

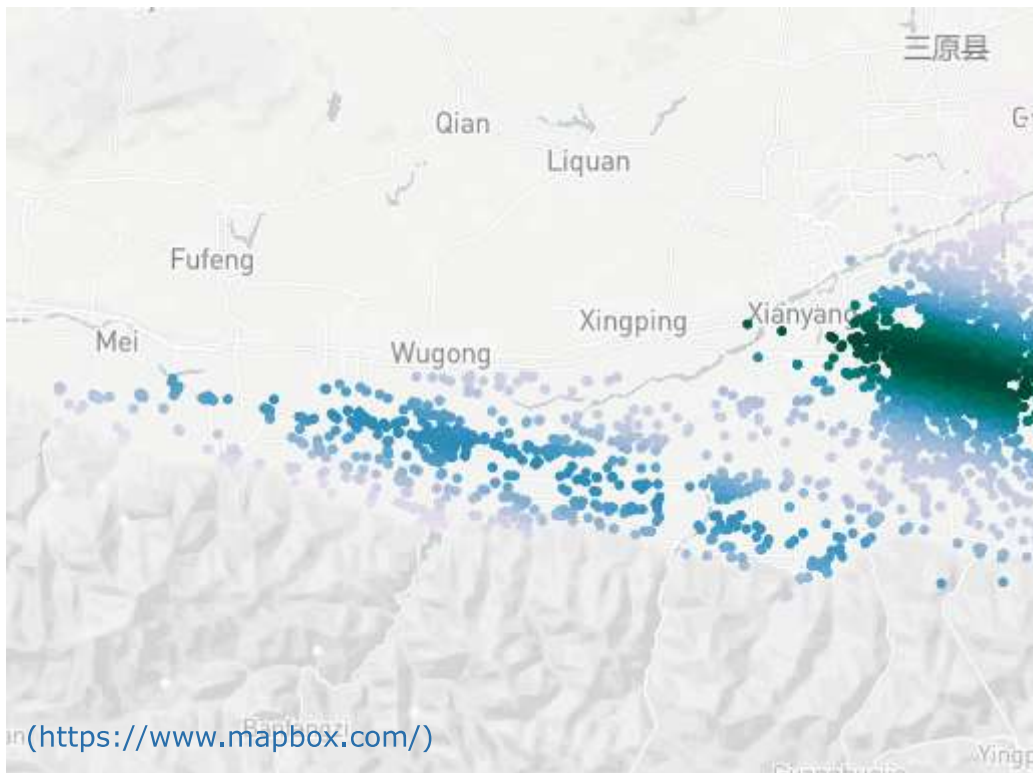
In [1]:

```

import pandas as pd
from scipy import stats
import plotly.express as px
import matplotlib
import math
poi_gpd=pd.read_pickle('C:\\Users\\97505\\Desktop\\code2\\bak1.pkl') #读取已经存储为.pkl格式的POI数据
poi_gpd.plot(marker=".", markersize=5)
poi_coordinates=poi_gpd[['location_lng', 'location_lat']].to_numpy().T #根据stats.gaussian_kde()输入
poi_coordi_kernel=stats.gaussian_kde(poi_coordinates) #核密度估计
poi_gpd['poi_kde']=poi_coordi_kernel(poi_coordinates)
poi_gpd.shopinfo=poi_gpd.shopinfo.fillna(0)
mapbox_token='pk.eyJJIjoicmljaGllYmFvIiwiaSI6ImNrYjB3N2NyMzB1MG8yc254dTRzNnMyeHMifQ.QT7MdjQKs9Y60taJ
px.set_mapbox_access_token(mapbox_token)
fig=px.scatter_mapbox(poi_gpd, lat=poi_gpd.location_lat, lon=poi_gpd.location_lng, color='poi_kde', co

fig.show()
poi_gpd.head()

```



Out [1]:

photo_url	location_lng	location_lat	scene	scene_id	pname	cityname	adname	to
0	[]	108.604087	34.089245	西安乾元新景鄠邑店	B0H3472CJK	陕西省	西安市	鄠邑区
1	[]	108.870518	34.139311	爱华实验幼儿园	B0FFFDAQ3U	陕西省	西安市	长安区
2	[]	108.905968	34.263305	爱贝儿幼儿园(团结东路)	B0FFQZHN	陕西省	西安市	莲湖区
3	[]	108.947608	34.221570	WOW魔兽VR电玩CLUB	B0FFL23PBX	陕西省	西安市	雁塔区
4	[]	108.215349	34.162494	周至西街小学	B001D091JR	陕西省	西安市	周至县

C:\Users\97505\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:238: RuntimeWarning:

Glyph 26597 missing from current font.

