fdi-province

August 24, 2024

[174]:
```
%%HTML
<script src="require.js"></script>
```

```
<IPython.core.display.HTML object>
```

[175]:
```
# Library cell
import pandas as pd
import geopandas as gpd
import geoplot
import geoplot.crs as gcrs
import matplotlib.pyplot as plt
import plotly.express as px
# ignore warnings
import warnings
warnings.filterwarnings('ignore')
import plotly.io as pio
pio.renderers.default='notebook'
```

[176]:
```
# Function cell
# Function cell
## Find non-numeric values
def find_non_numeric_values(df):
    non_numeric_columns = df.select_dtypes(include=['object']).columns
    non_numeric_values = {}
    for col in non_numeric_columns:
        # Change the column to numeric type, if it isn't numeric, it will be␣
 ↪converted to NaN
        temp_col = pd.to_numeric(df[col], errors='coerce')
        # Fill the NaN values with the original values
        non_numeric_data = df[temp_col.isna() & df[col].notna()]
        if not non_numeric_data.empty:
            non_numeric_values[col] = non_numeric_data[col].tolist()
    return non_numeric_values

## Remove non-numeric values
def remove_commas_and_convert(df):
    non_numeric_columns = df.select_dtypes(include=['object']).columns
```

```
    for col in non_numeric_columns:
        # Check if the column contains any non-numeric values
        try:
            # Remove commas from the column
            temp_col = df[col] = df[col].str.replace(',', '')
            temp_col_numeric = pd.to_numeric(temp_col, errors='raise')
            # If the column can be converted to numeric, replace the original␣
 ↪column with the new column
            df[col] = temp_col_numeric
        except ValueError:
            # If the column contains non-numeric values, keep it
            continue
    return df
```

0.1 LOAD DATA

[177]:
```
data = gpd.read_file(r'D:
 ↪\Repo-train\Jnotebook\FDI_Analytics\geo\diaphantinhenglish.geojson')
df = pd.read_csv(r'D:
 ↪\Repo-train\Jnotebook\FDI_Analytics\dataset\fdi_provinces_en.csv')
print(type(data))
```

```
<class 'geopandas.geodataframe.GeoDataFrame'>
```

[178]:
```
df.head()
```

[178]:

	Order	Provinces	Number of new projects	Newly registered capital (million USD)	Adjusted project number	Adjusted capital (million USD)
0	1	TP. Ho Chi Minh	836	1006.69	222	619.07
1	2	Hai Phong	52	2464.32	38	429.24
2	3	Ha Noi	453	1922.76	159	504.47
3	4	Binh Duong	256	1630.52	130	641.97
4	5	Dong Nai	91	1043.74	136	921.05

```
  Number of times of capital contribution to buy shares  \
0                                                 1935
1                                                   27
2                                                  228
3                                                   25
4                                                   55

  Value of capital contribution, share purchase\n(million USD)  Year
0                                                1802.56           2016
1                                                  96.34           2016
2                                                 367.21           2016
3                                                  94.72           2016
4                                                 273.44           2016
```

[179]:
```
df.tail()
```

[179]:
```
     Order   Provinces Number of new projects  \
436    437    Ha Giang                    NaN
437    438    Lai Chau                    NaN
438    439     Lao Cai                    NaN
439    440  Quang Binh                    NaN
440    441      Son La                    NaN

    Newly registered capital (million USD) Adjusted project number  \
436                                    NaN                     NaN
437                                    NaN                     NaN
438                                    NaN                     NaN
439                                    NaN                     NaN
440                                    NaN                     NaN

    Adjusted capital (million USD)  \
436                            NaN
437                            NaN
438                            NaN
439                            NaN
440                            NaN

    Number of times of capital contribution to buy shares  \
436                                                   NaN
437                                                   NaN
438                                                   NaN
439                                                   NaN
440                                                   NaN

    Value of capital contribution, share purchase\n(million USD)  Year
436                                                   NaN           2022
437                                                   NaN           2022
```

