Coursework_BohaoSu

April 5, 2024

1-Introduction | 2-Literature Review | 3-Research Question | 4-Presentation of Data | 5-Methodology | 6-Results | 7-Discussion | 8-Conclusion | Bibliography(#section9) | Appendix

0.1 1 | Introduction

This paper aims to investigate the impact of the Seed Enterprise Investment Scheme (SEIS) on the development and nurturing of businesses in London. Specifically, it explores whether the introduction of SEIS in 2012 has provided positive assistance to start-up enterprises in London. To achieve this, we employ casual inference and regression discontinuity methods to analyze Business Demographics and Survival Rates data for the period 2002 to 2021.

The data used in this analysis is the London Business Demographics and Survival Rates from the London Datastore. The data is available at the following link: `https://data.london.gov.uk/dataset/business-demographics-and-survival-rates-borough`, containing all the data of enterprise births, deaths, active enterprises and survival rates across boroughs from 2002 to 2021.

Findings at the boroughs level indicates that ————————

the SEIS has had a positive impact on the number of active enterprises in London. The results show that the number of active enterprises in London has increased by 10% since the introduction of the SEIS in 2012. This suggests that the SEIS has been successful in encouraging the development of new businesses in London.

[2]:
```python
# import all the necessary libraries
import pandas as pd
import geopandas as gpd
from shapely.geometry import Point
import matplotlib.pyplot as plt
import folium
import contextily as ctx

import os
import glob
```

1.Introduction | 2.Literature Review | 3.Research Question | 4.Presentation of Data | 5.Methodology | 6.Results | 7.Discussion | 8.Conclusion | Bibliography(#section9) | Appendix

0.2 2 | Literature Review

The exploration of traffic congestion patterns and characteristics in urban areas like London has been a subject of considerable research interest in transportation studies.

Most common approach is the spatio-temporal analysis techniques to identify clusters and trends in congestion levels. Studies(Yang and Wang, 2020) have employed similar methodologies to analyze traffic congestion patterns in cities, providing valuable insights for urban planners and policymakers.

Regarding the temporal characteristics of congestion, research has investigated the influence of different time measures on traffic patterns. For instance, some studies have examined the impact of weekdays versus weekends and the segmentation within a day on congestion levels(Wen, Sun and Zhang, 2014), highlighting the importance of considering temporal factors in understanding and mitigating traffic congestion in urban areas.

Furthermore, techniques such as machine learning algorithms and time series analysis have been employed to forecast congestion levels in the short term(Elfar, Talebpour and Mahmassani, 2018).

In summary, existing literature provides a foundation for the exploration of London's traffic congestion, offering insights into spatial and temporal patterns as well as predictive modeling techniques. Building upon this body of knowledge, the present study aims to contribute to our understanding of the transport situation in London during the specified period.

1.Introduction | 2.Literature Review | 3.Research Question | 4.Presentation of Data | 5.Methodology | 6.Results | 7.Discussion | 8.Conclusion | Bibliography(#section9) | Appendix

0.3 3 | Research Question

Even though government policies do play a vital role in shaping the entrepreneurial ecosystem, the key factor for a startup's surviving should still be the macroeconomic environment including supply-demand equilibrium, industry potential, and local currency.

Therefore, it is essential to exclude all of those factors off when evaluating the effectiveness of SEIS policies, seeking to unravel the extent to which this policy initiative has influenced the birth rates and survival rates of enterprises, furtherly in the different boroughs.

With these objectives in mind, we formulate the following research questions:

- RQ1: Is there a causal relationship between SEIS and enterprise growth in London?
- RQ2: Did the introduction of SEIS in 2012 significantly improve enterprise birth rates and survival rates in London?
- RQ3: Which borough in London experienced the most significant increase in start-up activity after the implementation of SEIS in 2012?

0.4 4 | Presentation of Data

Data Description

There are three datasets used in this analysis:

- London Boroughs(.gpkg): Accessed from London Datastore, this geopackage dataset contains information about the 32 boroughs in London, including the name of the borough, the area it covers.
- Business Demographics(.csv): This dataset contains information about the number of active enterprises, along with births and deaths rates from 2004 to 2022.

