Are later-built high-speed-rail less meaningful?

Using DID to analyze the effect of China's high-speed rails on urban development

---

- Github link: https://github.com/irrationalBI/CASA0006_Assessment
- Word count: 1953
- Operating Environment: Docker-based CASA Computing Environment
- Full runtime: 17 seconds
- Some markdown syntax cannot be printed. Viewing the original file can provide better display effects.

In [45]:
```
# Import all required packages
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
import statsmodels.formula.api as smf
from statsmodels.regression.mixed_linear_model import MixedLM
from scipy import stats
from IPython.display import Image, display
from IPython.display import Math
```

In [2]:
```
# Hiding warnings makes output cleaner
import warnings
warnings.filterwarnings('ignore')
```

---

1 Introduction

China's high-speed-rail (HSR) network has expanded rapidly over the past decade. According to the National Railway Administration of China (https://www.nra.gov.cn), since the opening of the first high-speed-rail in 2008, the Beijing-Tianjin Intercity Railway, China has surpassed 40,000 kilometers in total length of high-speed-rail by 2021.

In earlier years, people believed that HSR had greatly boosted economic development. However, in recent years, doubts have emerged on the Internet regarding the significance of some newly built HSR projects, particularly their economic benefits. Since newly constructed high-speed railways are often situated in economically underdeveloped provinces, especially in some non-plain areas, it is believed that their construction costs are excessively high and their economic promotional impact is limited.

This study aims to examine whether these newly built high-speed railways indeed yield lower economic promotion effects.

2 Literature review

There are a lot of studies on China's HSR. Initial studies focused on the impact of high-speed rail opening on different socio-economic indicators of the city. The most common indicators are GDP, GDP by industry, and GDP per capita. Some studies have developed new indicators and evaluated the impact of high-speed rail on them. For example, the study on the impact of high-speed rail on urban expansion (Deng et al., 2020), mainly used the Differences-in-Differences (DID) method and confirmed that high-speed rail will lead to greater urban expansion.

PSM (Propensity Score Matching) combined with DID can better perform this type of analysis. This method has been used to study urban specialization patterns (Lin, 2017), income distribution (Jin et al., 2022), and urban innovation (Fan and Xu, 2023).

In addition to innovations at the indicator level, some studies also focus on the promotion effect of high-speed rail in different regions and at different opening times. A comparative study of developed and underdeveloped regions shows that HSR has a more significant positive impact on underdeveloped areas with better basic conditions. (Liang et al., 2020). There are also analyses of the regional heterogeneity of the impact of HSR using a wider range of data combined with panel regression (Ke et al., 2017).

3 Research question

The research question is: Among China's high-speed rail lines, do newer lines have weaker positive effects on the cities they pass through than those of earlier-built lines?

It should be made clear that, "newer" does not refer to time, but refers to those high-speed rail lines that were built later due to poor location or high cost. Some short connecting lines newly built in developed areas do not fall into this category.

4 Presentation of data

4.1 Data introduction

The data used in the study is panel data for Chinese prefecture-level cities from 1990 to 2021, compiled based on the China Statistical Yearbook (https://www.stats.gov.cn/sj/ndsj). The dataset was sourced from an online data store (http://www.gis5g.com/data/tjnj?id=267), and comprises over 100 indicators for more than 300 prefecture-level cities.

4.2 Data filtering

4.2.1 Data filtering approach

The study will first use DID to analyze the impact of two HSR lines in different periods and regions, and then compared the two groups of impacts. Therefore, we need to carefully select two appropriate treatment and control groups.

The two selected HSR lines should meet the following requirements:

1. They pass through two distant regions and are opened at a long interval.

2. When the selected line is opened, there should be as few nearby HSR lines as possible to ensure the independence of their impact.
3. The line lengths should be similar to ensure that the coverage of the lines is similar.

As shown in the figure, two lines were selected: the Beijing-Shanghai HSR (opened in June 2011) and the Changsha-Kunming HSR (opened in December 2016). In particular, Shandong Province and Guizhou Province were selected to ensure that the cities in each DID analysis are from the same province.

In [3]:
```
display(Image(url='https://raw.githubusercontent.com/irrationalBI/CASA0006_Assessment/main/da
```

Basemap source: https://en.wikipedia.org/wiki/High-speed_rail_in_China

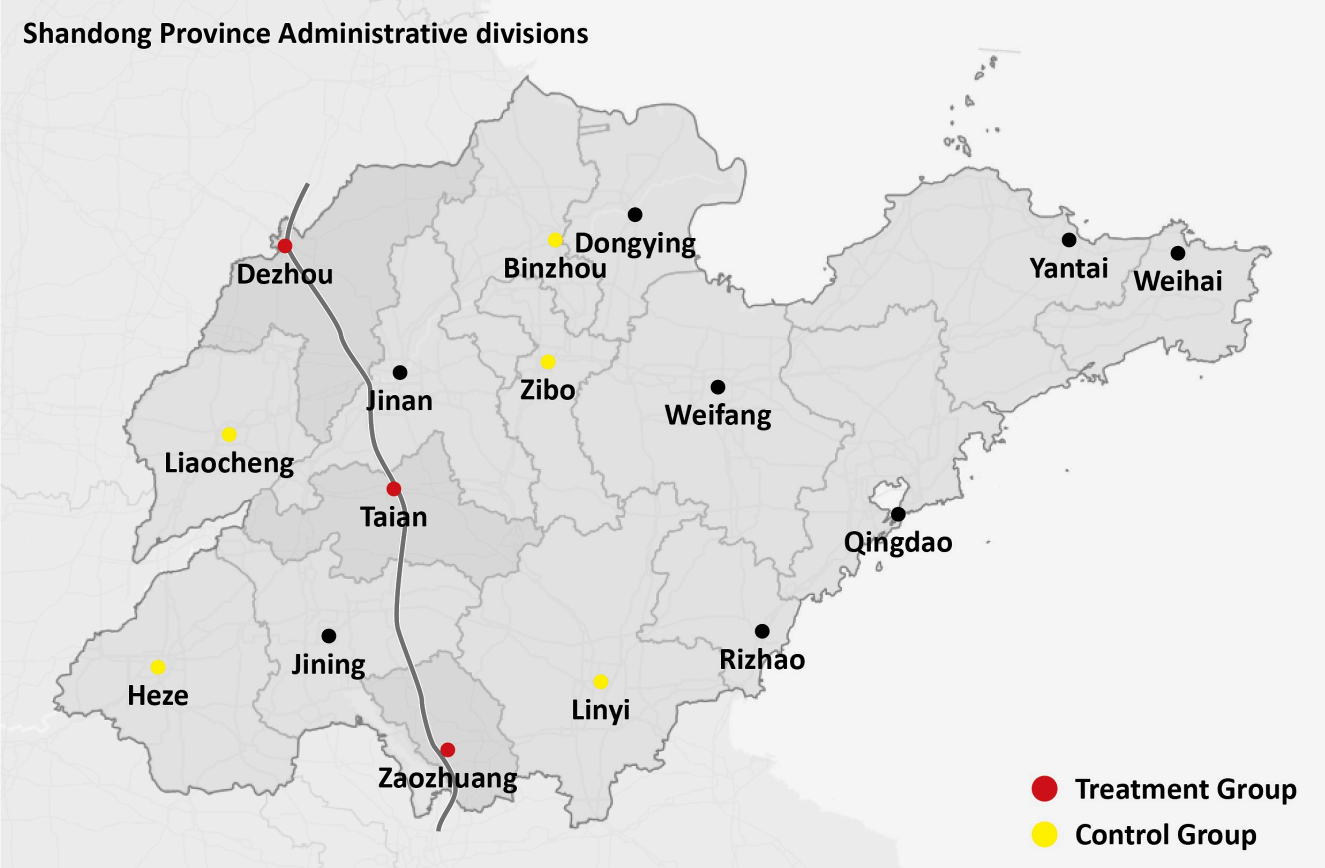
For each province (DID group), The following conditions should be met when selecting cities:

1. The administrative levels of the cities in the treatment and control group are same.
2. The city center of the treatment group has a HSR station, and the city center of the control group is far away enough from the station.
3. The geographical conditions of the cities in the treatment and control group are similar.

As shown in the figure, in the two provinces, red is the treatment group and yellow is the control group. The remaining cities are not included in the analysis for the following reasons:

1. Among the cities in Shandong Province, Jinan and Qingdao are sub-provincial cities, which are different from other prefecture-level cities. The distance between Jining and the HSR line is vague that its impact cannot be clearly defined. The remaining cities are too far away from the treatment group and are all coastal cities, with large geographical differences.

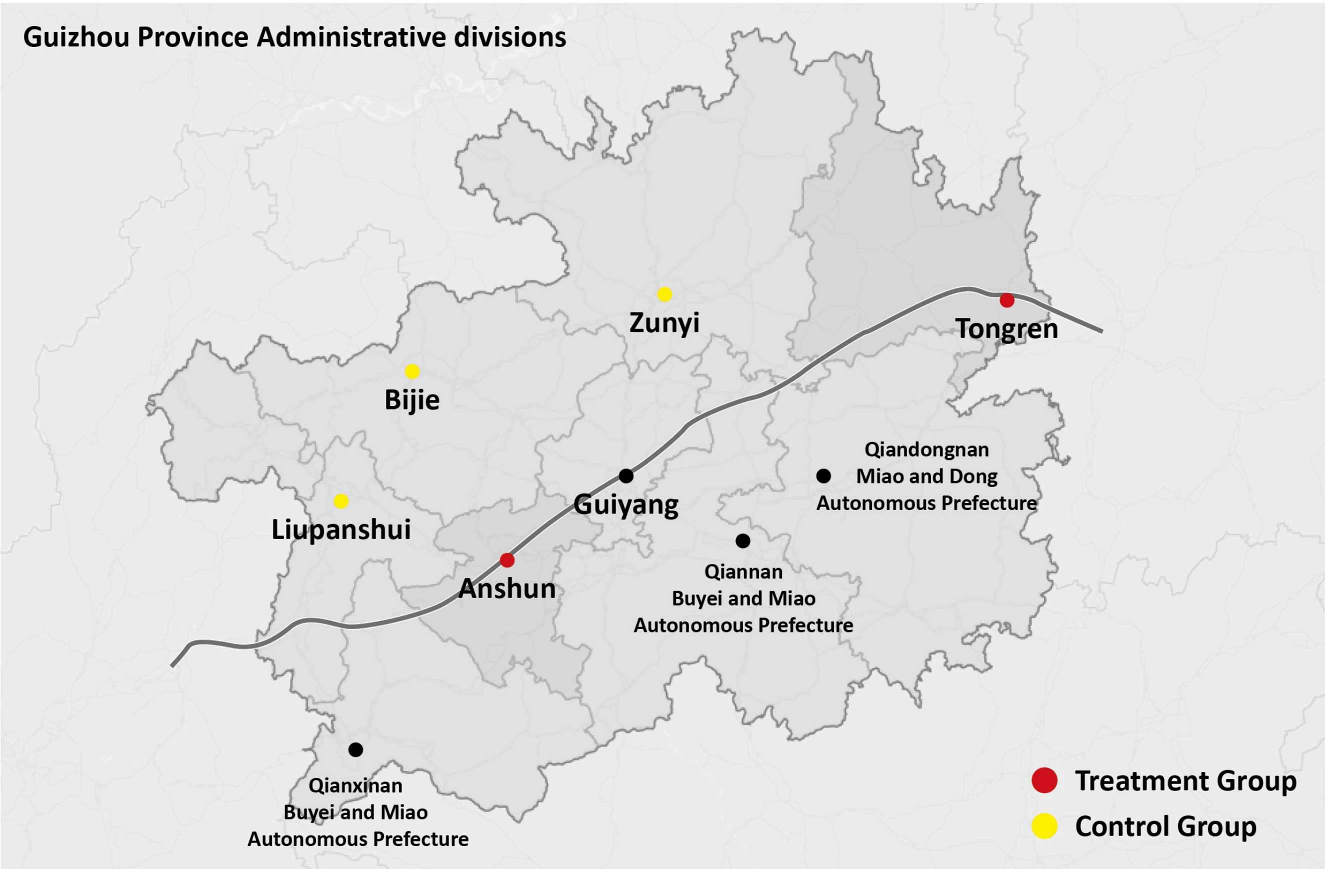
In [4]:
```
display(Image(url='https://raw.githubusercontent.com/irrationalBI/CASA0006_Assessment/main/da
```



Basemap source: https://datav.aliyun.com/portal/school/atlas/area_selector

2. Among the cities in Guizhou Province, Guiyang is a sub-provincial city, and a HSR has been connected to Guangzhou from Guiyang since 2014. The three autonomous prefectures in the south are also different from ordinary prefecture-level cities.