C:\Users\97505\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:238: RuntimeWarning:

Glyph 35810 missing from current font.

C:\Users\97505\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:238: RuntimeWarning:

Glyph 20851 missing from current font.

C:\Users\97505\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:238: RuntimeWarning:

Glyph 38190 missing from current font.

C:\Users\97505\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:238: RuntimeWarning:

Glyph 35789 missing from current font.

C:\Users\97505\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:238: RuntimeWarning:

Glyph 25163 missing from current font.

C:\Users\97505\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:238: RuntimeWarning:

Glyph 26426 missing from current font.

C:\Users\97505\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:238: RuntimeWarning:

Glyph 21495 missing from current font.

C:\Users\97505\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:238: RuntimeWarning:

Glyph 35782 missing from current font.

C:\Users\97505\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:238: RuntimeWarning:

Glyph 21035 missing from current font.

C:\Users\97505\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:201: RuntimeWarning:

Glyph 26597 missing from current font.

C:\Users\97505\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:201: RuntimeWarning:

Glyph 35810 missing from current font.

C:\Users\97505\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:201: RuntimeWarning:

Glyph 20851 missing from current font.

C:\Users\97505\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:201: RuntimeWarning:

Glyph 38190 missing from current font.

C:\Users\97505\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:201: RuntimeWarning:

Glyph 35789 missing from current font.

C:\Users\97505\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:201: RuntimeWarning:

Glyph 25163 missing from current font.

C:\Users\97505\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:201: RuntimeWarning:

Glyph 26426 missing from current font.

C:\Users\97505\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:201: RuntimeWarning:

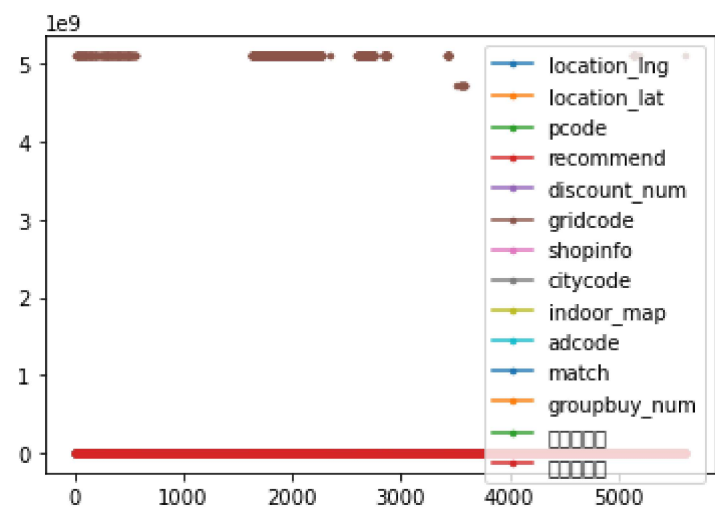
Glyph 21495 missing from current font.

C:\Users\97505\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:201: RuntimeWarning:

Glyph 35782 missing from current font.

C:\Users\97505\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:201: RuntimeWarning:

Glyph 21035 missing from current font.



In [2]:

```
import pickle
import pprint

file=open("C:\\Users\\97505\\Desktop\\bak1(1).pkl","rb")
data=pickle.load(file)
pprint.pprint(data)
file.close()
```