438	NaN	2022
439	NaN	2022
440	NaN	2022

0.2 Analyze the data

[180]:
```
# Drop column Order
n_df = df.drop(columns=['Order'])
# Show shape data
print(n_df.shape, end='\n ---------------- \n')
# Show info data
print(n_df.info(), end='\n ---------------- \n')
# Check for Duplicate
print(n_df.nunique(), end='\n ---------------- \n')
# Check data exist nan or not (bool)
print(n_df.isnull().any(), end='\n ---------------- \n')
# Check for missing value
print(n_df.isna().sum(), end='\n ---------------- \n')
```

```
(441, 8)
 ----------------
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 441 entries, 0 to 440
Data columns (total 8 columns):
 #   Column                                                  Non-Null Count
Dtype
---  ------                                                  --------------
-----
 0   Provinces                                               441 non-null
object
 1   Number of new projects                                  386 non-null
object
 2   Newly registered capital (million USD)                  387 non-null
object
 3   Adjusted project number                                 344 non-null
object
 4   Adjusted capital (million USD)                          344 non-null
object
 5   Number of times of capital contribution to buy shares   378 non-null
object
 6   Value of capital contribution, share purchase
(million USD)  377 non-null    object
 7   Year                                                    441 non-null
int64
dtypes: int64(1), object(7)
memory usage: 27.7+ KB
None
```

```
 ----------------
Provinces                                                      63
Number of new projects                                        100
Newly registered capital (million USD)                        351
Adjusted project number                                        76
Adjusted capital (million USD)                                282
Number of times of capital contribution to buy shares          99
Value of capital contribution, share purchase\n(million USD)  330
Year                                                            7
dtype: int64
 ----------------
Provinces                                                     False
Number of new projects                                         True
Newly registered capital (million USD)                         True
Adjusted project number                                        True
Adjusted capital (million USD)                                 True
Number of times of capital contribution to buy shares          True
Value of capital contribution, share purchase\n(million USD)   True
Year                                                          False
dtype: bool
 ----------------
Provinces                                                      0
Number of new projects                                        55
Newly registered capital (million USD)                        54
Adjusted project number                                       97
Adjusted capital (million USD)                                97
Number of times of capital contribution to buy shares         63
Value of capital contribution, share purchase\n(million USD)  64
Year                                                           0
dtype: int64
 ----------------
```

Observations

- The shape of dataset `fdi_provinces_en.csv` is 441 rows and 8 columns
- Only `Year` column dftype int, so we will convert some columns to numeric for consistency to calculate and explore the data.
- Check all columns to get boolean values indicating if missing values exist and determine which columns have missing values
- Only `Year` column not exist missing value.

0.3 Data Cleaning

0.3.1 Step-by-step

1. Get all "*not numeric*" from all columns with func `find_non_numeric_values()`
2. Format numeric with func `remove_commas_and_convert()`
3. Remove special character
4. Fill all NaN to 0

5. Drop `Province` and `Year` column for consistency data to numeric
6. Re-execute `find_non_numeric_values()` to check result
7. Random select rows to print for review