In [5]:
```
display(Image(url='https://raw.githubusercontent.com/irrationalBI/CASA0006_Assessment/main/da
```



Basemap source: https://datav.aliyun.com/portal/school/atlas/area_selector

Thus, we obtained the list of cities for the two DID analysis.

In [6]:
```python
cities_shandong_treatment = ['Dezhou', 'Taian', 'Zaozhuang']
cities_shandong_control = ['Liaocheng', 'Heze', 'Binzhou', 'Zibo', 'Linyi']
cities_guizhou_treatment = ['Anshun', 'Tongren']
cities_guizhou_control = ['Zunyi', 'Bijie', 'Liupanshui']

print('Shandong Province treatment group:', cities_shandong_treatment)
print('Shandong Province control group:', cities_shandong_control)
print('Guizhou Province treatment group:', cities_guizhou_treatment)
print('Guizhou Province control group:', cities_guizhou_control)
```

```
Shandong Province treatment group: ['Dezhou', 'Taian', 'Zaozhuang']
Shandong Province control group: ['Liaocheng', 'Heze', 'Binzhou', 'Zibo', 'Linyi']
Guizhou Province treatment group: ['Anshun', 'Tongren']
Guizhou Province control group: ['Zunyi', 'Bijie', 'Liupanshui']
```

Then, we select the time span. When dividing the impact, the impact of high-speed rail lines opened on or after July 1 is generally counted as the following year, while those opened before July 1 are counted as the current year (Chi and Han, 2023).

The Pre-Treatment Period is selected to be longer for better evaluation of trends. The cutoff time of the Post-Treatment Period depends on important events. For the Beijing-Shanghai HSR, other HSR lines opened in Shandong Province in 2016, which will bring new impacts. For the Changsha-Kunming HSR, the dataset is up to 2021. In 2020-2021, the COVID-19 may have a complex impact on GDP data, and we will further consider whether to adjust the cutoff point to 2019 in our analysis.

In [7]:
```python
years = list(range(2001, 2022))
years_shandong_pre = list(range(2001, 2011))
years_shandong_post = list(range(2011, 2016))
years_guizhou_pre = list(range(2007, 2017))
years_guizhou_post = list(range(2017, 2022))

print('Shandong Province Pre-Treatment Period:', years_shandong_pre)
print('Shandong Province Post-Treatment Period:', years_shandong_post)
print('Guizhou Province Pre-Treatment Period:', years_guizhou_pre)
print('Guizhou Province Post-Treatment Period:', years_guizhou_post)
```

```
Shandong Province Pre-Treatment Period: [2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010]
Shandong Province Post-Treatment Period: [2011, 2012, 2013, 2014, 2015]
Guizhou Province Pre-Treatment Period: [2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016]
Guizhou Province Post-Treatment Period: [2017, 2018, 2019, 2020, 2021]
```

Finally, we need to select the indicator columns to be used for measuring impact or for PSM analysis. GDP, GDP by industry, GDP per capita, and real estate-related data are usually used to measure impact (Liang et al., 2020). Some of these indicators require secondary calculations.

In [8]:
```python
# Filter the columns of the original data
indicator_columns = ['GDP', 'primary_GDP', 'secondary_GDP', 'tertiary_GDP', 'real_estate',
                     'population', 'population_re']

indicator = {
    # GDP data (for measuring impact)
    'GDP': 'Gross Regional Product (10k CNY)',
    'primary_GDP': 'Primary Industry Value Added (10k CNY)',
    'secondary_GDP': 'Secondary Industry Value Added (10k CNY)',
```

```python
    'tertiary_GDP': 'Tertiary Industry Value Added (10k CNY)',

    # Real estate investment (for measuring impact)
    'real_estate': 'Real Estate Development Investment (10k CNY)',

    # Population (for calculating per capita indicators)
    'population': 'Average annual population (10,000)',
    'population_re': 'Registered population (10,000)'
}

df_indicator = pd.DataFrame(list(indicator.items()), columns=['Indicator', 'Description'])
df_indicator
```

Out[8]:

	Indicator	Description
0	GDP	Gross Regional Product (10k CNY)
1	primary_GDP	Primary Industry Value Added (10k CNY)
2	secondary_GDP	Secondary Industry Value Added (10k CNY)
3	tertiary_GDP	Tertiary Industry Value Added (10k CNY)
4	real_estate	Real Estate Development Investment (10k CNY)
5	population	Average annual population (10,000)
6	population_re	Registered population (10,000)

4.2.2 Data filtering code

The dataset can be imported and filtered using the following code.

However, since all the city names and column names in the dataset are in Chinese, the code for the filtering process contains a large number of Chinese characters. We directly provide the filtered dataset in 4.2.3.

4.2.3 Data filtering result

In [14]:

```python
# Import filtered data
filtered_data = pd.read_csv('https://raw.githubusercontent.com/irrationalBI/CASA0006_Assessme
filtered_data
```

Out[14]:

	year	city	GDP	primary_GDP	secondary_GDP	tertiary_GDP	real_estate	populat
0	2021	Zibo	42010000	1810631.000	2.073194e+07	1.947164e+07	3742557	4:
1	2020	Zibo	36730000	1568371.000	1.776997e+07	1.739166e+07	3441747	4:
2	2019	Zibo	36420000	1489578.000	1.818086e+07	1.674956e+07	3133055	4:
3	2018	Zibo	50683500	1459684.800	2.640104e+07	2.282785e+07	2694898	4:
4	2017	Zibo	47713600	1484296.110	2.477763e+07	2.114241e+07	2418498	4:
...	...	...	...	...	...	...	...	
268	2005	Tongren	2073252	601142.214	5.802114e+05	8.918959e+05	365059	3(
269	2004	Tongren	1822602	528854.093	5.103967e+05	7.833478e+05	324747	2:
270	2003	Tongren	1571952	456565.972	4.405820e+05	6.747997e+05	284435	2(
271	2002	Tongren	1321301	384277.851	3.707674e+05	5.662516e+05	244123	2·
272	2001	Tongren	1070651	311989.729	3.009527e+05	4.577034e+05	203811	2:

273 rows × 9 columns

4.3 Data processing

First, calculate some new indicators.

In [15]:

```
new_indicator_columns = ['GDP_pc', 'secondary_GDP_pc', 'tertiary_GDP_pc']

new_indicator = {
    # GDP data (for measuring impact)
    'GDP_pc': 'GDP per capita',
    'secondary_GDP_pc': 'Per capita GDP of secondary industry',
    'tertiary_GDP_pc': 'Per capita GDP of tertiary industry',
}
```

In [16]:

```
# GDP per capita
filtered_data['GDP_pc'] = filtered_data['GDP'] / filtered_data['population']

# Per capita GDP of secondary industry
filtered_data['secondary_GDP_pc'] = filtered_data['secondary_GDP'] / filtered_data['populatio

# Per capita GDP of tertiary industry
filtered_data['tertiary_GDP_pc'] = filtered_data['tertiary_GDP'] / filtered_data['population'

filtered_data
```

Out[16]:

	year	city	GDP	primary_GDP	secondary_GDP	tertiary_GDP	real_estate	populat
0	2021	Zibo	42010000	1810631.000	2.073194e+07	1.947164e+07	3742557	4
1	2020	Zibo	36730000	1568371.000	1.776997e+07	1.739166e+07	3441747	4
2	2019	Zibo	36420000	1489578.000	1.818086e+07	1.674956e+07	3133055	4
3	2018	Zibo	50683500	1459684.800	2.640104e+07	2.282785e+07	2694898	4
4	2017	Zibo	47713600	1484296.110	2.477763e+07	2.114241e+07	2418498	4
...	...	...	...	...	...	...	...	
268	2005	Tongren	2073252	601142.214	5.802114e+05	8.918959e+05	365059	3
269	2004	Tongren	1822602	528854.093	5.103967e+05	7.833478e+05	324747	2
270	2003	Tongren	1571952	456565.972	4.405820e+05	6.747997e+05	284435	2
271	2002	Tongren	1321301	384277.851	3.707674e+05	5.662516e+05	244123	2
272	2001	Tongren	1070651	311989.729	3.009527e+05	4.577034e+05	203811	2

273 rows × 12 columns

In [17]:

```
# Add new indicators to existing indicators
indicator_columns = indicator_columns + new_indicator_columns

indicator.update(new_indicator)

df_indicator = pd.DataFrame(list(indicator.items()), columns=['Indicator', 'Description'])
df_indicator
```

Out[17]:

	Indicator	Description
0	GDP	Gross Regional Product (10k CNY)
1	primary_GDP	Primary Industry Value Added (10k CNY)
2	secondary_GDP	Secondary Industry Value Added (10k CNY)
3	tertiary_GDP	Tertiary Industry Value Added (10k CNY)
4	real_estate	Real Estate Development Investment (10k CNY)
5	population	Average annual population (10,000)
6	population_re	Registered population (10,000)
7	GDP_pc	GDP per capita
8	secondary_GDP_pc	Per capita GDP of secondary industry
9	tertiary_GDP_pc	Per capita GDP of tertiary industry

Now that we have all the indicators, we need to check the indicators to avoid data errors and take logarithms if necessary.

In [18]:

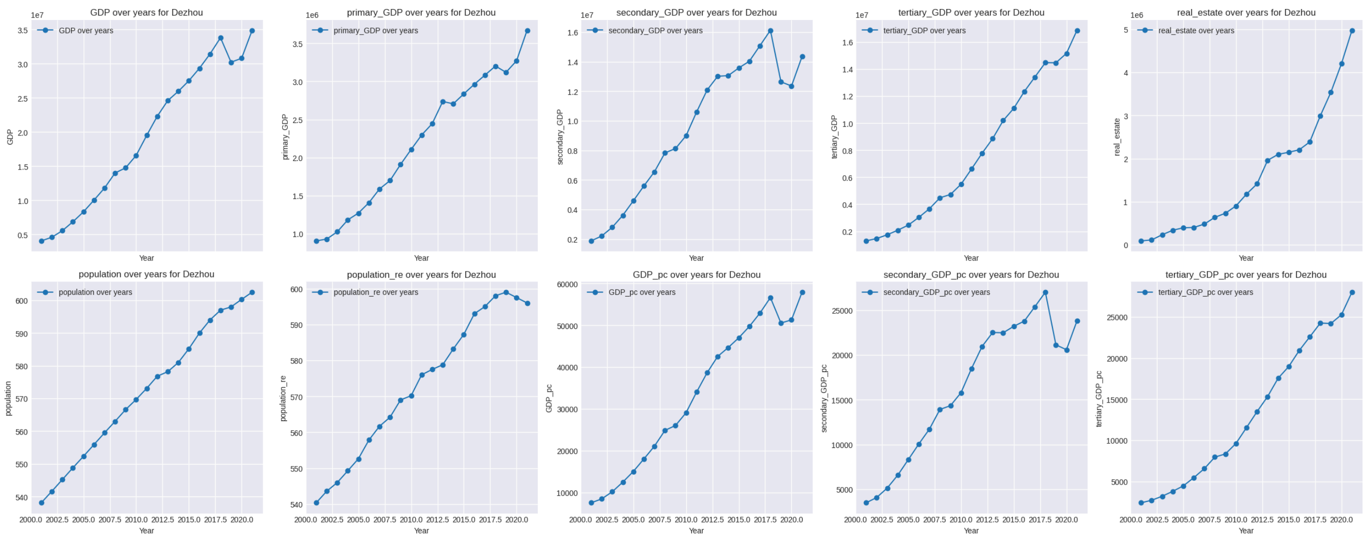
```
# Define a function to check the indicator trend
def plot_indicators(data, city_name, indicators, nrows, ncols):

    # Filter data for selected cities
    city_data = data[data['city'] == city_name]

    plt.style.use('seaborn-darkgrid')
```

```
    # Creating graphs and subplots by layout
    fig, axes = plt.subplots(nrows=nrows, ncols=ncols, figsize=(5 * ncols, 5 * nrows), sharex
    axes = axes.flatten()  # Flatten axes array for easy iteration

    # Traverse all indicators and draw a line chart for each
    for ax, ind in zip(axes, indicators):
        ax.plot(city_data['year'], city_data[ind], marker='o', label=f"{ind} over years")
        ax.set_title(f"{ind} over years for {city_name}")
        ax.set_xlabel('Year')
        ax.set_ylabel(ind)
        ax.legend(loc='best')

    # Hide extra subplots if there are more plots than indicators
    for i in range(len(indicators), len(axes)):
        axes[i].axis('off')

    plt.tight_layout()
    plt.show()
```