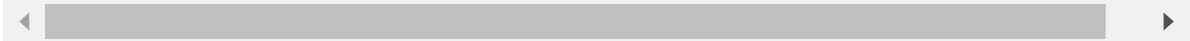
	photo_url	location_lng	location_lat	scene	scene_id	pname \	
0	[]	108.604087	34.089245	西安乾元新景鄠邑店	B0H3472CJK	陕西	
1	[]	108.870518	34.139311	爱华实验幼儿园	B0FFFDAQ3U	陕西	
2	[]	108.905968	34.263305	爱贝尔幼儿园(团结东路)	B0FFFQZHIN		
3	[]	108.947608	34.221570	WOW魔兽VR电玩CLUB	B0FFL23PBX	陕西省	
4	[]	108.215349	34.162494	周至西街小学	B001D091JR	陕西	
...	
5610	[]	108.982508	34.235468	天子幼儿园	B0FFF65CKL	陕西省	
5611	[]	108.988462	34.238835	西部法制报社	B001D11SI2	陕西	
5612	[]	108.986983	34.235096	西铁一中集团西安辅轮中学	B0FFHPAB6K		
5613	[]	108.984079	34.240556	国家法官学院西安司法警察分院	B001D09IMT		
5614	[]	108.296289	34.065326	楼观镇希望小学	B001D08XZN	陕西	

	cityname	adname	townName	address ...	childtype \	
0	西安市	鄠邑区	NaN	钓台路穆南村村口	...	[]
1	西安市	长安区	NaN	栾树大道与环五路交叉口北200米	...	
2	西安市	莲湖区	NaN	团结东路副17号	...	[]
3	西安市	雁塔区	NaN	华旗国际b座2212室	...	202
4	西安市	周至县	NaN	老城西街64号	...	[]
...
5610	西安市	雁塔区	NaN	二环南路东段192青龙小区	...	
5611	西安市	雁塔区	NaN	南二环东段62号伟业都市远景大厦4/5层BCD户	...	
5612	西安市	雁塔区	NaN	青龙北路8号	...	[]
5613	西安市	碑林区	NaN	太乙路街道友谊东路12号	...	
5614	西安市	周至县	全部	延生观路西50米	...	[]

	exit_location	shopid	navi_poiid	groupbuy_num	isin_region	查询关键词	手机号
0	NaN	[]	NaN	NaN	yes	NaN	1
1	NaN	[]	NaN	NaN	yes	NaN	1
2	NaN	[]	NaN	NaN	yes	NaN	1
3	NaN	[]	NaN	NaN	yes	NaN	1
4	NaN	[]	NaN	NaN	yes	NaN	1
...
5610	NaN	[]	NaN	NaN	yes	NaN	0
5611	NaN	[]	NaN	NaN	yes	NaN	0
5612	NaN	[]	NaN	NaN	yes	NaN	0
5613	NaN	[]	NaN	NaN	yes	NaN	0
5614	[]	[]	I49F024003_22	0.0	yes	NaN	0

	行业大类	行业子类
0	汽车销售	汽车销售
1	科教文化服务	学校
2	科教文化服务	学校
3	科教文化服务	科技馆
4	科教文化服务	学校
...
5610	科教文化服务	学校
5611	科教文化服务	传媒机构
5612	科教文化服务	学校
5613	科教文化服务	学校
5614	科教文化服务	学校

[5615 rows x 50 columns]



In [1]:

```

import geopandas as gpd
from matplotlib import pyplot as plt

data = gpd.read_file('C:\\Users\\97505\\Desktop\\xian.shp')#读取磁盘上的矢量文件
#data = gpd.read_file('shapefile/china.gdb', layer='province')#读取gdb中的矢量数据
print(data.crs) # 查看数据对应的投影信息
print(data.head()) # 查看前5行数据
data.plot()
plt.show()#简单展示
poi_gpd.head()
tmp = poi_gpd.copy()
poi_gpd.dropna(axis=0, inplace=True)
poi_gpd.reset_index(drop=True, inplace=True)
poi_gpd.isna().sum()
my_poi = []
for i in tqdm(range(poi_gpd.shape[0])):
    try:
        if poi_gpd['geometry'][i].exterior.coords.xy[0][0]<312800 and poi_gpd['geometry'][i].exterior.coords.xy[1][0]<312800:
            my_poi.append(list(poi_gpd.values[i]))
    except:
        continue
# my_poi_gpd = pd.DataFrame(my_poi, columns=poi_gpd.columns)
y_poi_gpd1=gpd.GeoDataFrame(my_poi, crs=crs_32749, columns=poi_gpd.columns)
my_poi_gpd1.plot(column='Floor', figsize=(18, 18), aspect=1) #提取index为'poi_0_delicacy'的行查看结果

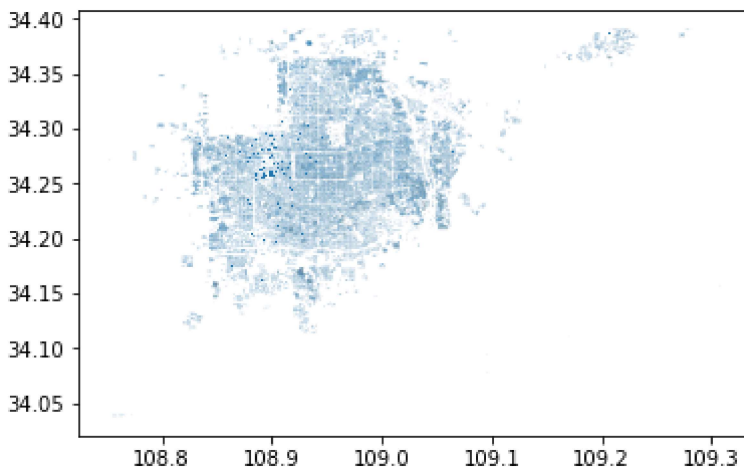
```