[181]:
```
## Check for not numeric value
non_numeric_dict = find_non_numeric_values(n_df)
if non_numeric_dict:
    for col, values in non_numeric_dict.items():
        print(f"Column '{col}' have values not numeric:")
        print(values)
else:
    print("No non-numeric values found.")
```

```
Column 'Provinces' have values not numeric:
['TP. Ho Chi Minh', 'Hai Phong', 'Ha Noi', 'Binh Duong', 'Dong Nai', 'Bac
Giang', 'Bac Ninh', 'Long An', 'Ha Nam', 'Tay Ninh', 'Phu Yen', 'Quang Ninh',
'Ba Ria - Vung Tau', 'Hai Duong', 'Tien Giang', 'Hung Yen', 'Ha Tinh', 'Vinh
Phuc', 'Nam Dinh', 'Tra Vinh', 'Can Tho', 'Thanh Hoa', 'Phu Tho', 'Thai Nguyen',
'Vinh Long', 'Quang Nam', 'Binh Phuoc', 'Da Nang', 'Ninh Binh', 'Ninh Thuan',
'Binh Dinh', 'Nghe An', 'Hau Giang', 'Khanh Hoa', 'Thai Binh', 'Tuyen Quang',
'Quang Binh', 'Lam Dong', 'Ben Tre', 'Ca Mau', 'Thua Thien Hue', 'Lao Cai',
'Quang Ngai', 'Dong Thap', 'Ha Giang', 'An Giang', 'Hoa Binh', 'Lang Son', 'Binh
Thuan', 'Kon Tum', 'Soc Trang', 'Kien Giang', 'Quang Tri', 'Yen Bai', 'Dak Lak',
'Dak Nong', 'Gia Lai', 'Bac Kan', 'Bac Lieu', 'Dien Bien', 'Cao Bang', 'Lai
Chau', 'Son La', 'TP. Ho Chi Minh', 'Bac Ninh', 'Thanh Hoa', 'Binh Duong',
'Khanh Hoa', 'Ha Noi', 'Nam Dinh', 'Dong Nai', 'Kien Giang', 'Tay Ninh', 'Hai
Phong', 'Bac Giang', 'Ba Ria - Vung Tau', 'Hung Yen', 'Binh Phuoc', 'Long An',
'Quang Ngai', 'Hai Duong', 'Ninh Thuan', 'Ha Nam', 'Yen Bai', 'Ben Tre', 'Ninh
Binh', 'Phu Tho', 'Quang Binh', 'Vinh Phuc', 'Binh Dinh', 'Tien Giang', 'Tra
Vinh', 'Da Nang', 'Quang Nam', 'Vinh Long', 'Nghe An', 'Thai Nguyen', 'Thai
Binh', 'Ha Tinh', 'Dong Thap', 'Lam Dong', 'Dak Lak', 'Quang Ninh', 'Hoa Binh',
'Binh Thuan', 'Can Tho', 'Dak Nong', 'Ca Mau', 'Soc Trang', 'Lao Cai', 'Son La',
'Cao Bang', 'An Giang', 'Thua Thien Hue', 'Dien Bien', 'Ha Giang', 'Quang Tri',
'Lang Son', 'Tuyen Quang', 'Phu Yen', 'Kon Tum', 'Hau Giang', 'Bac Lieu', 'Bac
Kan', 'Gia Lai', 'Lai Chau', 'Ha Noi', 'TP. Ho Chi Minh', 'Hai Phong', 'Binh
Duong', 'Ba Ria - Vung Tau', 'Dong Nai', 'Thua Thien Hue', 'Bac Ninh', 'Tay
Ninh', 'Long An', 'Hai Duong', 'Bac Giang', 'Binh Phuoc', 'Hung Yen', 'Quang
Nam', 'Thai Nguyen', 'Ha Nam', 'Ninh Thuan', 'Quang Ninh', 'Ben Tre', 'Vinh
Phuc', 'Bac Lieu', 'Quang Ngai', 'Thanh Hoa', 'Kien Giang', 'Da Nang', 'Nam
Dinh', 'Tien Giang', 'Hoa Binh', 'Ninh Binh', 'Vinh Long', 'Phu Tho', 'Binh
Dinh', 'Tra Vinh', 'Ha Tinh', 'Khanh Hoa', 'Soc Trang', 'Thai Binh', 'Dak Nong',
'Ca Mau', 'Can Tho', 'Quang Binh', 'Dak Lak', 'Tuyen Quang', 'Nghe An', 'Binh
Thuan', 'Phu Yen', 'Lang Son', 'Kon Tum', 'Lam Dong', 'Yen Bai', 'Dong Thap',
'Hau Giang', 'An Giang', 'Son La', 'Lao Cai', 'Quang Tri', 'Ha Giang', 'Bac
Kan', 'Cao Bang', 'Dien Bien', 'Lai Chau', 'Gia Lai', 'Ha Noi', 'TP. Ho Chi
Minh', 'Binh Duong', 'Dong Nai', 'Bac Ninh', 'Hai Phong', 'Tay Ninh', 'Bac
Giang', 'Ba Ria - Vung Tau', 'Ha Nam', 'Long An', 'Hai Duong', 'Thai Nguyen',
'Vinh Phuc', 'Da Nang', 'Hung Yen', 'Binh Phuoc', 'Tien Giang', 'Thanh Hoa',
```