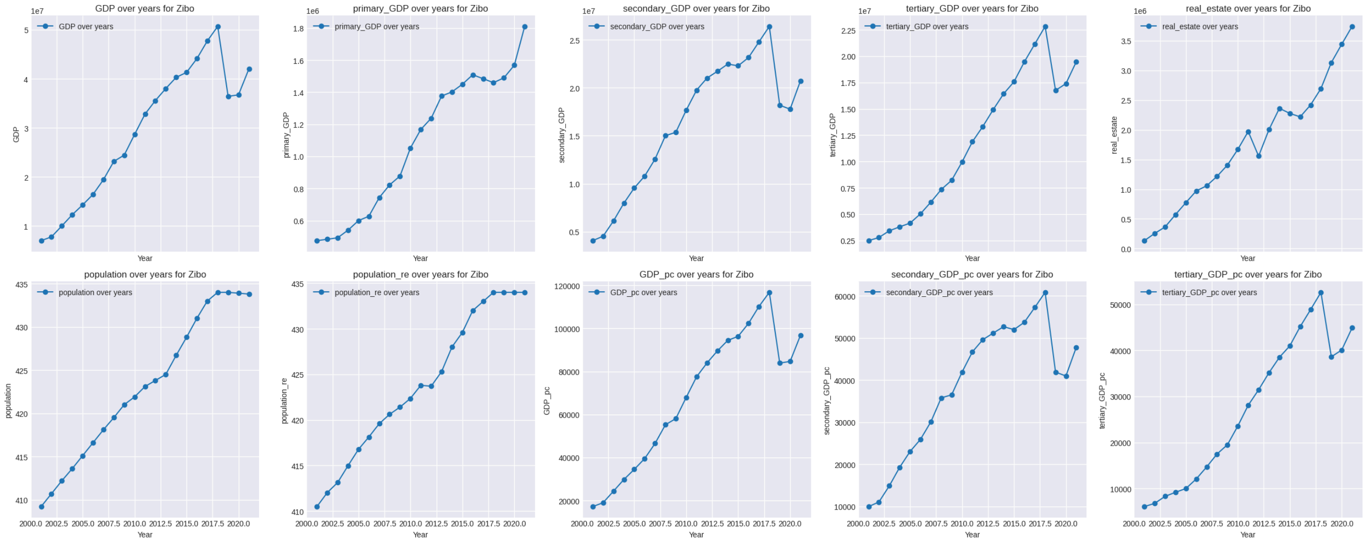
In [19]:
```
indicator_columns
```

Out[19]:
```
['GDP',
 'primary_GDP',
 'secondary_GDP',
 'tertiary_GDP',
 'real_estate',
 'population',
 'population_re',
 'GDP_pc',
 'secondary_GDP_pc',
 'tertiary_GDP_pc']
```

In [20]:
```
# Select a city from the treatment group in Shandong Province (DID group 1)
plot_indicators(filtered_data, 'Dezhou', indicator_columns, 2, 5)
```

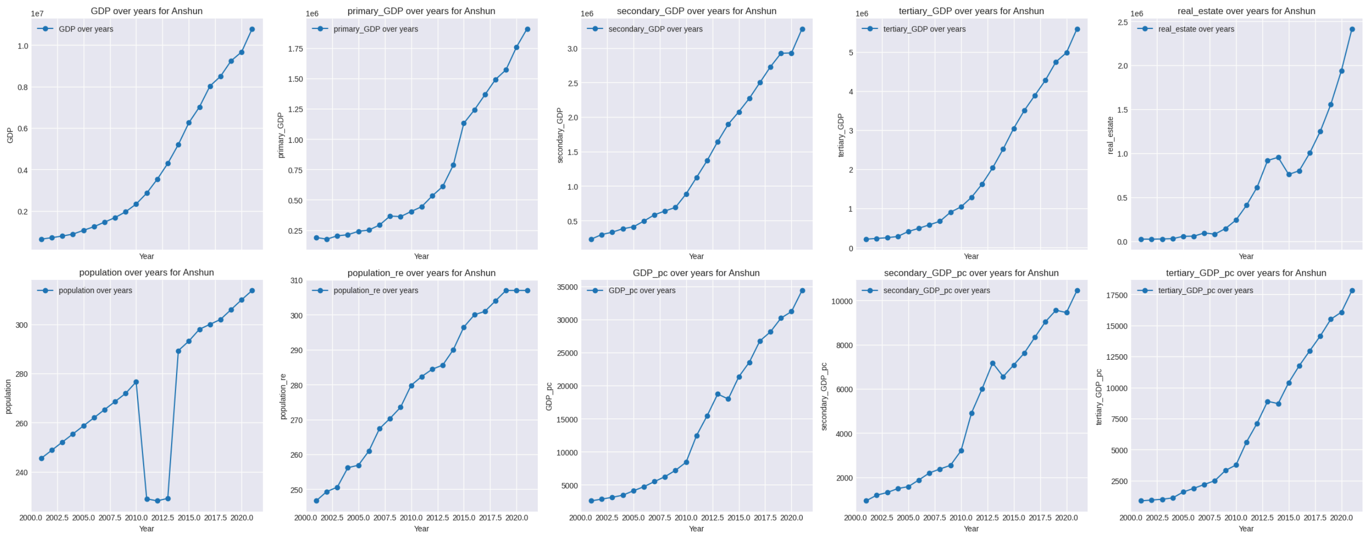


In [21]:
```
# Select a city from the control group in Shandong Province (DID group 1)
plot_indicators(filtered_data, 'Zibo', indicator_columns, 2, 5)
```

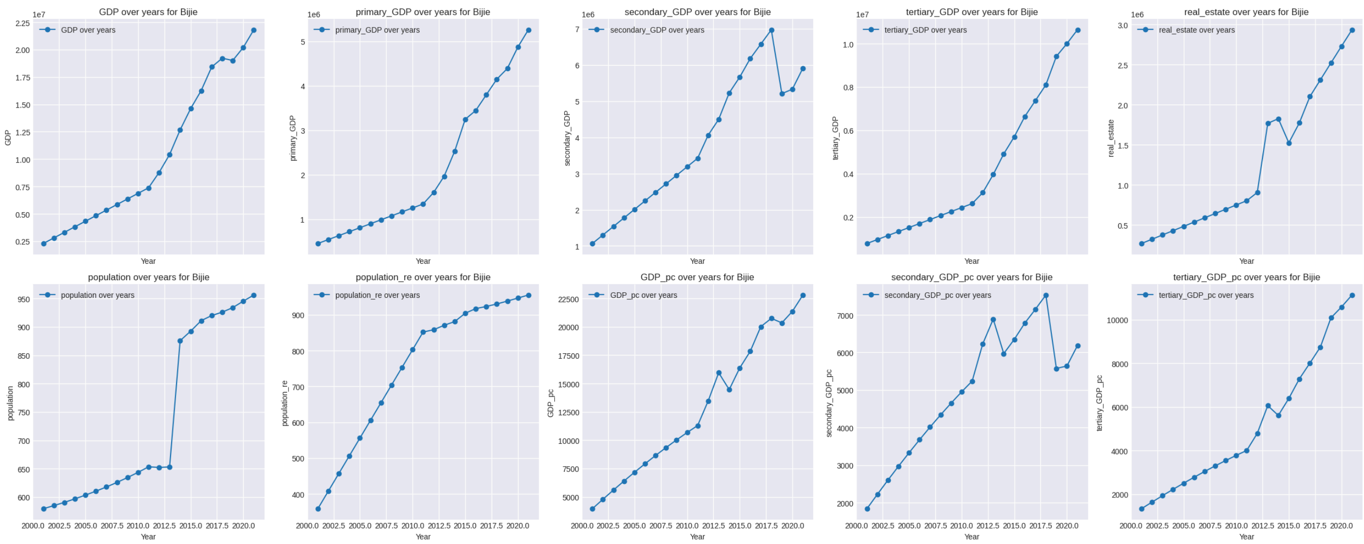


There's a significant problem that arises in the post-2019 portion of the GDP-related data. According to government policy, Shandong Province launched a GDP "squeeze" in 2019. To solve the problem of GDP over-reporting, the calculation standard was changed. However, our cut-off time for Shandong Province is 2015, so it will not be affected.

In [22]:
```
# Select a city from the treatment group in Guizhou Province (DID group 2)
plot_indicators(filtered_data, 'Anshun', indicator_columns, 2, 5)
```



In [23]:
```
# Select a city from the control group in Guizhou Province (DID group 2)
plot_indicators(filtered_data, 'Bijie', indicator_columns, 2, 5)
```



In the process of checking the data of all cities, we found that the data of individual cities in individual years were abnormal. We manually queried government information to correct these data. After checking, these data are all real data, so we will not change them. Due to the anomalies in population data, we will not use per capita data.

Regarding the impact of the COVID-19, Guizhou Province's GDP still had a positive growth rate in 2020 (4.5%) and 2021 (8.1%). Judging from the plot, there was no drastic change, so we retained the data for these two years.

Then, we can select the indicators to measure the impact in DID analysis, and take the logarithm of the necessary indicators. Due to data quality, we will not use all the filtered data.

In [24]:
```
# Take the logarithm of the GDP data we are going to use
indicator_log = ['GDP', 'secondary_GDP', 'tertiary_GDP', 'real_estate']

# Take the logarithm of the specified column and add a new column
for column in indicator_log:
    filtered_data[f'{column}_log'] = np.log(filtered_data[column])

filtered_data
```

Out[24]:

	year	city	GDP	primary_GDP	secondary_GDP	tertiary_GDP	real_estate	populat
0	2021	Zibo	42010000	1810631.000	2.073194e+07	1.947164e+07	3742557	4
1	2020	Zibo	36730000	1568371.000	1.776997e+07	1.739166e+07	3441747	4
2	2019	Zibo	36420000	1489578.000	1.818086e+07	1.674956e+07	3133055	4
3	2018	Zibo	50683500	1459684.800	2.640104e+07	2.282785e+07	2694898	4
4	2017	Zibo	47713600	1484296.110	2.477763e+07	2.114241e+07	2418498	4
...	...	...	...	...	...	...	...	
268	2005	Tongren	2073252	601142.214	5.802114e+05	8.918959e+05	365059	3
269	2004	Tongren	1822602	528854.093	5.103967e+05	7.833478e+05	324747	2
270	2003	Tongren	1571952	456565.972	4.405820e+05	6.747997e+05	284435	2
271	2002	Tongren	1321301	384277.851	3.707674e+05	5.662516e+05	244123	2
272	2001	Tongren	1070651	311989.729	3.009527e+05	4.577034e+05	203811	2

273 rows × 16 columns

In [25]:
```
# Indicators for DID analysis
indicator_DID = ['GDP_log', 'secondary_GDP_log', 'tertiary_GDP_log', 'real_estate_log']
indicator_DID
```

Out[25]:
```
['GDP_log', 'secondary_GDP_log', 'tertiary_GDP_log', 'real_estate_log']
```

Finally, we can split the table into two groups for DID analysis, adjust the year range, and add the HSR opening variable.