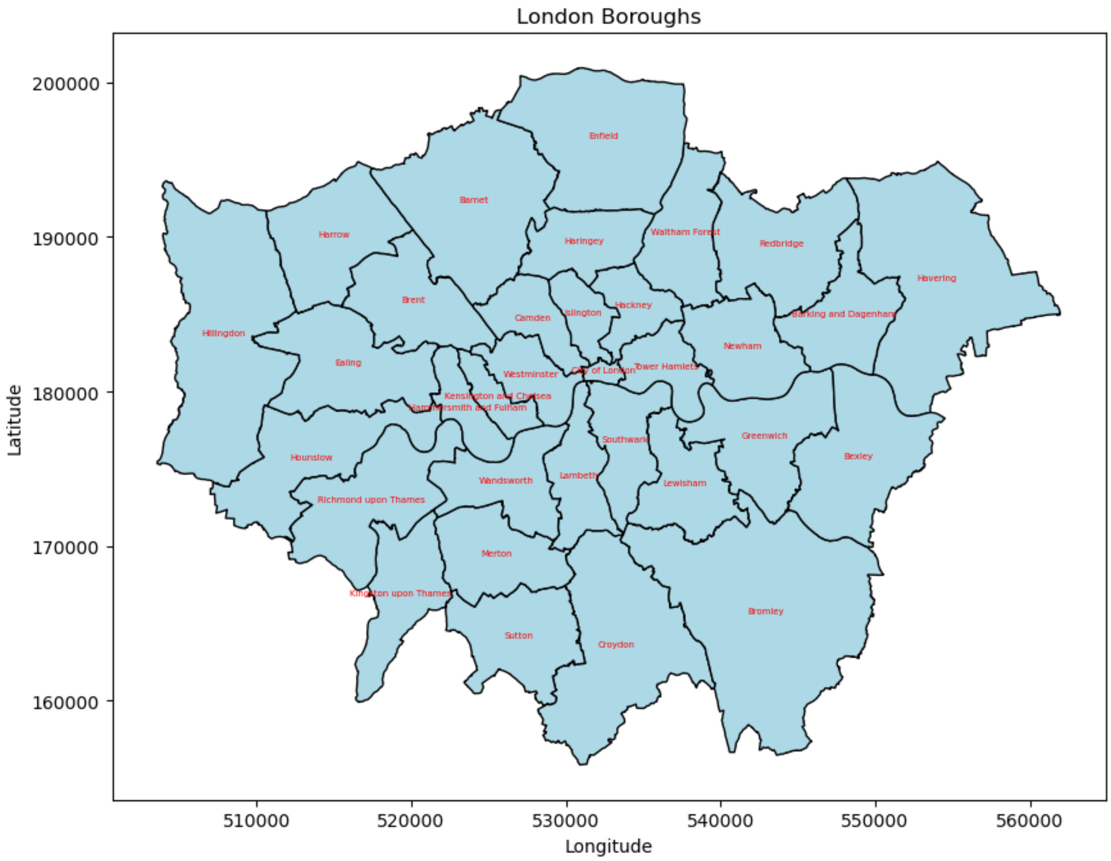
- Business Survival Rates(.csv): active enterprises, and survival rates across boroughs from 2002 to 2021.

[7]:
```python
# reading All data and have a look
# read London Boroug gpkg file
LondonBorough = gpd.read_file('Data/London_Boroughs.gpkg')

# read London Business Demographics data
Demographic = pd.read_csv('Data/business-demographics.csv')

# read London Business Survival Rates data
Survival = pd.read_csv('Data/business-survival-rates.csv')
```

[8]:
```python
LondonBorough.sample(5)
```

[8]:
```
    objectid            name   gss_code  hectares  nonld_area ons_inner  \
7          8          Harrow  E09000015  5046.330       0.000         F
14        15  Waltham Forest  E09000031  3880.793       0.000         F
3          4        Hounslow  E09000018  5658.541      60.755         F
19        21          Merton  E09000024  3762.466       0.000         F
26        27          Camden  E09000007  2178.932       0.000         T

   sub_2011                                           geometry
7      West  POLYGON ((515767.200 186062.800, 515761.000 18…
14     East  POLYGON ((539923.100 191863.100, 539928.100 19…
3      West  POLYGON ((509703.400 175356.600, 509712.600 17…
19    South  POLYGON ((529906.200 167417.300, 529902.200 16…
26  Central  POLYGON ((528840.200 187217.800, 528834.600 18…
```

[13]:
```python
# plot the map of London Boroughs and mark the name of each Borough
fig, ax = plt.subplots(figsize=(10, 10))
LondonBorough.plot(ax=ax, color='lightblue', edgecolor='k')
for x, y, label in zip(LondonBorough.geometry.centroid.x, LondonBorough.
 ↪geometry.centroid.y, LondonBorough['name']):
    ax.text(x, y, label, fontsize=5, ha='center', color='red')
plt.title('London Boroughs')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.show()
```