'Phu Tho', 'Thua Thien Hue', 'Nghe An', 'Quang Ninh', 'Phu Yen', 'Khanh Hoa',
'Quang Nam', 'Binh Thuan', 'Vinh Long', 'Ninh Binh', 'Quang Ngai', 'Ninh Thuan',
'Bac Lieu', 'Soc Trang', 'Tra Vinh', 'Binh Dinh', 'Ca Mau', 'Hau Giang', 'Can
Tho', 'Thai Binh', 'An Giang', 'Nam Dinh', 'Ben Tre', 'Ha Tinh', 'Lam Dong',
'Kien Giang', 'Quang Tri', 'Tuyen Quang', 'Dong Thap', 'Yen Bai', 'Kon Tum',
'Bac Kan', 'Dak Lak', 'Lang Son', 'Dien Bien', 'Lao Cai', 'Quang Binh', 'Ha
Giang', 'Son La', 'Dak Nong', 'Cao Bang', 'Lai Chau', 'Hoa Binh', 'Gia Lai',
'TP. Ho Chi Minh', 'Bac Lieu', 'Ha Noi', 'Ba Ria - Vung Tau', 'Binh Duong', 'Hai
Phong', 'Dong Nai', 'Bac Ninh', 'Bac Giang', 'Long An', 'Ha Nam', 'Ben Tre',
'Tay Ninh', 'Vinh Phuc', 'Hai Duong', 'Quang Ninh', 'Hung Yen', 'Thai Nguyen',
'Binh Phuoc', 'Thanh Hoa', 'Phu Tho', 'Quang Binh', 'Vinh Long', 'Da Nang',
'Nghe An', 'Tien Giang', 'Dak Nong', 'Quang Ngai', 'Tra Vinh', 'Ninh Binh',
'Binh Thuan', 'Nam Dinh', 'Thai Binh', 'Khanh Hoa', 'Soc Trang', 'Binh Dinh',
'Can Tho', 'Thua Thien Hue', 'Ca Mau', 'Hoa Binh', 'Quang Tri', 'Lam Dong',
'Dong Thap', 'Ha Tinh', 'Kien Giang', 'Tuyen Quang', 'Gia Lai', 'Yen Bai', 'Lao
Cai', 'An Giang', 'Hau Giang', 'Dak Lak', 'Son La', 'Phu Yen', 'Kon Tum', 'Dien
Bien', 'Cao Bang', 'Quang Nam', 'Ninh Thuan', 'Bac Kan', 'Ha Giang', 'Lai Chau',
'Lao Cai', 'Hai Phong', 'Long An', 'TP. Ho Chi Minh', 'Binh Duong', 'Bac Ninh',
'Ha Noi', 'Dong Nai', 'Can Tho', 'Bac Giang', 'Quang Ninh', 'Tay Ninh', 'Vinh
Phuc', 'Hung Yen', 'Phu Tho', 'Thai Binh', 'Dak Lak', 'Binh Phuoc', 'Ba Ria -
Vung Tau', 'Hai Duong', 'Nghe An', 'Ha Nam', 'Thai Nguyen', 'Thanh Hoa', 'Thua
Thien Hue', 'Da Nang', 'Ninh Binh', 'Hau Giang', 'Kon Tum', 'Nam Dinh', 'Tien
Giang', 'Binh Dinh', 'Quang Tri', 'Ninh Thuan', 'Quang Binh', 'Yen Bai', 'Vinh
Long', 'Dong Thap', 'Quang Ngai', 'Quang Nam', 'Khanh Hoa', 'Ca Mau', 'Binh
Thuan', 'Lam Dong', 'Tra Vinh', 'An Giang', 'Ha Tinh', 'Lao Cai', 'Dak Nong',
'Kien Giang', 'Lang Son', 'Gia Lai', 'Phu Yen', 'Cao Bang', 'Lai Chau', 'Soc
Trang', 'Bac Lieu', 'Hoa Binh', 'Tuyen Quang', 'Ben Tre', 'Bac Kan', 'Dien
Bien', 'Ha Giang', 'Son La', 'TP. Ho Chi Minh', 'Binh Duong', 'Quang Ninh', 'Bac
Ninh', 'Hai Phong', 'Ha Noi', 'Thai Nguyen', 'Dong Nai', 'Bac Giang', 'Ba Ria -
Vung Tau', 'Nghe An', 'Long An', 'Hung Yen', 'Phu Tho', 'Tay Ninh', 'Ha Nam',
'Hai Duong', 'Thai Binh', 'Ha Tinh', 'Vinh Phuc', 'Binh Phuoc', 'Tien Giang',
'Thua Thien Hue', 'Can Tho', 'Vinh Long', 'Da Nang', 'Soc Trang', 'Thanh Hoa',
'Ninh Thuan', 'Quang Ngai', 'Quang Nam', 'Ninh Binh', 'Nam Dinh', 'Binh Dinh',
'Binh Thuan', 'An Giang', 'Dak Lak', 'Yen Bai', 'Khanh Hoa', 'Kien Giang', 'Tra