In [26]:
```
# Extract the required columns
columns_to_extract = ['year', 'city'] + indicator_DID
filtered_data_final = filtered_data[columns_to_extract]
```

In [27]:
```
# Split the data of the first group of DID analysis (Shandong province)
cities_shandong = cities_shandong_treatment + cities_shandong_control
data_shandong = filtered_data_final[filtered_data_final['city'].isin(cities_shandong)]

# Keep the years required for analysis
```

```python
data_shandong = data_shandong[data_shandong['year'].isin(list(range(2001, 2016)))]
data_shandong.reset_index(drop=True, inplace=True)

# Add the HSR opening variable
data_shandong['hsr'] = 0
mask = (data_shandong['year'].isin(years_shandong_post) &
        data_shandong['city'].isin(cities_shandong_treatment))
data_shandong.loc[mask, 'hsr'] = 1

data_shandong
```

Out[27]:

	year	city	GDP_log	secondary_GDP_log	tertiary_GDP_log	real_estate_log	hsr
0	2015	Zibo	17.536431	16.919504	16.681471	14.638315	0
1	2014	Zibo	17.511804	16.927870	16.614090	14.675013	0
2	2013	Zibo	17.453423	16.893407	16.518204	14.510529	0
3	2012	Zibo	17.387072	16.860625	16.404909	14.258039	0
4	2011	Zibo	17.306009	16.798844	16.290450	14.494457	0
...	...	...	...	...	...	...	...
115	2005	Heze	15.321475	14.477505	13.838670	12.242419	0
116	2004	Heze	15.110840	14.200532	13.606663	11.813297	0
117	2003	Heze	14.880566	13.848584	13.478142	11.324473	0
118	2002	Heze	14.731161	13.570248	13.322621	10.508159	0
119	2001	Heze	14.629636	13.351657	13.216997	10.055436	0

120 rows × 7 columns

In [28]:

```python
# Split the data of the second group of DID analysis (Guizhou province)
cities_guizhou = cities_guizhou_treatment + cities_guizhou_control
data_guizhou = filtered_data_final[filtered_data_final['city'].isin(cities_guizhou)]

# Keep the years required for analysis
data_guizhou = data_guizhou[data_guizhou['year'].isin(list(range(2006, 2022)))]
data_guizhou.reset_index(drop=True, inplace=True)

# Add the HSR opening variable
data_guizhou['hsr'] = 0
mask = (data_guizhou['year'].isin(years_guizhou_post) &
        data_guizhou['city'].isin(cities_guizhou_treatment))
data_guizhou.loc[mask, 'hsr'] = 1

data_guizhou
```

Out[28]:

	year	city	GDP_log	secondary_GDP_log	tertiary_GDP_log	real_estate_log	hsr
0	2021	Liupanshui	16.506075	15.717518	15.646401	13.657505	0
1	2020	Liupanshui	16.410765	15.607357	15.554158	13.657664	0
2	2019	Liupanshui	16.353958	15.577647	15.480966	13.656293	0
3	2018	Liupanshui	16.540542	15.819613	15.665154	13.668088	0
4	2017	Liupanshui	16.497703	15.762810	15.576805	13.566571	0
...	...	...	...	...	...	...	...
75	2010	Tongren	15.017433	13.742170	14.176422	13.247442	0
76	2009	Tongren	14.939093	13.664071	14.097744	13.173640	0
77	2008	Tongren	14.854091	13.579352	14.012343	13.093954	0
78	2007	Tongren	14.761186	13.486785	13.918962	13.007363	0
79	2006	Tongren	14.658759	13.384768	13.815955	12.912558	0

80 rows × 7 columns

These are the final data used for analysis.

5 Methodology

5.1 Method Overview

The core idea is to use Fixed-effects-DID to analyze the impact of HSR opening in Shandong Province and Guizhou Province respectively, and then compare the two impacts. The process is as follows:

1. Conduct parallel trends test for Shandong Province data.
2. Based on the results, conduct Fixed-effects-DID analysis.
3. Apply the same procedure to Guizhou Province.
4. Compare the results of the two groups and draw conclusions.

The standard DID model is as follows:

In [46]:
```
display(Math(r'Y_{it} = \beta_0 + \beta_1 treat_{i} + \beta_2 period_{t} + \beta_3 treat_{i}
```

$\displaystyle Y_{it} = \beta_0 + \beta_1 treat_{i} + \beta_2 period_{t} + \beta_3 treat_{i} \times period_{t} + \varepsilon_{it}$

The Fixed-effects-DID model is DID combined with the fixed effect model to better handle panel data:

In [47]:
```
display(Math(r'Y_{it} = \beta_0 + \beta_1 treat_{i} \times period_{t} + \lambda_{i} + \nu_{t}
```

$\displaystyle Y_{it} = \beta_0 + \beta_1 treat_{i} \times period_{t} + \lambda_{i} + \nu_{t} + \varepsilon_{it}$

Here, "lambda_i" is the individual fixed effect, which replaces the original treatment grouping variable.

"nu_t" is the time fixed effect, which replaces the original treatment time variable.

5.2 Fixed-effects-DID analysis for Shandong

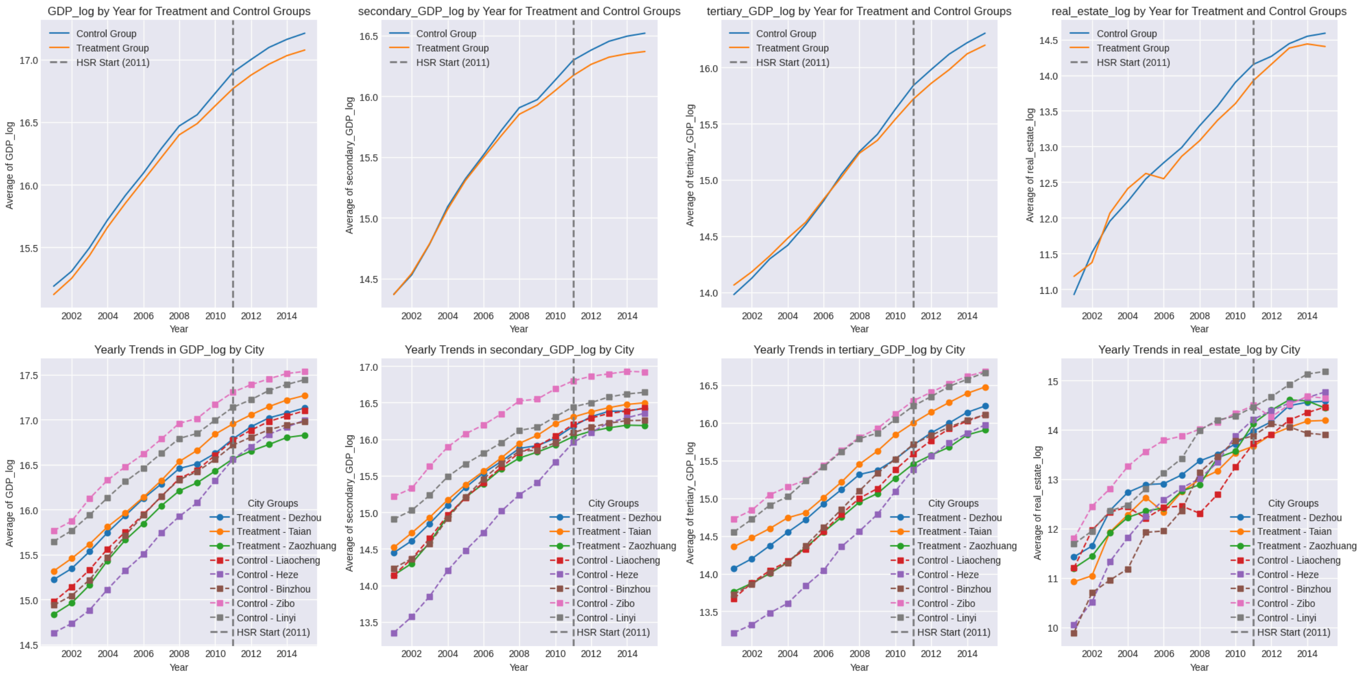
5.2.1 Parallel Trends Test

In [29]:
```
# Function for comparing the average trends of two sets of data
def plot_average_trends_subplot(data, column, treatment_cities, hsr_start_year, ax):

    # Distinguish between treatment and control groups
    data['treatment'] = data['city'].isin(treatment_cities).astype(int)

    # Calculate the mean value for each group in each year
    average_trends = data.groupby(['year', 'treatment'])[column].mean().unstack()

    ax.plot(average_trends.index, average_trends[0], label='Control Group')
    ax.plot(average_trends.index, average_trends[1], label='Treatment Group')
    ax.axvline(x=hsr_start_year, color='grey', linestyle='--', linewidth=2, label=f'HSR Start
    ax.set_title(f'{column} by Year for Treatment and Control Groups')
    ax.set_xlabel('Year')
    ax.set_ylabel(f'Average of {column}')
    ax.legend()
    ax.grid(True)
```

In [30]:
```
# Function for comparing data trends of all cities
def plot_city_trends_subplot(data, column, treatment_cities, control_cities, hsr_start_year, 

    grouped = data.groupby(['year', 'city'])[column].mean().reset_index()

    # Draw a trend line for each city
    for city in treatment_cities:
        city_data = grouped[grouped['city'] == city]
        ax.plot(city_data['year'], city_data[column], marker='o', label=f'Treatment - {city}'

    for city in control_cities:
        city_data = grouped[grouped['city'] == city]
        ax.plot(city_data['year'], city_data[column], marker='s', linestyle='--', label=f'Con

    # Add a vertical dashed line at the specified year
    ax.axvline(x=hsr_start_year, color='grey', linestyle='--', linewidth=2, label=f'HSR Start

    ax.set_title(f'Yearly Trends in {column} by City')
    ax.set_xlabel('Year')
    ax.set_ylabel(f'Average of {column}')
    ax.grid(True)
    ax.legend(title='City Groups', bbox_to_anchor=(1.1, 0), loc='lower right')
```

In [31]:
```
def plot_all(data, treatment_cities, control_cities, hsr_start_year):

    fig, axs = plt.subplots(2, 4, figsize=(20, 10))

    indicator_DID = ['GDP_log', 'secondary_GDP_log', 'tertiary_GDP_log', 'real_estate_log']

    # average trends
    for i, column in enumerate(indicator_DID):
        plot_average_trends_subplot(data_shandong, column, cities_shandong_treatment, 2011, a

    # trends of all cities
    for i, column in enumerate(indicator_DID):
        plot_city_trends_subplot(data_shandong, column, cities_shandong_treatment, cities_sha

    plt.tight_layout()
    plt.show()
```

```
In [32]: plot_all(data_shandong, cities_shandong_treatment, cities_shandong_control, 2011)
```



5.2.2 DID analysis

```
In [33]: # Functions used to fit the model
def run_fixed_effects_model(data_pro, treatment_cities, post_hsr_years, response_variable, hs

    data = data_pro.copy()

    # Create treatment and control group markers
    data['treatment'] = data['city'].isin(treatment_cities).astype(int)

    # Create time period markers before and after the high-speed rail opening
    data['post'] = data['year'].isin(post_hsr_years).astype(int)

    # Creating interaction terms
    data['treatment_post'] = data['treatment'] * data['post']

    # Convert the city column into a categorical variable
    data['city'] = data['city'].astype('category')

    # Year as a continuous variable
    data['t'] = data['year'].astype(int) - hsr_start_year

    # Construct a fixed effect model formula and add fixed effects of city and year
    model_formula = f'{response_variable} ~ treatment_post + C(city) + C(t)'

    # Fitting the model
    model = smf.ols(model_formula, data=data).fit()

    # Output model results
    print(model.summary())

    return model
```

```
In [34]: # Functions for printing and storing results
def model_results(fixed_effects_model, data_pro, treatment_cities, post_hsr_years, hsr_start_

    # Indicators to use for analysis
    indicator_DID = ['GDP_log', 'secondary_GDP_log', 'tertiary_GDP_log', 'real_estate_log']

    # Initialize an empty DataFrame to store the results
    results_df = pd.DataFrame(columns=['Response Variable', 'Coefficient', 'Std Error',
```

```
                                          't-value', 'P-value', 'Conf. Int. Low', 'Conf. Int. Hi

    # Fit the model
    for response_variable in indicator_DID:
        print('///////////////////////////////////////////////////////////////////////////////
        print(f'{response_variable}')
        print('///////////////////////////////////////////////////////////////////////////////
        model = fixed_effects_model(data_pro, treatment_cities, post_hsr_years, response_vari
        print(' ')
        print(' ')
        print(' ')

        # Extracting model results
        conf_int = model.conf_int()

        # Add the desired results to the DataFrame
        temp_df = pd.DataFrame({
            'Response Variable': [response_variable],
            'Coefficient': [model.params['treatment_post']],
            'Std Error': [model.bse['treatment_post']],
            't-value': [model.tvalues['treatment_post']],
            'P-value': [model.pvalues['treatment_post']],
            'Conf. Int. Low': [conf_int.loc['treatment_post', 0]],
            'Conf. Int. High': [conf_int.loc['treatment_post', 1]]
        })

        results_df = pd.concat([results_df, temp_df], ignore_index=True)

    return results_df
```