[14]:
```
Demographic.sample(5)
```

[14]:
```
                               area  year  active_enterprises  births  \
434  E09000027  Richmond upon Thames  2012               12705    1595
164  E09000012               Hackney  2007               10160    1520
169  E09000017            Hillingdon  2007                9525    1220
393  E12000002            North West  2011              231345   25695
789  E09000025                Newham  2019               15865    3220

     birth_rate  deaths  death_rate
434        12.6    1275        10.0
164        15.0    1105        10.9
169        12.8     990        10.4
393        11.1   25305        10.9
789        20.3    3220        20.3
```

[30]:
```
Survival.sample(5)
```

[30]:
```
          code          area  year  births  1_year_survival_number  \
289  E13000002  Outer London  2007   25130                   24095
```

```
68   E09000018      Hounslow  2003    1225                       1115
654  E12000008    South East  2014   51280                      47775
890  E09000024        Merton  2019    1770                       1665
935  E09000018      Hounslow  2020    1950                       1835

     1_year_survival_rate 2_year_survival_number 2_year_survival_rate  \
289                  95.9                  20095                   80
68                   91.0                    930                 75.9
654                  93.2                  39900                 77.8
890                  94.1                   1315                 74.3
935                  94.1                   1440                 73.8

    3_year_survival_number 3_year_survival_rate 4_year_survival_number  \
289                  15095                 60.1                  12390
68                     745                 60.8                    610
654                  32615                 63.6                  26875
890                   1040                 58.8                      :
935                      :                    :                      :

    4_year_survival_rate 5_year_survival_number 5_year_survival_rate
289                 49.3                  10645                 42.4
68                  49.8                    500                 40.8
654                 52.4                  23400                 45.6
890                    :                      :                    :
935                    :                      :                    :
```

Data Preprocessing

[4]:
```python
# Define all the IDs that are in the metadata
#  metadata    ID
all_ids = metadata['ID'].unique()
```

[7]:
```python
# Define a new function to get all filenames in the folder
def get_filenames_without_extension(directory):
    filenames = []
    for filename in os.listdir(directory):
        # use os.path.splitext to split the filename and extension
        #   os.path.splitext
        base_name, _ = os.path.splitext(filename)
        filenames.append(base_name)
    return filenames

directory_path = 'Data/CSV3-180215-180316'
filenames = get_filenames_without_extension(directory_path)
```

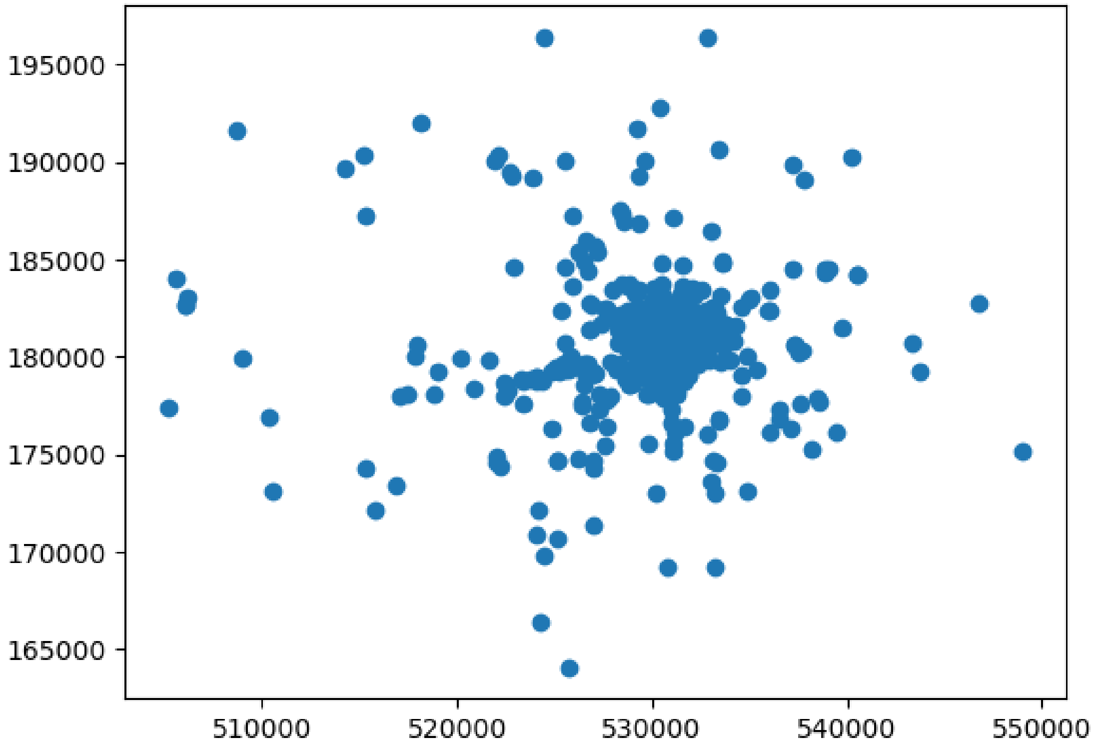
[8]:
```python
setA = set(filenames)
setB = set(all_ids)
unique_ids = setA ^ setB
```

```python
unique_ids
```