Vinh', 'Ben Tre', 'Phu Yen', 'Lang Son', 'Hoa Binh', 'Tuyen Quang', 'Dak Nong',
'Kon Tum', 'Quang Tri', 'Hau Giang', 'Ca Mau', 'Gia Lai', 'Lam Dong', 'Bac
Lieu', 'Bac Kan', 'Cao Bang', 'Dien Bien', 'Dong Thap', 'Ha Giang', 'Lai Chau',
'Lao Cai', 'Quang Binh', 'Son La']
Column 'Number of new projects' have values not numeric:
[' - ', ' - ', ' - ', ' - ', ' - ', ' - ', ' - ', ' - ', ' -
', ' - ', ' - ', ' - ', ' - ', ' - ', ' - ', ' - ', ' - ', ' -
', ' - ', ' - ', ' - ', ' - ']
Column 'Newly registered capital (million USD)' have values not numeric:
['2,313.95', '3,159.40', '1,356.46', '2,584.86', '1,111.25', '2,134.30',
'1,002.38', '1,342.30', ' - ', '5,041.05', '1,216.58', '1,803.51', '1,382.06',
'1,841.35', '1,546.59', '1,296.70', '4,000.00', '1,064.13', ' - ', ' - ', '
- ', ' - ', ' - ', ' - ', ' - ', '3,518.84', '1,170.51', '1,316.82',
'1,011.55', ' - ', ' - ', ' - ', ' - ', ' - ', ' - ', ' - ',

```
'1,909.08', '2,181.17', '1,139.00', ' -   ', ' -   ', ' -   ', ' -   ', ' -   ',
' -   ', ' -   ', ' -   ']
Column 'Adjusted project number' have values not numeric:
[' -   ', ' -   ', ' -   ', ' -   ', ' -   ', ' -   ', ' -   ', ' -   ', ' -
', ' -   ', ' -   ', ' -   ', ' -   ', ' -   ', ' -   ', ' -   ', ' -   ', ' -
', ' -   ', ' -   ', ' -   ', ' -   ', ' -   ', ' -   ', ' -   ', ' -   ', ' -
', ' -   ', ' -   ', ' -   ', ' -   ', ' -   ', ' -   ', ' -   ', ' -   ', ' -
', ' -   ', ' -   ', ' -   ', ' -   ', ' -   ']
Column 'Adjusted capital (million USD)' have values not numeric:
['1,000.11', '2,888.34', '1,117.00', '1,829.64', '1,140.00', ' -   ',
'1,261.91', '1,489.66', ' -   ', ' -   ', ' -   ', ' -   ', ' -   ', ' -   ', '
-   ', ' -   ', ' -   ', ' -   ', ' -   ', ' -   ', ' -   ', ' -   ', ' -   ', '
-   ', '2,727.59', '1,124.28', ' -   ', ' -   ', ' -   ', ' -   ', ' -   ', ' -
', ' -   ', ' -   ', ' -   ', ' -   ', ' -   ', ' -   ', ' -   ', ' -   ', ' -
', ' -   ', ' -   ', ' -   ', '1,600.72', ' -   ', '1,685.63', '1,212.16', ' -
', ' -   ', ' -   ', ' -   ', ' -   ', ' -   ', ' -   ', ' -   ', ' -   ', ' -
', ' -   ', ' -   ', ' -   ', ' -   ']
Column 'Number of times of capital contribution to buy shares' have values not
numeric:
['2,788', '3,710', '1,351', '5,720', '3,640', ' -   ', ' -   ', '2,289', ' -
', ' -   ', ' -   ', ' -   ', '2,411', ' -   ', ' -   ']
Column 'Value of capital contribution, share purchase
(million USD)' have values not numeric:
['3,191.90', '1,703.13', '4,993.11', '6,472.60', '5,595.33', '1,009.55',
'3,177.38', ' -   ', '1,611.82', ' -   ', '2,217.58', '1,927.21', ' -   ', ' -
', ' -   ', ' -   ', '1,738.61', '1,196.48', ' -   ', ' -   ']
```