In [35]:
```
# Get results for Shandong Province
results_shandong = model_results(run_fixed_effects_model, data_shandong, cities_shandong_trea
```

```
////////////////////////////////////////////////////////////////////////////////////////
GDP_log
////////////////////////////////////////////////////////////////////////////////////////
                            OLS Regression Results
==============================================================================
Dep. Variable:                GDP_log   R-squared:                       0.990
Model:                            OLS   Adj. R-squared:                  0.987
Method:                 Least Squares   F-statistic:                     421.1
Date:                Mon, 29 Apr 2024   Prob (F-statistic):           2.89e-86
Time:                        01:35:20   Log-Likelihood:                 142.46
No. Observations:                 120   AIC:                            -238.9
Df Residuals:                      97   BIC:                            -174.8
Df Model:                          22
Covariance Type:            nonrobust
========================================================================================
                           coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------------
Intercept               14.9875      0.035    423.441      0.000      14.917      15.058
C(city)[T.Dezhou]        0.1945      0.032      6.092      0.000       0.131       0.258
C(city)[T.Heze]         -0.2564      0.030     -8.554      0.000      -0.316      -0.197
C(city)[T.Liaocheng]     0.0606      0.030      2.022      0.046       0.001       0.120
C(city)[T.Linyi]         0.5284      0.030     17.626      0.000       0.469       0.588
C(city)[T.Taian]         0.3000      0.032      9.398      0.000       0.237       0.363
C(city)[T.Zaozhuang]    -0.0899      0.032     -2.818      0.006      -0.153      -0.027
C(city)[T.Zibo]          0.6804      0.030     22.694      0.000       0.621       0.740
C(t)[T.-9]               0.1230      0.041      2.996      0.003       0.042       0.204
C(t)[T.-8]               0.3092      0.041      7.532      0.000       0.228       0.391
C(t)[T.-7]               0.5316      0.041     12.950      0.000       0.450       0.613
C(t)[T.-6]               0.7266      0.041     17.699      0.000       0.645       0.808
C(t)[T.-5]               0.9050      0.041     22.045      0.000       0.824       0.986
C(t)[T.-4]               1.0963      0.041     26.703      0.000       1.015       1.178
C(t)[T.-3]               1.2776      0.041     31.121      0.000       1.196       1.359
C(t)[T.-2]               1.3672      0.041     33.303      0.000       1.286       1.449
C(t)[T.-1]               1.5271      0.041     37.197      0.000       1.446       1.609
C(t)[T.0]                1.7078      0.043     39.846      0.000       1.623       1.793
C(t)[T.1]                1.8110      0.043     42.253      0.000       1.726       1.896
C(t)[T.2]                1.9042      0.043     44.428      0.000       1.819       1.989
C(t)[T.3]                1.9702      0.043     45.968      0.000       1.885       2.055
C(t)[T.4]                2.0177      0.043     47.075      0.000       1.933       2.103
treatment_post          -0.0627      0.033     -1.909      0.059      -0.128       0.002
==============================================================================
Omnibus:                        7.503   Durbin-Watson:                   0.339
Prob(Omnibus):                  0.023   Jarque-Bera (JB):                7.955
Skew:                           0.434   Prob(JB):                       0.0187
Kurtosis:                       3.915   Cond. No.                         17.1
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.


////////////////////////////////////////////////////////////////////////////////////////
secondary_GDP_log
////////////////////////////////////////////////////////////////////////////////////////
                            OLS Regression Results
==============================================================================
Dep. Variable:      secondary_GDP_log   R-squared:                       0.971
Model:                            OLS   Adj. R-squared:                  0.965
Method:                 Least Squares   F-statistic:                     149.0
Date:                Mon, 29 Apr 2024   Prob (F-statistic):           7.34e-65
Time:                        01:35:20   Log-Likelihood:                 72.745
No. Observations:                 120   AIC:                            -99.49
Df Residuals:                      97   BIC:                            -35.38
Df Model:                          22
Covariance Type:            nonrobust
```

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	14.2148	0.063	224.640	0.000	14.089	14.340
C(city)[T.Dezhou]	0.1698	0.057	2.976	0.004	0.057	0.283
C(city)[T.Heze]	-0.4398	0.054	-8.205	0.000	-0.546	-0.333
C(city)[T.Liaocheng]	0.0474	0.054	0.885	0.378	-0.059	0.154
C(city)[T.Linyi]	0.4268	0.054	7.963	0.000	0.320	0.533
C(city)[T.Taian]	0.2539	0.057	4.449	0.000	0.141	0.367
C(city)[T.Zaozhuang]	-0.0121	0.057	-0.212	0.833	-0.125	0.101
C(city)[T.Zibo]	0.7870	0.054	14.683	0.000	0.681	0.893
C(t)[T.-9]	0.1652	0.073	2.250	0.027	0.020	0.311
C(t)[T.-8]	0.4156	0.073	5.662	0.000	0.270	0.561
C(t)[T.-7]	0.7136	0.073	9.723	0.000	0.568	0.859
C(t)[T.-6]	0.9479	0.073	12.915	0.000	0.802	1.094
C(t)[T.-5]	1.1399	0.073	15.532	0.000	0.994	1.286
C(t)[T.-4]	1.3351	0.073	18.190	0.000	1.189	1.481
C(t)[T.-3]	1.5175	0.073	20.676	0.000	1.372	1.663
C(t)[T.-2]	1.5871	0.073	21.624	0.000	1.441	1.733
C(t)[T.-1]	1.7325	0.073	23.606	0.000	1.587	1.878
C(t)[T.0]	1.9212	0.077	25.072	0.000	1.769	2.073
C(t)[T.1]	2.0077	0.077	26.201	0.000	1.856	2.160
C(t)[T.2]	2.0755	0.077	27.086	0.000	1.923	2.228
C(t)[T.3]	2.1121	0.077	27.564	0.000	1.960	2.264
C(t)[T.4]	2.1344	0.077	27.855	0.000	1.982	2.287
treatment_post	-0.1068	0.059	-1.819	0.072	-0.223	0.010

Omnibus:	12.755	Durbin-Watson:	0.254
Prob(Omnibus):	0.002	Jarque-Bera (JB):	32.356
Skew:	0.260	Prob(JB):	9.42e-08
Kurtosis:	5.490	Cond. No.	17.1

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

//
tertiary_GDP_log
//

OLS Regression Results

Dep. Variable:	tertiary_GDP_log	R-squared:	0.988
Model:	OLS	Adj. R-squared:	0.985
Method:	Least Squares	F-statistic:	352.6
Date:	Mon, 29 Apr 2024	Prob (F-statistic):	1.42e-82
Time:	01:35:20	Log-Likelihood:	113.43
No. Observations:	120	AIC:	-180.9
Df Residuals:	97	BIC:	-116.8
Df Model:	22		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	13.8153	0.045	306.446	0.000	13.726	13.905
C(city)[T.Dezhou]	0.2357	0.041	5.798	0.000	0.155	0.316
C(city)[T.Heze]	-0.4232	0.038	-11.083	0.000	-0.499	-0.347
C(city)[T.Liaocheng]	-0.0551	0.038	-1.443	0.152	-0.131	0.021
C(city)[T.Linyi]	0.6864	0.038	17.975	0.000	0.611	0.762
C(city)[T.Taian]	0.4571	0.041	11.243	0.000	0.376	0.538
C(city)[T.Zaozhuang]	-0.0918	0.041	-2.259	0.026	-0.173	-0.011
C(city)[T.Zibo]	0.7506	0.038	19.655	0.000	0.675	0.826
C(t)[T.-9]	0.1357	0.052	2.595	0.011	0.032	0.239
C(t)[T.-8]	0.2962	0.052	5.664	0.000	0.192	0.400
C(t)[T.-7]	0.4304	0.052	8.232	0.000	0.327	0.534

```
C(t)[T.-6]              0.5986      0.052     11.447      0.000       0.495       0.702
C(t)[T.-5]              0.8079      0.052     15.450      0.000       0.704       0.912
C(t)[T.-4]              1.0307      0.052     19.712      0.000       0.927       1.134
C(t)[T.-3]              1.2363      0.052     23.644      0.000       1.133       1.340
C(t)[T.-2]              1.3751      0.052     26.298      0.000       1.271       1.479
C(t)[T.-1]              1.5854      0.052     30.320      0.000       1.482       1.689
C(t)[T.0]               1.8311      0.055     33.541      0.000       1.723       1.939
C(t)[T.1]               1.9732      0.055     36.145      0.000       1.865       2.082
C(t)[T.2]               2.1043      0.055     38.547      0.000       1.996       2.213
C(t)[T.3]               2.2206      0.055     40.676      0.000       2.112       2.329
C(t)[T.4]               2.3029      0.055     42.185      0.000       2.195       2.411
treatment_post         -0.1259      0.042     -3.011      0.003      -0.209      -0.043
==============================================================================
Omnibus:                        1.980   Durbin-Watson:                   0.349
Prob(Omnibus):                  0.372   Jarque-Bera (JB):                1.506
Skew:                           0.140   Prob(JB):                        0.471
Kurtosis:                       3.472   Cond. No.                         17.1
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.



/////////////////////////////////////////////////////////////////////////////////////
real_estate_log
/////////////////////////////////////////////////////////////////////////////////////
                            OLS Regression Results
==============================================================================
Dep. Variable:        real_estate_log   R-squared:                       0.941
Model:                            OLS   Adj. R-squared:                  0.928
Method:                 Least Squares   F-statistic:                     70.61
Date:                Mon, 29 Apr 2024   Prob (F-statistic):           6.39e-50
Time:                        01:35:20   Log-Likelihood:                -20.936
No. Observations:                 120   AIC:                             87.87
Df Residuals:                      97   BIC:                             152.0
Df Model:                          22
Covariance Type:            nonrobust
=========================================================================================
                            coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------------
Intercept                10.4769      0.138     75.846      0.000      10.203      10.751
C(city)[T.Dezhou]         0.7063      0.125      5.671      0.000       0.459       0.954
C(city)[T.Heze]           0.3276      0.117      2.800      0.006       0.095       0.560
C(city)[T.Liaocheng]      0.3113      0.117      2.661      0.009       0.079       0.544
C(city)[T.Linyi]          1.0279      0.117      8.785      0.000       0.796       1.260
C(city)[T.Taian]          0.3157      0.125      2.535      0.013       0.068       0.563
C(city)[T.Zaozhuang]      0.5025      0.125      4.034      0.000       0.255       0.750
C(city)[T.Zibo]           1.1592      0.117      9.908      0.000       0.927       1.391
C(t)[T.-9]                0.4400      0.160      2.746      0.007       0.122       0.758
C(t)[T.-8]                0.9747      0.160      6.084      0.000       0.657       1.293
C(t)[T.-7]                1.2779      0.160      7.976      0.000       0.960       1.596
C(t)[T.-6]                1.5541      0.160      9.700      0.000       1.236       1.872
C(t)[T.-5]                1.6683      0.160     10.413      0.000       1.350       1.986
C(t)[T.-4]                1.9182      0.160     11.973      0.000       1.600       2.236
C(t)[T.-3]                2.1911      0.160     13.676      0.000       1.873       2.509
C(t)[T.-2]                2.4718      0.160     15.428      0.000       2.154       2.790
C(t)[T.-1]                2.7682      0.160     17.278      0.000       2.450       3.086
C(t)[T.0]                 3.0773      0.167     18.397      0.000       2.745       3.409
C(t)[T.1]                 3.2325      0.167     19.325      0.000       2.900       3.564
C(t)[T.2]                 3.4297      0.167     20.504      0.000       3.098       3.762
C(t)[T.3]                 3.5153      0.167     21.015      0.000       3.183       3.847
C(t)[T.4]                 3.5288      0.167     21.097      0.000       3.197       3.861
treatment_post           -0.0822      0.128     -0.641      0.523      -0.337       0.172
==============================================================================
Omnibus:                        2.460   Durbin-Watson:                   0.511
```

```
Prob(Omnibus):                  0.292   Jarque-Bera (JB):                2.027
Skew:                          -0.309   Prob(JB):                        0.363
Kurtosis:                       3.155   Cond. No.                         17.1
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

It can be found from the results that R-Squared is very high. Considering that adding time fixed effects significantly increases the number of parameters, there may be a risk of overfitting.

To solve the problem, we adjust the model to treat time as a continuous variable and assume that the logarithm of GDP and other related indicators grows linearly over time. A simplified model is used for analysis.

In [36]:
```
# Functions used to fit the simplified model
def run_fixed_effects_model_simplified(data_pro, treatment_cities, post_hsr_years, response_v

    data = data_pro.copy()

    # Create treatment and control group markers
    data['treatment'] = data['city'].isin(treatment_cities).astype(int)

    # Create time period markers before and after the high-speed rail opening
    data['post'] = data['year'].isin(post_hsr_years).astype(int)

    # Creating interaction terms
    data['treatment_post'] = data['treatment'] * data['post']

    # Convert the city column into a categorical variable
    data['city'] = data['city'].astype('category')

    # Year as a continuous variable
    data['t'] = data['year'].astype(int) - hsr_start_year

    # Construct a fixed effect model formula and add fixed effects of city and year
    model_formula = f'{response_variable} ~ treatment_post + C(city) + t'

    # Fitting the model
    model = smf.ols(model_formula, data=data).fit()

    # Output model results
    print(model.summary())

    return model
```

In [37]:
```
# Get the results for Shandong Province for the simplified model
results_shandong_simplified = model_results(run_fixed_effects_model_simplified, data_shandong
```