None

```

                                geometry
0  POLYGON ((108.75168 34.27170, 108.75191 34.271...
1  POLYGON ((108.75501 34.03893, 108.75607 34.038...
2  POLYGON ((108.75498 34.03852, 108.75547 34.038...
3  POLYGON ((108.75540 34.03874, 108.75548 34.038...
4  POLYGON ((108.75640 34.03894, 108.75657 34.038...

```



```

-----
NameError                                Traceback (most recent call last)
<ipython-input-1-e86a364b5c90> in <module>
      8 data.plot()
      9 plt.show()#简单展示
--> 10 poi_gpd.head()
     11 tmp = poi_gpd.copy()
     12 poi_gpd.dropna(axis=0, inplace=True)

```



```
NameError: name 'poi_gpd' is not defined
```



In [11]:

```

import numpy as np
import os

def compute_euclidean_distance(point, centroid):
    return np.sqrt(np.sum((point - centroid)**2))

def assign_label_cluster(distance, data_point, centroids):
    index_of_minimum = min(distance, key=distance.get)
    return [index_of_minimum, data_point, centroids[index_of_minimum]]

def compute_new_centroids(cluster_label, centroids):
    return np.array(cluster_label + centroids)/2

def iterate_k_means(data_points, centroids, total_iteration):
    label = []
    cluster_label = []
    total_points = len(data_points)
    k = len(centroids)

    for iteration in range(0, total_iteration):
        for index_point in range(0, total_points):
            distance = {}
            for index_centroid in range(0, k):
                distance[index_centroid] = compute_euclidean_distance(data_points[index_point], centroids[index_centroid])
            label = assign_label_cluster(distance, data_points[index_point], centroids)
            centroids[label[0]] = compute_new_centroids(label[1], centroids[label[0]])

            if iteration == (total_iteration - 1):
                cluster_label.append(label)

    return [cluster_label, centroids]

def print_label_data(result):
    print("Result of k-Means Clustering: \n")
    for data in result[0]:
        print("data point: {}".format(data[1]))
        print("cluster number: {} \n".format(data[0]))
    print("Last centroids position: \n {}".format(result[1]))

def create_centroids():
    centroids = []
    centroids.append([5.0, 0.0])
    centroids.append([45.0, 70.0])
    centroids.append([50.0, 90.0])
    return np.array(centroids)

if __name__ == "__main__":
    filename = "C:\\Users\\97505\\Desktop\\bbbb.txt"
    data_points = np.genfromtxt(filename, delimiter=",")
    centroids = create_centroids()
    total_iteration = 100

    [cluster_label, new_centroids] = iterate_k_means(data_points, centroids, total_iteration)
    print_label_data([cluster_label, new_centroids])
    print()

```

Result of k-Means Clustering:

data point: nan

cluster number: 0

data point: nan
cluster number: 0

data point: nan
cluster number: 0

data point: nan
cluster number: 0

data point: nan
cluster number: 0

data point: nan
cluster number: 0

data point: nan
cluster number: 0

data point: nan
cluster number: 0

data point: nan
cluster number: 0

data point: nan
cluster number: 0

data point: nan
cluster number: 0

data point: nan
cluster number: 0

data point: nan
cluster number: 0

data point: nan
cluster number: 0

data point: nan
cluster number: 0

data point: nan
cluster number: 0

data point: nan
cluster number: 0

data point: nan
cluster number: 0

data point: nan
cluster number: 0

Last centroids position:
[[nan nan]
[45. 70.]
[50. 90.]]