[182]:
```
## Drop comma value
n_df = remove_commas_and_convert(n_df)
```

[183]:
```
# Drop ' - ' value
### Drop ' - ' value column 'Number of new projects'
n_df['Number of new projects'] = n_df['Number of new projects'].
 ↪replace(to_replace=r'[^0-9.]', value=0, regex=True)
### Drop ' - ' value column 'Newly registered capital (million USD)'
n_df['Newly registered capital (million USD)'] = n_df['Newly registered capital␣
 ↪(million USD)'].replace(to_replace=r'[^0-9.]', value=0, regex=True)
### Drop ' - ' value column 'Adjusted project number'
n_df['Adjusted project number'] = n_df['Adjusted project number'].
 ↪replace(to_replace=r'[^0-9.]', value=0, regex=True)
### Drop ' - ' value column 'Adjusted capital (million USD)'
n_df['Adjusted capital (million USD)'] = n_df['Adjusted capital (million USD)'].
 ↪replace(to_replace=r'[^0-9.]', value=0, regex=True)
### Drop ' - ' value column 'Number of times of capital contribution to buy␣
 ↪shares'
```

```
n_df['Number of times of capital contribution to buy shares'] = n_df['Number of␣
 ↪times of capital contribution to buy shares'].replace(to_replace=r'[^0-9.]',␣
 ↪value=0, regex=True)
### Drop ' - ' value column 'Value of capital contribution, share␣
 ↪purchase\n(million USD)'
n_df['Value of capital contribution, share purchase\n(million USD)'] =␣
 ↪n_df['Value of capital contribution, share purchase\n(million USD)'].
 ↪replace(to_replace=r'[^0-9.]', value=0, regex=True)
```