```
////////////////////////////////////////////////////////////////////////////////////
GDP_log
////////////////////////////////////////////////////////////////////////////////////
                            OLS Regression Results
==============================================================================
Dep. Variable:                GDP_log   R-squared:                       0.974
Model:                            OLS   Adj. R-squared:                  0.972
Method:                 Least Squares   F-statistic:                     458.2
Date:                Mon, 29 Apr 2024   Prob (F-statistic):           7.17e-83
Time:                        01:35:20   Log-Likelihood:                 87.300
No. Observations:                 120   AIC:                            -154.6
Df Residuals:                     110   BIC:                            -126.7
Df Model:                           9
Covariance Type:            nonrobust
========================================================================================
                           coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------------
Intercept               16.6039      0.033    506.687      0.000      16.539      16.669
C(city)[T.Dezhou]        0.2165      0.047      4.606      0.000       0.123       0.310
C(city)[T.Heze]         -0.2564      0.045     -5.752      0.000      -0.345      -0.168
C(city)[T.Liaocheng]     0.0606      0.045      1.359      0.177      -0.028       0.149
C(city)[T.Linyi]         0.5284      0.045     11.853      0.000       0.440       0.617
C(city)[T.Taian]         0.3220      0.047      6.851      0.000       0.229       0.415
C(city)[T.Zaozhuang]    -0.0679      0.047     -1.445      0.151      -0.161       0.025
C(city)[T.Zibo]          0.6804      0.045     15.261      0.000       0.592       0.769
treatment_post          -0.1288      0.045     -2.886      0.005      -0.217      -0.040
t                        0.1549      0.003     51.967      0.000       0.149       0.161
==============================================================================
Omnibus:                        8.196   Durbin-Watson:                   0.405
Prob(Omnibus):                  0.017   Jarque-Bera (JB):                8.600
Skew:                          -0.654   Prob(JB):                       0.0136
Kurtosis:                       2.917   Cond. No.                         45.4
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.


////////////////////////////////////////////////////////////////////////////////////
secondary_GDP_log
////////////////////////////////////////////////////////////////////////////////////
                            OLS Regression Results
==============================================================================
Dep. Variable:      secondary_GDP_log   R-squared:                       0.934
Model:                            OLS   Adj. R-squared:                  0.929
Method:                 Least Squares   F-statistic:                     174.0
Date:                Mon, 29 Apr 2024   Prob (F-statistic):           8.50e-61
Time:                        01:35:20   Log-Likelihood:                 23.178
No. Observations:                 120   AIC:                            -26.36
Df Residuals:                     110   BIC:                             1.519
Df Model:                           9
Covariance Type:            nonrobust
========================================================================================
                           coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------------
Intercept               16.0244      0.056    286.581      0.000      15.914      16.135
C(city)[T.Dezhou]        0.2125      0.080      2.650      0.009       0.054       0.371
C(city)[T.Heze]         -0.4398      0.076     -5.781      0.000      -0.591      -0.289
C(city)[T.Liaocheng]     0.0474      0.076      0.624      0.534      -0.103       0.198
C(city)[T.Linyi]         0.4268      0.076      5.611      0.000       0.276       0.578
C(city)[T.Taian]         0.2966      0.080      3.698      0.000       0.138       0.456
C(city)[T.Zaozhuang]     0.0306      0.080      0.382      0.703      -0.128       0.190
C(city)[T.Zibo]          0.7870      0.076     10.345      0.000       0.636       0.938
treatment_post          -0.2350      0.076     -3.087      0.003      -0.386      -0.084
t                        0.1631      0.005     32.057      0.000       0.153       0.173
```

```
==============================================================================
Omnibus:                       12.034   Durbin-Watson:                   0.299
Prob(Omnibus):                  0.002   Jarque-Bera (JB):               12.833
Skew:                          -0.790   Prob(JB):                      0.00163
Kurtosis:                       3.264   Cond. No.                         45.4
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.



////////////////////////////////////////////////////////////////////////////////////////
tertiary_GDP_log
////////////////////////////////////////////////////////////////////////////////////////
                            OLS Regression Results
==============================================================================
Dep. Variable:       tertiary_GDP_log   R-squared:                       0.983
Model:                            OLS   Adj. R-squared:                  0.982
Method:                 Least Squares   F-statistic:                     711.7
Date:                Mon, 29 Apr 2024   Prob (F-statistic):           3.73e-93
Time:                        01:35:20   Log-Likelihood:                 94.676
No. Observations:                 120   AIC:                            -169.4
Df Residuals:                     110   BIC:                            -141.5
Df Model:                           9
Covariance Type:            nonrobust
=========================================================================================
                            coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------------
Intercept                15.5386      0.031    504.237      0.000      15.478      15.600
C(city)[T.Dezhou]         0.2301      0.044      5.207      0.000       0.143       0.318
C(city)[T.Heze]          -0.4232      0.042    -10.094      0.000      -0.506      -0.340
C(city)[T.Liaocheng]     -0.0551      0.042     -1.315      0.191      -0.138       0.028
C(city)[T.Linyi]          0.6864      0.042     16.372      0.000       0.603       0.769
C(city)[T.Taian]          0.4515      0.044     10.215      0.000       0.364       0.539
C(city)[T.Zaozhuang]     -0.0974      0.044     -2.204      0.030      -0.185      -0.010
C(city)[T.Zibo]           0.7506      0.042     17.902      0.000       0.667       0.834
treatment_post           -0.1092      0.042     -2.602      0.011      -0.192      -0.026
t                         0.1760      0.003     62.784      0.000       0.170       0.182
==============================================================================
Omnibus:                        2.272   Durbin-Watson:                   0.556
Prob(Omnibus):                  0.321   Jarque-Bera (JB):                1.742
Skew:                          -0.205   Prob(JB):                        0.418
Kurtosis:                       3.424   Cond. No.                         45.4
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.



////////////////////////////////////////////////////////////////////////////////////////
real_estate_log
////////////////////////////////////////////////////////////////////////////////////////
                            OLS Regression Results
==============================================================================
Dep. Variable:        real_estate_log   R-squared:                       0.920
Model:                            OLS   Adj. R-squared:                  0.913
Method:                 Least Squares   F-statistic:                     140.0
Date:                Mon, 29 Apr 2024   Prob (F-statistic):           5.32e-56
Time:                        01:35:20   Log-Likelihood:                -39.675
No. Observations:                 120   AIC:                             99.35
Df Residuals:                     110   BIC:                             127.2
Df Model:                           9
Covariance Type:            nonrobust
=========================================================================================
```

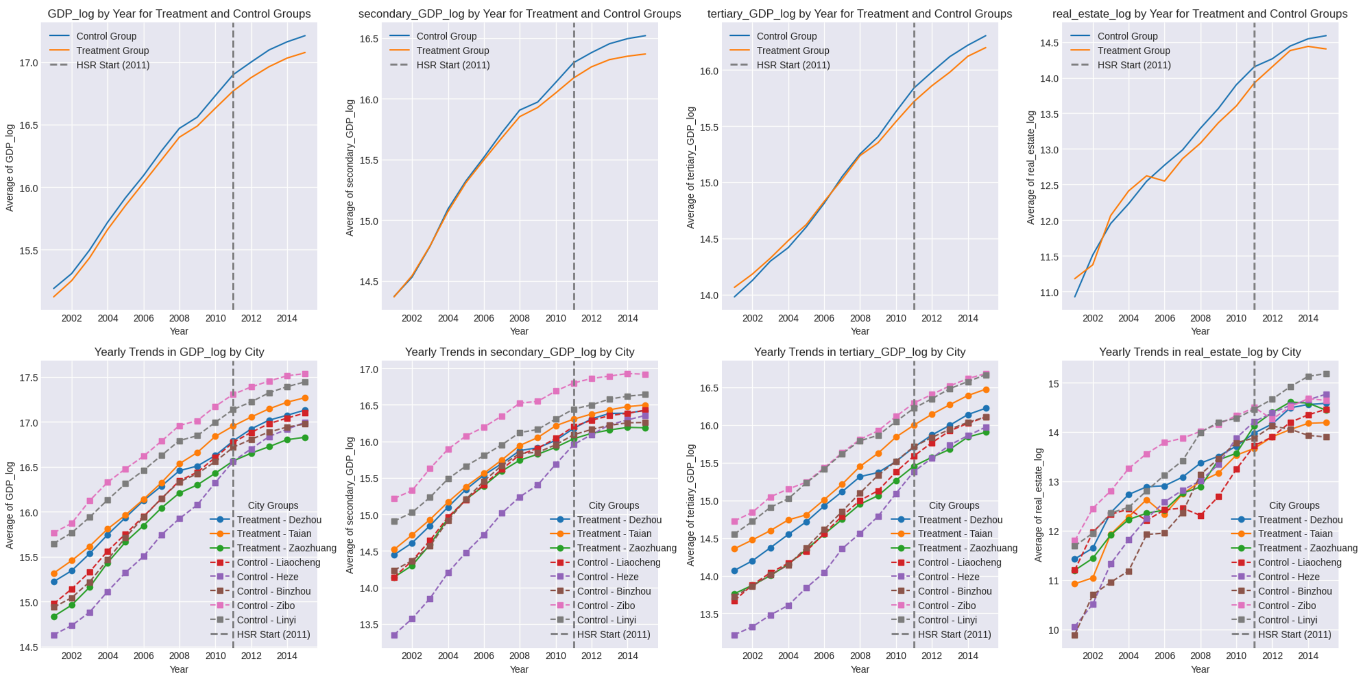
```
                         coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------
Intercept             13.3775      0.094    141.700      0.000      13.190      13.565
C(city)[T.Dezhou]      0.7330      0.135      5.414      0.000       0.465       1.001
C(city)[T.Heze]        0.3276      0.128      2.550      0.012       0.073       0.582
C(city)[T.Liaocheng]   0.3113      0.128      2.424      0.017       0.057       0.566
C(city)[T.Linyi]       1.0279      0.128      8.003      0.000       0.773       1.282
C(city)[T.Taian]       0.3424      0.135      2.529      0.013       0.074       0.611
C(city)[T.Zaozhuang]   0.5292      0.135      3.908      0.000       0.261       0.798
C(city)[T.Zibo]        1.1592      0.128      9.025      0.000       0.905       1.414
treatment_post        -0.1623      0.129     -1.262      0.209      -0.417       0.092
t                      0.2547      0.009     29.658      0.000       0.238       0.272
==============================================================================
Omnibus:                        8.807   Durbin-Watson:                   0.518
Prob(Omnibus):                  0.012   Jarque-Bera (JB):                9.031
Skew:                          -0.531   Prob(JB):                       0.0109
Kurtosis:                       3.823   Cond. No.                         45.4
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

The R-square of the simplified model is still high, and the results of the "treatment_post" items we are concerned about are basically consistent with the former model. Then, the same analysis process is carried out for Guizhou Province.