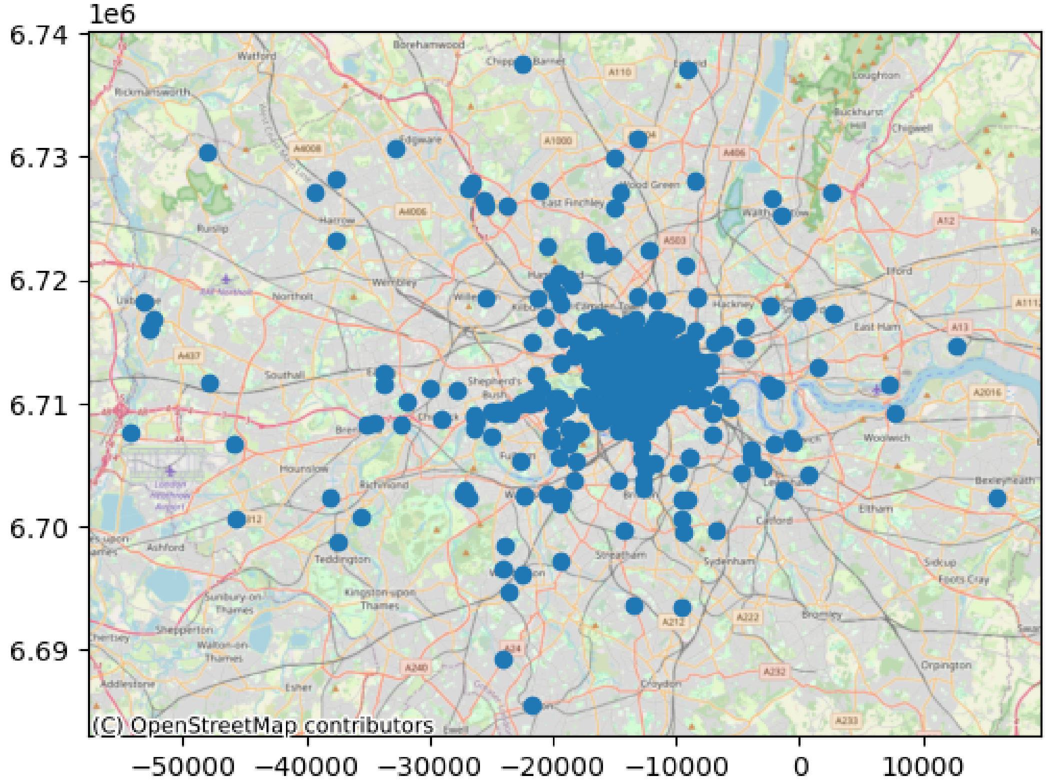
[8]:
```
{'01-018',
 '01-470',
 '01-850',
 '01-857',
 '01-858',
 '01-859',
 '01-861',
 '01-863',
 '02-025',
 '02-079',
 '03-850',
 '04-102',
 '04-194',
 '05-852',
 '09-857',
 '12-173',
 '13-077',
 '13-172',
 '18-088',
 '23-850',
 '28-262',
 '32-019',
 '32-194',
 '32-194-3'}
```

[9]:
```python
#  gpkg  gdf
# Read the gpkg file into a GeoDataFrame and then display it on the map
gdf = gpd.read_file('Data/cultural_venues_in_GIS_format.gpkg')
gdf.plot()
```

[9]:
```
<Axes: >
```



[11]:
```
# gdf epsg 27700  3857
# Convert the epsg of the gdf from 27700 to 3857
gdf = gdf.to_crs(epsg=3857)

# OSM   gdf
# Display the points in the gdf with OSM as the basemap
ax = gdf.plot()
ctx.add_basemap(ax, source=ctx.providers.OpenStreetMap.Mapnik)
```



0.5 5.Methodology

0.6 6.Results

0.7 7.Discussion

0.8 8.Conclusion