[184]:
```
## Check for not numeric value
non_numeric_dict = find_non_numeric_values(n_df)
if non_numeric_dict:
    for col, values in non_numeric_dict.items():
        print(f"Column '{col}' have values not numeric:")
        print(values)
else:
    print("No non-numeric values found.")
```

```
Column 'Provinces' have values not numeric:
['TP. Ho Chi Minh', 'Hai Phong', 'Ha Noi', 'Binh Duong', 'Dong Nai', 'Bac
Giang', 'Bac Ninh', 'Long An', 'Ha Nam', 'Tay Ninh', 'Phu Yen', 'Quang Ninh',
'Ba Ria - Vung Tau', 'Hai Duong', 'Tien Giang', 'Hung Yen', 'Ha Tinh', 'Vinh
Phuc', 'Nam Dinh', 'Tra Vinh', 'Can Tho', 'Thanh Hoa', 'Phu Tho', 'Thai Nguyen',
'Vinh Long', 'Quang Nam', 'Binh Phuoc', 'Da Nang', 'Ninh Binh', 'Ninh Thuan',
'Binh Dinh', 'Nghe An', 'Hau Giang', 'Khanh Hoa', 'Thai Binh', 'Tuyen Quang',
'Quang Binh', 'Lam Dong', 'Ben Tre', 'Ca Mau', 'Thua Thien Hue', 'Lao Cai',
'Quang Ngai', 'Dong Thap', 'Ha Giang', 'An Giang', 'Hoa Binh', 'Lang Son', 'Binh
Thuan', 'Kon Tum', 'Soc Trang', 'Kien Giang', 'Quang Tri', 'Yen Bai', 'Dak Lak',
'Dak Nong', 'Gia Lai', 'Bac Kan', 'Bac Lieu', 'Dien Bien', 'Cao Bang', 'Lai
Chau', 'Son La', 'TP. Ho Chi Minh', 'Bac Ninh', 'Thanh Hoa', 'Binh Duong',
'Khanh Hoa', 'Ha Noi', 'Nam Dinh', 'Dong Nai', 'Kien Giang', 'Tay Ninh', 'Hai
Phong', 'Bac Giang', 'Ba Ria - Vung Tau', 'Hung Yen', 'Binh Phuoc', 'Long An',
'Quang Ngai', 'Hai Duong', 'Ninh Thuan', 'Ha Nam', 'Yen Bai', 'Ben Tre', 'Ninh
Binh', 'Phu Tho', 'Quang Binh', 'Vinh Phuc', 'Binh Dinh', 'Tien Giang', 'Tra
Vinh', 'Da Nang', 'Quang Nam', 'Vinh Long', 'Nghe An', 'Thai Nguyen', 'Thai
Binh', 'Ha Tinh', 'Dong Thap', 'Lam Dong', 'Dak Lak', 'Quang Ninh', 'Hoa Binh',
'Binh Thuan', 'Can Tho', 'Dak Nong', 'Ca Mau', 'Soc Trang', 'Lao Cai', 'Son La',
'Cao Bang', 'An Giang', 'Thua Thien Hue', 'Dien Bien', 'Ha Giang', 'Quang Tri',
'Lang Son', 'Tuyen Quang', 'Phu Yen', 'Kon Tum', 'Hau Giang', 'Bac Lieu', 'Bac
Kan', 'Gia Lai', 'Lai Chau', 'Ha Noi', 'TP. Ho Chi Minh', 'Hai Phong', 'Binh
Duong', 'Ba Ria - Vung Tau', 'Dong Nai', 'Thua Thien Hue', 'Bac Ninh', 'Tay
Ninh', 'Long An', 'Hai Duong', 'Bac Giang', 'Binh Phuoc', 'Hung Yen', 'Quang
Nam', 'Thai Nguyen', 'Ha Nam', 'Ninh Thuan', 'Quang Ninh', 'Ben Tre', 'Vinh
Phuc', 'Bac Lieu', 'Quang Ngai', 'Thanh Hoa', 'Kien Giang', 'Da Nang', 'Nam