5.3 Fixed-effects-DID analysis for Guizhou

5.3.1 Parallel Trends Test

In [38]:
```python
plot_all(data_guizhou, cities_guizhou_treatment, cities_guizhou_control, 2011)
```



5.3.2 DID analysis

In [39]:
```python
# Get results for Guizhou Province
results_guizhou = model_results(run_fixed_effects_model, data_guizhou, cities_guizhou_treatme
```

```
////////////////////////////////////////////////////////////////////////////////
GDP_log
////////////////////////////////////////////////////////////////////////////////
                            OLS Regression Results
==============================================================================
Dep. Variable:                GDP_log   R-squared:                       0.986
Model:                            OLS   Adj. R-squared:                  0.982
Method:                 Least Squares   F-statistic:                     214.4
Date:                Mon, 29 Apr 2024   Prob (F-statistic):           1.36e-47
Time:                        01:35:22   Log-Likelihood:                 78.208
No. Observations:                  80   AIC:                            -114.4
Df Residuals:                      59   BIC:                            -64.39
Df Model:                          20
Covariance Type:            nonrobust
=========================================================================================
                            coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------------
Intercept                14.1595      0.054    262.714      0.000      14.052      14.267
C(city)[T.Bijie]          1.0069      0.041     24.634      0.000       0.925       1.089
C(city)[T.Liupanshui]     0.6853      0.041     16.767      0.000       0.604       0.767
C(city)[T.Tongren]        0.3234      0.037      8.630      0.000       0.248       0.398
C(city)[T.Zunyi]          1.3642      0.041     33.376      0.000       1.282       1.446
C(t)[T.-10]               0.1375      0.067      2.051      0.045       0.003       0.272
C(t)[T.-9]                0.2898      0.067      4.323      0.000       0.156       0.424
C(t)[T.-8]                0.4111      0.067      6.132      0.000       0.277       0.545
C(t)[T.-7]                0.5385      0.067      8.032      0.000       0.404       0.673
C(t)[T.-6]                0.6909      0.067     10.305      0.000       0.557       0.825
C(t)[T.-5]                0.8842      0.067     13.189      0.000       0.750       1.018
C(t)[T.-4]                1.0636      0.067     15.865      0.000       0.929       1.198
C(t)[T.-3]                1.2463      0.067     18.590      0.000       1.112       1.380
C(t)[T.-2]                1.4040      0.067     20.942      0.000       1.270       1.538
C(t)[T.-1]                1.5080      0.067     22.493      0.000       1.374       1.642
C(t)[T.0]                 1.5687      0.070     22.341      0.000       1.428       1.709
C(t)[T.1]                 1.6337      0.070     23.266      0.000       1.493       1.774
C(t)[T.2]                 1.6725      0.070     23.819      0.000       1.532       1.813
C(t)[T.3]                 1.7305      0.070     24.645      0.000       1.590       1.871
C(t)[T.4]                 1.8291      0.070     26.049      0.000       1.689       1.970
treatment_post            0.1600      0.052      3.065      0.003       0.056       0.264
==============================================================================
Omnibus:                        1.057   Durbin-Watson:                   0.527
Prob(Omnibus):                  0.590   Jarque-Bera (JB):                1.145
Skew:                           0.238   Prob(JB):                        0.564
Kurtosis:                       2.657   Cond. No.                         18.5
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.



////////////////////////////////////////////////////////////////////////////////
secondary_GDP_log
////////////////////////////////////////////////////////////////////////////////
                            OLS Regression Results
==============================================================================
Dep. Variable:      secondary_GDP_log   R-squared:                       0.972
Model:                            OLS   Adj. R-squared:                  0.963
Method:                 Least Squares   F-statistic:                     103.3
Date:                Mon, 29 Apr 2024   Prob (F-statistic):           1.74e-38
Time:                        01:35:22   Log-Likelihood:                 44.450
No. Observations:                  80   AIC:                            -46.90
Df Residuals:                      59   BIC:                             3.123
Df Model:                          20
Covariance Type:            nonrobust
=========================================================================================
                            coef    std err          t      P>|t|      [0.025      0.975]
```

```
------------------------------------------------------------------------------------------
Intercept               13.1743      0.082    160.287      0.000      13.010      13.339
C(city)[T.Bijie]         1.1356      0.062     18.219      0.000       1.011       1.260
C(city)[T.Liupanshui]    1.1364      0.062     18.232      0.000       1.012       1.261
C(city)[T.Tongren]       0.0872      0.057      1.526      0.132      -0.027       0.202
C(city)[T.Zunyi]         1.6281      0.062     26.121      0.000       1.503       1.753
C(t)[T.-10]              0.1488      0.102      1.456      0.151      -0.056       0.353
C(t)[T.-9]               0.3120      0.102      3.051      0.003       0.107       0.517
C(t)[T.-8]               0.3755      0.102      3.672      0.001       0.171       0.580
C(t)[T.-7]               0.5253      0.102      5.138      0.000       0.321       0.730
C(t)[T.-6]               0.7019      0.102      6.866      0.000       0.497       0.907
C(t)[T.-5]               0.8964      0.102      8.768      0.000       0.692       1.101
C(t)[T.-4]               1.0564      0.102     10.333      0.000       0.852       1.261
C(t)[T.-3]               1.2120      0.102     11.855      0.000       1.007       1.417
C(t)[T.-2]               1.3132      0.102     12.844      0.000       1.109       1.518
C(t)[T.-1]               1.4000      0.102     13.694      0.000       1.195       1.605
C(t)[T.0]                1.4061      0.107     13.131      0.000       1.192       1.620
C(t)[T.1]                1.4936      0.107     13.948      0.000       1.279       1.708
C(t)[T.2]                1.4397      0.107     13.446      0.000       1.225       1.654
C(t)[T.3]                1.4579      0.107     13.615      0.000       1.244       1.672
C(t)[T.4]                1.5781      0.107     14.737      0.000       1.364       1.792
treatment_post           0.2274      0.080      2.858      0.006       0.068       0.387
==============================================================================
Omnibus:                        3.682   Durbin-Watson:                   0.527
Prob(Omnibus):                  0.159   Jarque-Bera (JB):                2.946
Skew:                           0.353   Prob(JB):                        0.229
Kurtosis:                       3.621   Cond. No.                         18.5
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.


////////////////////////////////////////////////////////////////////////////////////
tertiary_GDP_log
////////////////////////////////////////////////////////////////////////////////////
                            OLS Regression Results
==============================================================================
Dep. Variable:       tertiary_GDP_log   R-squared:                       0.986
Model:                            OLS   Adj. R-squared:                  0.981
Method:                 Least Squares   F-statistic:                     207.1
Date:                Mon, 29 Apr 2024   Prob (F-statistic):           3.67e-47
Time:                        01:35:22   Log-Likelihood:                 73.244
No. Observations:                  80   AIC:                            -104.5
Df Residuals:                      59   BIC:                            -54.47
Df Model:                          20
Covariance Type:            nonrobust
==========================================================================================
                            coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------
Intercept               13.2848      0.057    231.655      0.000      13.170      13.400
C(city)[T.Bijie]         0.8266      0.043     19.008      0.000       0.740       0.914
C(city)[T.Liupanshui]    0.4395      0.043     10.106      0.000       0.352       0.527
C(city)[T.Tongren]       0.3192      0.040      8.005      0.000       0.239       0.399
C(city)[T.Zunyi]         1.2095      0.043     27.813      0.000       1.123       1.297
C(t)[T.-10]              0.1487      0.071      2.084      0.041       0.006       0.291
C(t)[T.-9]               0.2722      0.071      3.816      0.000       0.130       0.415
C(t)[T.-8]               0.4872      0.071      6.830      0.000       0.344       0.630
C(t)[T.-7]               0.6050      0.071      8.481      0.000       0.462       0.748
C(t)[T.-6]               0.7492      0.071     10.502      0.000       0.606       0.892
C(t)[T.-5]               0.9495      0.071     13.311      0.000       0.807       1.092
C(t)[T.-4]               1.1693      0.071     16.393      0.000       1.027       1.312
C(t)[T.-3]               1.3767      0.071     19.300      0.000       1.234       1.519
C(t)[T.-2]               1.5362      0.071     21.536      0.000       1.393       1.679
C(t)[T.-1]               1.6708      0.071     23.423      0.000       1.528       1.814
```

```
C(t)[T.0]                  1.7343      0.075     23.212      0.000       1.585       1.884
C(t)[T.1]                  1.8392      0.075     24.617      0.000       1.690       1.989
C(t)[T.2]                  1.9262      0.075     25.782      0.000       1.777       2.076
C(t)[T.3]                  1.9940      0.075     26.689      0.000       1.845       2.144
C(t)[T.4]                  2.0807      0.075     27.849      0.000       1.931       2.230
treatment_post             0.1357      0.056      2.443      0.018       0.025       0.247
==============================================================================
Omnibus:                        1.589   Durbin-Watson:                   0.550
Prob(Omnibus):                  0.452   Jarque-Bera (JB):                1.245
Skew:                           0.075   Prob(JB):                        0.537
Kurtosis:                       2.407   Cond. No.                         18.5
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.



///////////////////////////////////////////////////////////////////////////////////
real_estate_log
///////////////////////////////////////////////////////////////////////////////////
                            OLS Regression Results
==============================================================================
Dep. Variable:        real_estate_log   R-squared:                       0.900
Model:                            OLS   Adj. R-squared:                  0.866
Method:                 Least Squares   F-statistic:                     26.59
Date:                Mon, 29 Apr 2024   Prob (F-statistic):           2.26e-22
Time:                        01:35:22   Log-Likelihood:                -19.276
No. Observations:                  80   AIC:                             80.55
Df Residuals:                      59   BIC:                             130.6
Df Model:                          20
Covariance Type:            nonrobust
==========================================================================================
                             coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------
Intercept                 11.7500      0.182     64.456      0.000      11.385      12.115
C(city)[T.Bijie]           0.9517      0.138      6.884      0.000       0.675       1.228
C(city)[T.Liupanshui]     -0.0565      0.138     -0.408      0.684      -0.333       0.220
C(city)[T.Tongren]         0.5003      0.127      3.947      0.000       0.247       0.754
C(city)[T.Zunyi]           1.1664      0.138      8.437      0.000       0.890       1.443
C(t)[T.-10]                0.2774      0.227      1.224      0.226      -0.176       0.731
C(t)[T.-9]                 0.2979      0.227      1.314      0.194      -0.156       0.752
C(t)[T.-8]                 0.4556      0.227      2.009      0.049       0.002       0.909
C(t)[T.-7]                 0.6687      0.227      2.949      0.005       0.215       1.122
C(t)[T.-6]                 0.9886      0.227      4.360      0.000       0.535       1.442
C(t)[T.-5]                 1.2764      0.227      5.629      0.000       0.823       1.730
C(t)[T.-4]                 1.7719      0.227      7.814      0.000       1.318       2.226
C(t)[T.-3]                 1.8694      0.227      8.244      0.000       1.416       2.323
C(t)[T.-2]                 1.7964      0.227      7.922      0.000       1.343       2.250
C(t)[T.-1]                 1.8187      0.227      8.021      0.000       1.365       2.272
C(t)[T.0]                  1.9153      0.237      8.064      0.000       1.440       2.390
C(t)[T.1]                  2.0683      0.237      8.709      0.000       1.593       2.544
C(t)[T.2]                  2.1725      0.237      9.148      0.000       1.697       2.648
C(t)[T.3]                  2.2705      0.237      9.560      0.000       1.795       2.746
C(t)[T.4]                  2.3607      0.237      9.940      0.000       1.886       2.836
treatment_post             0.0438      0.177      0.248      0.805      -0.309       0.397
==============================================================================
Omnibus:                        1.010   Durbin-Watson:                   0.456
Prob(Omnibus):                  0.604   Jarque-Bera (JB):                1.026
Skew:                          -0.143   Prob(JB):                        0.599
Kurtosis:                       2.524   Cond. No.                         18.5
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

In [40]:

```
# Get results for Guizhou Province for the simplified model
results_guizhou_simplified = model_results(run_fixed_effects_model_simplified, data_guizhou,
```

```
////////////////////////////////////////////////////////////////////////////////
GDP_log
////////////////////////////////////////////////////////////////////////////////
                            OLS Regression Results                            
==============================================================================
Dep. Variable:                GDP_log   R-squared:                       0.971
Model:                            OLS   Adj. R-squared:                  0.969
Method:                 Least Squares   F-statistic:                     410.6
Date:                Mon, 29 Apr 2024   Prob (F-statistic):           3.91e-54
Time:                        01:35:22   Log-Likelihood:                 48.145
No. Observations:                  80   AIC:                            -82.29
Df Residuals:                      73   BIC:                            -65.62
Df Model:                           6                                         
Covariance Type:            nonrobust                                         
=========================================================================================
                            coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------------
Intercept                15.6959      0.045    348.608      0.000      15.606      15.786
C(city)[T.Bijie]          0.9713      0.053     18.434      0.000       0.866       1.076
C(city)[T.Liupanshui]     0.6497      0.053     12.332      0.000       0.545       0.755
C(city)[T.Tongren]        0.3234      0.049      6.593      0.000       0.226       0.421
C(city)[T.Zunyi]          1.3286      0.053     25.216      0.000       1.224       1.434
treatment_post            0.0461      0.061      0.750      0.455      -0.076       0.169
t                         0.1322      0.004     33.819      0.000       0.124       0.140
==============================================================================
Omnibus:                        5.479   Durbin-Watson:                   0.306
Prob(Omnibus):                  0.065   Jarque-Bera (JB):                5.353
Skew:                          -0.633   Prob(JB):                       0.0688
Kurtosis:                       2.937   Cond. No.                         36.6
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.