In [16]:

```

# K-means Practice
# Data x1=(1 1 4), x2=(2 2 1), x3=(7 5 6), x4=(6 5 5), x5=(1 0 2), x6=(9 8 7), x7=(4 5 7), x8=(2 1 2)
# 3D => 3 Dimention Data
# Amin Zayeromali ==> amin.zayeromali@gmail.com Likedin Profile : https://ir.linkedin.com/in/aminzayeromali

# Import Python Library
import pandas as pd
from sklearn.cluster import KMeans
from matplotlib import pyplot as plt

# Use Pandas lib for Create Dataframe
MyData = pd.DataFrame(
    [[1, 1, 4],
     [2, 2, 1],
     [7, 5, 6],
     [6, 5, 5],
     [1, 0, 2],
     [9, 8, 7],
     [4, 5, 7],
     [2, 1, 2]], columns=['F1', 'F2', 'F3'])

# Use Matplotlib For plotting data
# plot 3D Data
fig = plt.figure()
ax = plt.axes(projection='3d')
ax = plt.axes(projection='3d')
ax.scatter3D(MyData['F1'], MyData['F2'], MyData['F3'])

# Use K-Means Alogrithm from Sklearn Lib
K = 2 # Number of Clusters
km = KMeans(K)
clusts = km.fit_predict(MyData)

# Add Clusters Label to DataFrame
MyData['clusts'] = clusts
print(MyData)

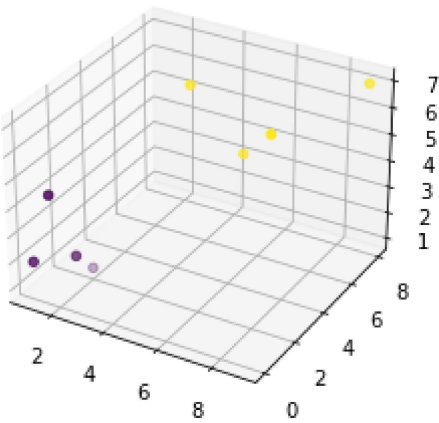
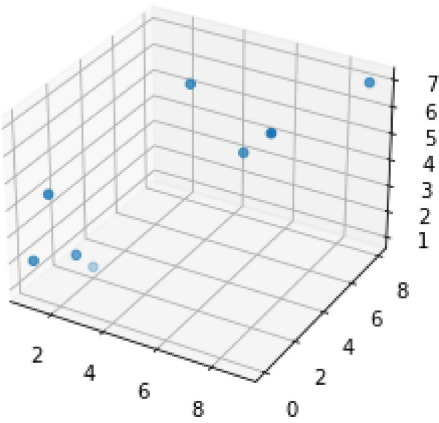
# Use Matplotlib For plotting Clustered data
fig = plt.figure()
ax = plt.axes(projection='3d')
ax = plt.axes(projection='3d')
ax.scatter3D(MyData['F1'], MyData['F2'], MyData['F3'], c=MyData['clusts'])

```

	F1	F2	F3	clusts
0	1	1	4	0
1	2	2	1	0
2	7	5	6	1
3	6	5	5	1
4	1	0	2	0
5	9	8	7	1
6	4	5	7	1
7	2	1	2	0

Out[16]:

<mpl_toolkits.mplot3d.art3d.Path3DCollection at 0x1b5f24366d0>



In [12]:

```

import plotly.express as px
import geopandas as gpd
poi_gpd=gpd.read_file('C:\\Users\\97505\\Desktop\\xian.shp', crs='EPSG:32749') #读取存储的.shp格式文件
poi_gpd.loc[:, :].plot(column='Floor', figsize=(55, 55)) #提取index为'poi_0_delicacy'的行查看结果
my_poi = []
for i in tqdm(range(poi_gpd.shape[0])):
    try:
        if poi_gpd['geometry'][i].exterior.coords.xy[0][0]<312800 and poi_gpd['geometry'][i].exterior.coords.xy[1][0]>312800:
            my_poi.append(list(poi_gpd.values[i]))
    except:
        continue
# my_poi_gpd = pd.DataFrame(my_poi, columns=poi_gpd.columns)
my_poi_gpd1=gpd.GeoDataFrame(my_poi, crs=crs_32749, columns=poi_gpd.columns)
my_poi_gpd1.plot(column='Floor', figsize=(5, 5), aspect=1) #提取index为'poi_0_delicacy'的行查看结果
from sklearn.cluster import KMeans
estimator = KMeans(n_clusters=5)
estimator.fit(my_poi_gpd1['Floor'].values.reshape(-1, 1))
label_pred = estimator.labels_
label_pred
my_poi_gpd1['lable'] = label_pred
my_poi_gpd1.plot(column='lable', figsize=(5, 5), aspect=10)
my_poi_gpd1.head()

```

```

-----
KeyError                                Traceback (most recent call last)
~\anaconda3\lib\site-packages\pandas\core\indexes\base.py in get_loc(self,
    key, method, tolerance)
    3079         try:
-> 3080             return self._engine.get_loc(casted_key)
    3081         except KeyError as err:

pandas\_libs\index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas\_libs\index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

```

KeyError: 'Floor'

The above exception was the direct cause of the following exception:

```

KeyError                                Traceback (most recent call last)
<ipython-input-12-ad95510abc15> in <module>
      2 import geopandas as gpd
      3 poi_gpd=gpd.read_file('C:\\Users\\97505\\Desktop\\xian.shp', crs='EPSG:32749') #读取存储的.shp格式文件
----> 4 poi_gpd.loc[:, :].plot(column='Floor', figsize=(55, 55)) #提取index为'poi_0_delicacy'的行查看结果
      5 my_poi = []
      6 for i in tqdm(range(poi_gpd.shape[0])):

~\anaconda3\lib\site-packages\geopandas\plotting.py in __call__(self, *args, **kwargs)

```

```

948         kind = kwargs.pop("kind", "geo")
949         if kind == "geo":
--> 950             return plot_dataframe(data, *args, **kwargs)
951         if kind in self._pandas_kinds:
952             # Access pandas plots

```

```

~\anaconda3\lib\site-packages\geopandas\plotting.py in plot_dataframe(df, column, cmap, color, ax, cax, categorical, legend, scheme, k, vmin, vmax, markersize, figsize, legend_kwds, categories, classification_kwds, missing_kwds, aspect, **style_kwds)

```

```

714         values = values.reindex(df.index)
715     else:
--> 716         values = df[column]
717
718     if pd.api.types.is_categorical_dtype(values.dtype):

```

```

~\anaconda3\lib\site-packages\geopandas\geodataframe.py in __getitem__(self, key)

```

```

1325         GeoDataFrame.
1326         """
-> 1327         result = super().__getitem__(key)
1328         geo_col = self._geometry_column_name
1329         if isinstance(result, Series) and isinstance(result.dtype, GeometryDtype):

```

```

~\anaconda3\lib\site-packages\pandas\core\frame.py in __getitem__(self, key)

```

```

3022         if self.columns.nlevels > 1:
3023             return self._getitem_multilevel(key)
-> 3024         indexer = self.columns.get_loc(key)
3025         if is_integer(indexer):
3026             indexer = [indexer]

```

```

~\anaconda3\lib\site-packages\pandas\core\indexes\base.py in get_loc(self, key, method, tolerance)

