对于东京湾区，我们从 https://www.toukei.metro.tokyo.lg.jp/tnenkan/tn-index.htm 上搜集了其相关的数据，对其采用相同的方法进行分析。选取零售、企业数量等在粤港澳大湾区中对GDP的相关性最高的因素，结合东京湾区实际情况，另外选取了社会福祉机构作为因素。为简化模型，故仅使用这三种因素对东京湾区的GDP建立模型

import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import os  
import statsmodels.api as sm

def extract\_cell\_data(folder\_path, column, row):  
 data = {}  
 for filename in os.listdir(folder\_path):  
 if filename.endswith('.csv'):  
 file\_path = os.path.join(folder\_path, filename)  
 df = pd.read\_csv(file\_path)  
 cell\_data = df.iloc[row-1, column-1]  
 data[os.path.splitext(filename)[0]] = cell\_data  
 return data  
  
folders = {  
 'GDP': os.path.join(os.getcwd(),'jp\_data','GDP'),  
 'industry': os.path.join(os.getcwd(),'jp\_data','industry'),  
 'retailing': os.path.join(os.getcwd(),'jp\_data','retailing'),  
 'Welfare\_Facilities': os.path.join(os.getcwd(),'jp\_data','Welfare\_Facilities')  
}  
dataframe\_dict = {}  
processed\_data = pd.DataFrame()  
for folder\_name, folder\_path in folders.items():  
 if folder\_name == 'GDP':  
 data = extract\_cell\_data(folder\_path, 3, 7)  
 elif folder\_name == 'industry':  
 data = extract\_cell\_data(folder\_path, 6, 2)  
 elif folder\_name == 'retailing':  
 data = extract\_cell\_data(folder\_path, 6, 2)  
 elif folder\_name == 'Welfare\_Facilities':  
 data = extract\_cell\_data(folder\_path, 4, 1)  
 dataframe\_dict = pd.DataFrame.from\_dict(data, orient='index', columns=[folder\_name])  
 processed\_data[folder\_name] = dataframe\_dict[folder\_name]  
processed\_data.index = range(2015,2023)  
processed\_data.to\_csv('combined\_data.csv')  
print(processed\_data)

GDP industry retailing Welfare\_Facilities  
2015 39123.8 9415.0 84067.0 4644  
2016 39123.8 22302.0 84067.0 5147  
2017 43773.8 8256.0 84067.0 5468  
2018 43924.5 7837.0 84067.0 5883  
2019 45013.9 7425.0 86582 6760  
2020 45013.9 7450.0 86582 7056  
2021 43405.0 7450.0 86582 7363  
2022 43540.5 7450.0 82756.0 7598

X = processed\_data[['industry', 'retailing', 'Welfare\_Facilities']]  
Y = processed\_data['GDP']  
print(X.dtypes)  
processed\_data['industry'] = pd.to\_numeric(processed\_data['industry'], errors='coerce')  
processed\_data['retailing'] = pd.to\_numeric(processed\_data['retailing'], errors='coerce')  
processed\_data['Welfare\_Facilities'] = pd.to\_numeric(processed\_data['Welfare\_Facilities'], errors='coerce')  
processed\_data['GDP'] = pd.to\_numeric(processed\_data['GDP'], errors='coerce')  
X = sm.add\_constant(X)  
model = sm.OLS(Y, X)  
results = model.fit()  
  
print(results.summary())

industry float64  
retailing float64  
Welfare\_Facilities int64  
dtype: object  
 OLS Regression Results   
==============================================================================  
Dep. Variable: GDP R-squared: 0.716  
Model: OLS Adj. R-squared: 0.504  
Method: Least Squares F-statistic: 3.367  
Date: Fri, 08 Nov 2024 Prob (F-statistic): 0.136  
Time: 13:13:39 Log-Likelihood: -68.004  
No. Observations: 8 AIC: 144.0  
Df Residuals: 4 BIC: 144.3  
Df Model: 3   
Covariance Type: nonrobust   
======================================================================================  
 coef std err t P>|t| [0.025 0.975]  
--------------------------------------------------------------------------------------  
const 1.671e+04 3.81e+04 0.439 0.683 -8.9e+04 1.22e+05  
industry -0.2129 0.144 -1.476 0.214 -0.613 0.187  
retailing 0.2641 0.459 0.576 0.596 -1.009 1.537  
Welfare\_Facilities 0.9314 0.701 1.328 0.255 -1.016 2.879  
==============================================================================  
Omnibus: 0.848 Durbin-Watson: 1.050  
Prob(Omnibus): 0.654 Jarque-Bera (JB): 0.670  
Skew: -0.502 Prob(JB): 0.715  
Kurtosis: 1.998 Cond. No. 5.48e+06  
==============================================================================  
  
Notes:  
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.  
[2] The condition number is large, 5.48e+06. This might indicate that there are  
strong multicollinearity or other numerical problems.  
  
  
C:\Users\EricW\anaconda3\envs\modeling\Lib\site-packages\scipy\stats\\_axis\_nan\_policy.py:418: UserWarning: `kurtosistest` p-value may be inaccurate with fewer than 20 observations; only n=8 observations were given.  
 return hypotest\_fun\_in(\*args, \*\*kwds)