////////////////////////////////////////////////////////////////////////////////
secondary_GDP_log
////////////////////////////////////////////////////////////////////////////////
                            OLS Regression Results                            
==============================================================================
Dep. Variable:      secondary_GDP_log   R-squared:                       0.949
Model:                            OLS   Adj. R-squared:                  0.945
Method:                 Least Squares   F-statistic:                     226.3
Date:                Mon, 29 Apr 2024   Prob (F-statistic):           4.45e-45
Time:                        01:35:22   Log-Likelihood:                 20.092
No. Observations:                  80   AIC:                            -26.18
Df Residuals:                      73   BIC:                            -9.511
Df Model:                           6                                         
Covariance Type:            nonrobust                                         
=========================================================================================
                            coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------------
Intercept                14.5826      0.064    228.086      0.000      14.455      14.710
C(city)[T.Bijie]          1.0850      0.075     14.502      0.000       0.936       1.234
C(city)[T.Liupanshui]     1.0857      0.075     14.512      0.000       0.937       1.235
C(city)[T.Tongren]        0.0872      0.070      1.252      0.214      -0.052       0.226
C(city)[T.Zunyi]          1.5775      0.075     21.085      0.000       1.428       1.727
treatment_post            0.0655      0.087      0.750      0.456      -0.109       0.239
t                         0.1144      0.006     20.606      0.000       0.103       0.125
==============================================================================
Omnibus:                       12.777   Durbin-Watson:                   0.355
Prob(Omnibus):                  0.002   Jarque-Bera (JB):               14.087
Skew:                          -1.017   Prob(JB):                     0.000873
Kurtosis:                       3.300   Cond. No.                         36.6
==============================================================================
```

```
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.



////////////////////////////////////////////////////////////////////////////////
tertiary_GDP_log
////////////////////////////////////////////////////////////////////////////////
                            OLS Regression Results
==============================================================================
Dep. Variable:       tertiary_GDP_log   R-squared:                       0.974
Model:                            OLS   Adj. R-squared:                  0.972
Method:                 Least Squares   F-statistic:                     450.4
Date:                Mon, 29 Apr 2024   Prob (F-statistic):           1.46e-55
Time:                        01:35:22   Log-Likelihood:                 48.140
No. Observations:                  80   AIC:                            -82.28
Df Residuals:                      73   BIC:                            -65.61
Df Model:                           6
Covariance Type:            nonrobust
=========================================================================================
                            coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------------
Intercept                15.0027      0.045    333.192      0.000      14.913      15.092
C(city)[T.Bijie]          0.7941      0.053     15.072      0.000       0.689       0.899
C(city)[T.Liupanshui]     0.4070      0.053      7.724      0.000       0.302       0.512
C(city)[T.Tongren]        0.3192      0.049      6.506      0.000       0.221       0.417
C(city)[T.Zunyi]          1.1770      0.053     22.338      0.000       1.072       1.282
treatment_post            0.0317      0.061      0.515      0.608      -0.091       0.154
t                         0.1505      0.004     38.492      0.000       0.143       0.158
==============================================================================
Omnibus:                        2.045   Durbin-Watson:                   0.377
Prob(Omnibus):                  0.360   Jarque-Bera (JB):                1.938
Skew:                          -0.373   Prob(JB):                        0.379
Kurtosis:                       2.839   Cond. No.                         36.6
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.



////////////////////////////////////////////////////////////////////////////////
real_estate_log
////////////////////////////////////////////////////////////////////////////////
                            OLS Regression Results
==============================================================================
Dep. Variable:        real_estate_log   R-squared:                       0.860
Model:                            OLS   Adj. R-squared:                  0.849
Method:                 Least Squares   F-statistic:                     74.87
Date:                Mon, 29 Apr 2024   Prob (F-statistic):           3.47e-29
Time:                        01:35:22   Log-Likelihood:                -32.733
No. Observations:                  80   AIC:                             79.47
Df Residuals:                      73   BIC:                             96.14
Df Model:                           6
Covariance Type:            nonrobust
=========================================================================================
                            coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------------
Intercept                13.7935      0.124    111.472      0.000      13.547      14.040
C(city)[T.Bijie]          0.8807      0.145      6.082      0.000       0.592       1.169
C(city)[T.Liupanshui]    -0.1275      0.145     -0.880      0.382      -0.416       0.161
C(city)[T.Tongren]        0.5003      0.135      3.711      0.000       0.232       0.769
C(city)[T.Zunyi]          1.0954      0.145      7.565      0.000       0.807       1.384
treatment_post           -0.1834      0.169     -1.086      0.281      -0.520       0.153
t                         0.1706      0.011     15.876      0.000       0.149       0.192
```

```
==============================================================================
Omnibus:                        3.283   Durbin-Watson:                   0.441
Prob(Omnibus):                  0.194   Jarque-Bera (JB):                3.255
Skew:                          -0.466   Prob(JB):                        0.196
Kurtosis:                       2.673   Cond. No.                         36.6
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

6 Results

6.1 Results display

Here, we list all the results corresponding to the "treatment_post" item for comparison.

In [41]:
```
# Results of Shandong province
results_shandong
```

Out[41]:

	Response Variable	Coefficient	Std Error	t-value	P-value	Conf. Int. Low	Conf. Int. High
0	GDP_log	-0.062684	0.032842	-1.908623	0.059267	-0.127867	0.002499
1	secondary_GDP_log	-0.106793	0.058716	-1.818824	0.072024	-0.223328	0.009741
2	tertiary_GDP_log	-0.125935	0.041832	-3.010518	0.003324	-0.208960	-0.042911
3	real_estate_log	-0.082191	0.128173	-0.641252	0.522871	-0.336579	0.172196

In [42]:
```
# Results of Shandong province (simplified model)
results_shandong_simplified
```

Out[42]:

	Response Variable	Coefficient	Std Error	t-value	P-value	Conf. Int. Low	Conf. Int. High
0	GDP_log	-0.128755	0.044617	-2.885768	0.004700	-0.217175	-0.040334
1	secondary_GDP_log	-0.235012	0.076132	-3.086915	0.002560	-0.385887	-0.084137
2	tertiary_GDP_log	-0.109188	0.041957	-2.602362	0.010534	-0.192338	-0.026039
3	real_estate_log	-0.162275	0.128540	-1.262448	0.209457	-0.417010	0.092461

In [43]:
```
# Results of Guizhou province
results_guizhou
```

Out[43]:

	Response Variable	Coefficient	Std Error	t-value	P-value	Conf. Int. Low	Conf. Int. High
0	GDP_log	0.159980	0.052192	3.065211	0.003278	0.055544	0.264415
1	secondary_GDP_log	0.227448	0.079591	2.857691	0.005887	0.068186	0.386710
2	tertiary_GDP_log	0.135689	0.055533	2.443383	0.017561	0.024567	0.246810
3	real_estate_log	0.043796	0.176528	0.248096	0.804922	-0.309436	0.397028

In [44]:
```
# Results of Guizhou province (simplified model)
results_guizhou_simplified
```

Out[44]:

	Response Variable	Coefficient	Std Error	t-value	P-value	Conf. Int. Low	Conf. Int. High
0	GDP_log	0.046134	0.061473	0.750473	0.455382	-0.076382	0.168650
1	secondary_GDP_log	0.065459	0.087292	0.749886	0.455733	-0.108513	0.239432
2	tertiary_GDP_log	0.031689	0.061477	0.515462	0.607788	-0.090835	0.154213
3	real_estate_log	-0.183399	0.168946	-1.085549	0.281250	-0.520107	0.153309

6.2 Interpretation

First, let's compare the performance of the two models within their respective groups.

- In the Shandong Province group, both models performed well and the results tended to be consistent.
- In the Guizhou Province group, the two models are quite different. The "treatment_post" item we care about is not significant for all indicators in the simplified model. We were unable to identify the effect, which may be due to the linear treatment in the simplified model masking its effect.

We can then compare the results of the two sets of analyses.

- For Shandong Province, which is the economically developed region where HSR was built earlier, the coefficients of the four indicators in both models are negative, indicating that the opening of the HSR has brought negative impacts.
- For Guizhou Province, the estimated effects of the two models are positive and insignificant respectively. It shows that the opening of HSR will at least have no negative impact.

7 Discussion and conclusion

In conclusion, the role of high-speed rail in promoting underdeveloped regions is greater than that in developed regions. This is contrary to the view that newly built HSR has lower promotion effects. However, it is not reflected in the different degree of promotion, but because the role of HSR in Shandong area is negative.

This counterintuitive conclusion has also appeared in other studies. A study on residents' income showed that HSR had a positive or no impact in the Midwest and a negative impact in the East (Jin et al., 2022). The reason behind this may be that the siphon effect brought by HSR is greater than the diffusion effect (Wang et al., 2022). The two ends of Shandong Province's HSR are Beijing and Shanghai, which may make its economic population more easily transferred and lost.

In addition, we can find that the coefficients of the secondary industry is the largest. People generally believe that HSR has a closer relationship with the tertiary industry because of tourism. However, studies have shown that the HSR network has accelerated the transfer of low-end manufacturing companies to less developed regions. In comparison, tourism is only a side effect. This may be the reason why the secondary industry has been more significantly affected.

Finally, although people generally believe that HSR will significantly promote real estate investment, the real-estate-related data we use is not suitable. This may be due to the low quality of the data. Limited by the data, this study failed to test more indicators such as GDP per capita. Future research can combine more reasonable new indicators to analyze a wider range.

References

Chen, C.-L. and Hall, P. (2012) 'The wider spatial-economic impacts of high-speed trains: a comparative case study of Manchester and Lille sub-regions', Journal of Transport Geography, 24, pp. 89–110. Available at: https://doi.org/10.1016/j.jtrangeo.2011.09.002.

Chen, Z. et al. (2023) 'The Influence of High-speed Rail on the Level and Structure of Urban Economic Vitality: An Empirical Study Based on Real GDP Data', Applied Spatial Analysis and Policy [Preprint]. Available at: https://doi.org/10.1007/s12061-023-09554-7.

Chi, F. and Han, H. (2023) 'The Impact of High-Speed Rail on Economic Development: A County-Level Analysis', Land, 12(4), p. 874. Available at: https://doi.org/10.3390/land12040874.

China Statistical Yearbook 2023 (no date) Available at: https://www.stats.gov.cn/sj/ndsj/2023/indexeh.htm (Accessed: 29 April 2024).

DataV.GeoAtlas Geography Gadget Series (no date). Available at: https://datav.aliyun.com/portal/school/atlas/area_selector (Accessed: 29 April 2024).

Deng, T. et al. (2020) 'Did high-speed railway cause urban space expansion? ——Empirical evidence from China's prefecture-level cities', Research in Transportation Economics, 80, p. 100840. Available at: https://doi.org/10.1016/j.retrec.2020.100840.

Fan, X. and Xu, Y. (2023) 'Does high-speed railway promote urban innovation? Evidence from China', Socio-Economic Planning Sciences, 86, p. 101464. Available at: https://doi.org/10.1016/j.seps.2022.101464.

Gis5g (no date) Statistical Yearbook – gis5g. (Chinese) Available at: http://www.gis5g.com/data/tjnj?id=267 (Accessed: 29 April 2024).

'High-speed rail in China' (2024) Wikipedia. Available at: https://en.wikipedia.org/w/index.php?title=High-speed_rail_in_China&oldid=1220267490 (Accessed: 29 April 2024).

Imai, K. and Kim, I.S. (2019) 'When Should We Use Unit Fixed Effects Regression Models for Causal Inference with Longitudinal Data?', American Journal of Political Science, 63(2), pp. 467–490. Available at: https://doi.org/10.1111/ajps.12417.

Jin, M. et al. (2022) 'Heterogeneous impact of high speed railway on income distribution: A case study in China', Socio-Economic Planning Sciences, 79, p. 101128. Available at: https://doi.org/10.1016/j.seps.2021.101128.

Ke, X. et al. (2017) 'Do China's high-speed-rail projects promote local economy?—New evidence from a panel data approach', China Economic Review, 44, pp. 203–226. Available at: https://doi.org/10.1016/j.chieco.2017.02.008.

Liang, Y. et al. (2020) 'Effectiveness of high-speed railway on regional economic growth for less developed areas', Journal of Transport Geography, 82, p. 102621. Available at: https://doi.org/10.1016/j.jtrangeo.2019.102621.

Lin, Y. (2017) 'Travel costs and urban specialization patterns: Evidence from China's high speed railway system', Journal of Urban Economics, 98, pp. 98–123. Available at:

https://doi.org/10.1016/j.jue.2016.11.002.

Oakes, J.M. and Kaufman, J.S. (2017) Methods in Social Epidemiology. John Wiley & Sons.

Reichardt, A. et al. (2012) 'Sustainable Building Certification and the Rent Premium: A Panel Data Approach', Journal of Real Estate Research, 34(1), pp. 99–126. Available at: https://doi.org/10.1080/10835547.2012.12091325.

Wang, F. et al. (2022) 'High-speed railway development and its impact on urban economy and population: A case study of nine provinces along the Yellow River, China', Sustainable Cities and Society, 87, p. 104172. Available at: https://doi.org/10.1016/j.scs.2022.104172.