

自行分析 不同行业的地理位置分布是否有差异

```
In [ ]: from sklearn.decomposition import PCA
import numpy as np
import pandas as pd
import scipy as sp
from scipy import stats
```

```
In [ ]: filename = 'poi_2_shopping.json'
with open(os.path.join(path, filename) , 'r', encoding='utf8') as fp:
    json_data = json.load(fp)
    df1 = [str(i) for i in json_data]
    pd.DataFrame(df1).to_csv('data/shopping.csv', index=False)
```

```
In [ ]: filename = 'poi_0_delicacy.json'
with open(os.path.join(path, filename) , 'r', encoding='utf8') as fp:
    json_data = json.load(fp)
    df1 = [str(i) for i in json_data]
    pd.DataFrame(df1).to_csv('data/delicacy.csv', index=False)
```

```
In [ ]: poi_shopping=util_poi.csv2df('data/shopping.csv')
poi_delicacy=util_poi.csv2df('data/delicacy.csv')
```

incorrect format of data_row number:0

.csv to DataFrame is completed!

incorrect format of data_row number:0

.csv to DataFrame is completed!

```
In [ ]: poi_fieldsExtraction=poi_shopping.loc[:,fields_extraction]
poi_geodf=poi_fieldsExtraction.copy(deep=True)
poi_geodf['geometry']=poi_geodf.apply(lambda row:Point(row.location_lng,row.location_lat),axis=1)
crs_32749=CRS('epsg:32749') #配置坐标系统, 参考: https://spatialreference.org/
poishopping_gpd=gpd.GeoDataFrame(poi_geodf,crs=crs_32749)

poi_fieldsExtraction=poi_delicacy.loc[:,fields_extraction]
poi_geodf=poi_fieldsExtraction.copy(deep=True)
poi_geodf['geometry']=poi_geodf.apply(lambda row:Point(row.location_lng,row.location_lat),axis=1)
crs_32749=CRS('epsg:32749') #配置坐标系统, 参考: https://spatialreference.org/
poidelicacy_gpd=gpd.GeoDataFrame(poi_geodf,crs=crs_32749)
```

```
In [ ]: pca = PCA(n_components=1)
shopping = pca.fit_transform(poishopping_gpd.iloc[:,1:3].values)
delicacy = pca.fit_transform(poidelicacy_gpd.iloc[:,1:3].values)
```

```
In [ ]: dic = {
    'shopping':shopping.reshape(1,-1)[0][:9451],
    'delicacy':delicacy.reshape(1,-1)[0]
}
```

```
In [ ]: data = pd.DataFrame(dic)
data.head()
```

```
Out[ ]:   shopping  delicacy
0  0.295044 -0.356739
1  0.293029 -0.363445
2  0.294325 -0.342711
3  0.301760 -0.354382
4  0.307980 -0.312305
```

```
In [ ]: # 样本数组
weight_A = data["shopping"]
weight_B = data["delicacy"]
# 转化为数组型
weight_A = np.array(weight_A)
weight_B = np.array(weight_B)
# 计算差值
diff = weight_A - weight_B
diff
```

```
Out [ ]: array([ 0.65178344,  0.65647488,  0.63703561, ..., -0.53707246,
               -0.53684127, -0.53430948])
```

```
In [ ]: # 进行单样本t检验
stats.ttest_lsamp(diff, 0)
```

```
Out [ ]: Ttest_lsampResult(statistic=15.762028153238543, pvalue=2.86281582184302e-55)
```

```
In [ ]: # stats中的配对样本t检验方法，直接计算配对样本t检验的t值和p值
paired_tvalue, paried_pvalue = stats.ttest_rel(weight_B, weight_A)
paired_tvalue, paried_pvalue
```

```
Out [ ]: (-15.762028153238543, 2.86281582184302e-55)
```

```
In [ ]: # 按公式计算p值
paired_alpha = stats.t.cdf(abs(paired_tvalue), df = 9)
paired_p_value = (1 - paired_alpha) * 2
paired_p_value
```

```
Out [ ]: 7.330330253907391e-08
```

```
In [ ]: alpha = 0.05
if paired_p_value <= alpha:
    print("拒绝零假设, 在0.05的显著性水平上有和显地理位置著性差异")
else:
    print("接受零假设, 在0.05的显著性水平上没有和地理位置有显著性差异")
```

拒绝零假设, 在0.05的显著性水平上有和没有显地理位置著性差异