_2[1])
    sin_d_lat = math.sin(lat_dif / 2)
    sin_d_long = math.sin(long_dif / 2)
```

```
    step_1 = (sin_d_lat ** 2) + (sin_d_long ** 2) * math.cos(math.
↪radians(loc_1[0])) * math.cos(math.radians(loc_2[0]))
    step_2 = 2 * math.atan2(math.sqrt(step_1), math.sqrt(1-step_1))
    dist = step_2 * earth_radius
    return dist

locations = np.array(data_high_use_frequency[['latitude','longitude']])
distance_matrix = get_distances(locations, locations)
print(f'the maximum distance is {distance_matrix.max()} km')
print(f'the average distance is {distance_matrix.mean()} km')
```

```
the maximum distance is 20.717927505163097 km
the average distance is 7.842522650069849 km
```