```

```

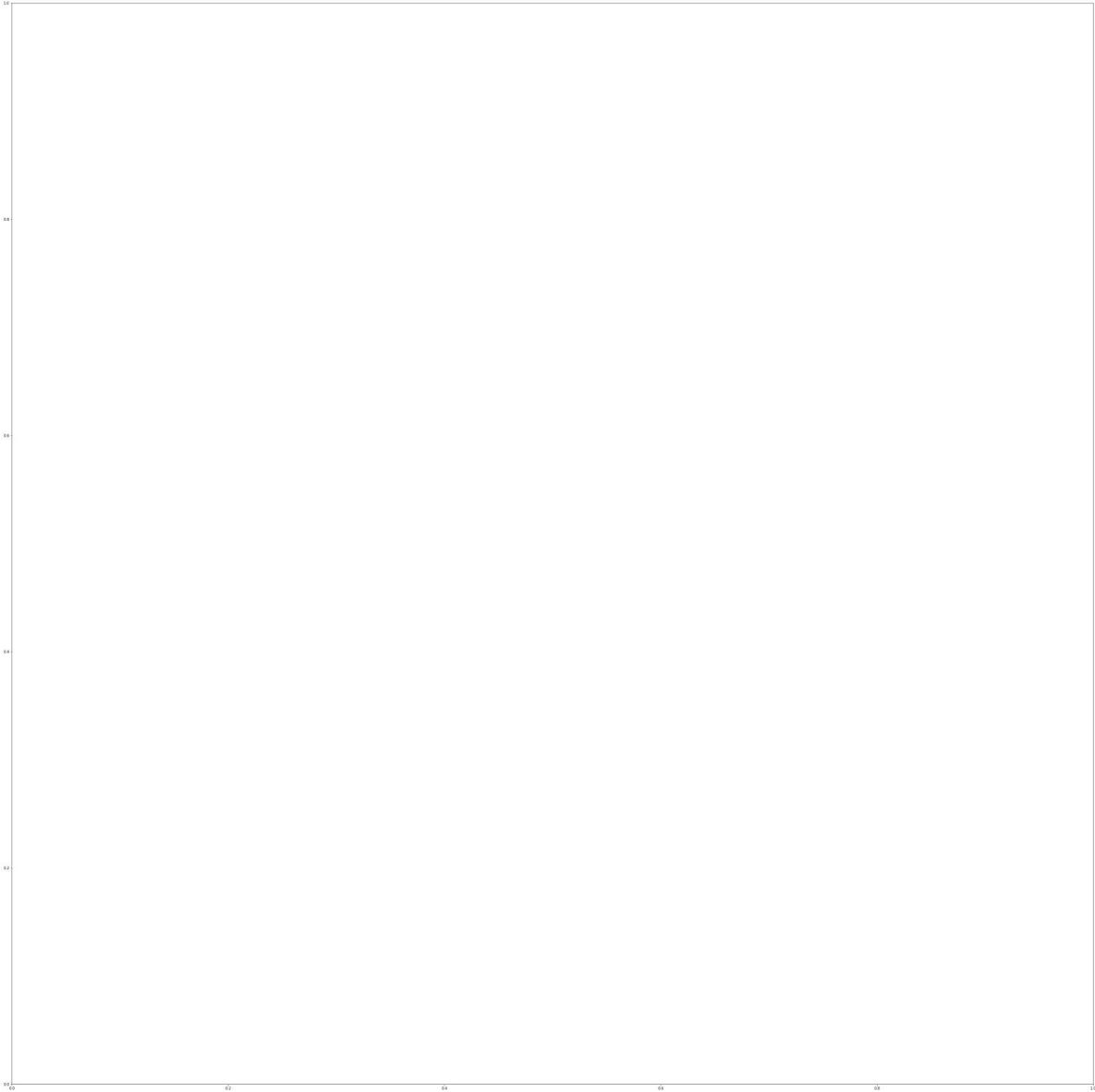
3080         return self._engine.get_loc(casted_key)
3081     except KeyError as err:
-> 3082         raise KeyError(key) from err
3083
3084         if tolerance is not None:

```

```

KeyError: 'Floor'

```

In [21]:

```
# -*- coding: utf-8 -*-

import shapefile# 使用pyshp

file = shapefile.Reader("C:\\Users\\97505\\Desktop\\xian.shp")#读取
#读取元数据
print(str(file.shapeType)) # 输出shp类型
print(file.encoding)# 输出shp文件编码
print(file.bbox) # 输出shp的文件范围（外包矩形）
print(file.numRecords) # 输出shp文件的要素数据
print(file.fields)# 输出所有字段信息
# print(file.records()) # 输出所有属性表
```

5
utf-8
[108.7516827660001, 34.03824211130012, 109.30779195199995, 34.39031987260006]
None
[]