Dinh', 'Tien Giang', 'Hoa Binh', 'Ninh Binh', 'Vinh Long', 'Phu Tho', 'Binh
Dinh', 'Tra Vinh', 'Ha Tinh', 'Khanh Hoa', 'Soc Trang', 'Thai Binh', 'Dak Nong',
'Ca Mau', 'Can Tho', 'Quang Binh', 'Dak Lak', 'Tuyen Quang', 'Nghe An', 'Binh
```

```
Thuan', 'Phu Yen', 'Lang Son', 'Kon Tum', 'Lam Dong', 'Yen Bai', 'Dong Thap',
'Hau Giang', 'An Giang', 'Son La', 'Lao Cai', 'Quang Tri', 'Ha Giang', 'Bac
Kan', 'Cao Bang', 'Dien Bien', 'Lai Chau', 'Gia Lai', 'Ha Noi', 'TP. Ho Chi
Minh', 'Binh Duong', 'Dong Nai', 'Bac Ninh', 'Hai Phong', 'Tay Ninh', 'Bac
Giang', 'Ba Ria - Vung Tau', 'Ha Nam', 'Long An', 'Hai Duong', 'Thai Nguyen',
'Vinh Phuc', 'Da Nang', 'Hung Yen', 'Binh Phuoc', 'Tien Giang', 'Thanh Hoa',
'Phu Tho', 'Thua Thien Hue', 'Nghe An', 'Quang Ninh', 'Phu Yen', 'Khanh Hoa',
'Quang Nam', 'Binh Thuan', 'Vinh Long', 'Ninh Binh', 'Quang Ngai', 'Ninh Thuan',
'Bac Lieu', 'Soc Trang', 'Tra Vinh', 'Binh Dinh', 'Ca Mau', 'Hau Giang', 'Can
Tho', 'Thai Binh', 'An Giang', 'Nam Dinh', 'Ben Tre', 'Ha Tinh', 'Lam Dong',
'Kien Giang', 'Quang Tri', 'Tuyen Quang', 'Dong Thap', 'Yen Bai', 'Kon Tum',
'Bac Kan', 'Dak Lak', 'Lang Son', 'Dien Bien', 'Lao Cai', 'Quang Binh', 'Ha
Giang', 'Son La', 'Dak Nong', 'Cao Bang', 'Lai Chau', 'Hoa Binh', 'Gia Lai',
'TP. Ho Chi Minh', 'Bac Lieu', 'Ha Noi', 'Ba Ria - Vung Tau', 'Binh Duong', 'Hai
Phong', 'Dong Nai', 'Bac Ninh', 'Bac Giang', 'Long An', 'Ha Nam', 'Ben Tre',
'Tay Ninh', 'Vinh Phuc', 'Hai Duong', 'Quang Ninh', 'Hung Yen', 'Thai Nguyen',
'Binh Phuoc', 'Thanh Hoa', 'Phu Tho', 'Quang Binh', 'Vinh Long', 'Da Nang',
'Nghe An', 'Tien Giang', 'Dak Nong', 'Quang Ngai', 'Tra Vinh', 'Ninh Binh',
'Binh Thuan', 'Nam Dinh', 'Thai Binh', 'Khanh Hoa', 'Soc Trang', 'Binh Dinh',
'Can Tho', 'Thua Thien Hue', 'Ca Mau', 'Hoa Binh', 'Quang Tri', 'Lam Dong',
'Dong Thap', 'Ha Tinh', 'Kien Giang', 'Tuyen Quang', 'Gia Lai', 'Yen Bai', 'Lao
Cai', 'An Giang', 'Hau Giang', 'Dak Lak', 'Son La', 'Phu Yen', 'Kon Tum', 'Dien
Bien', 'Cao Bang', 'Quang Nam', 'Ninh Thuan', 'Bac Kan', 'Ha Giang', 'Lai Chau',
'Lao Cai', 'Hai Phong', 'Long An', 'TP. Ho Chi Minh', 'Binh Duong', 'Bac Ninh',
'Ha Noi', 'Dong Nai', 'Can Tho', 'Bac Giang', 'Quang Ninh', 'Tay Ninh', 'Vinh
Phuc', 'Hung Yen', 'Phu Tho', 'Thai Binh', 'Dak Lak', 'Binh Phuoc', 'Ba Ria -
Vung Tau', 'Hai Duong', 'Nghe An', 'Ha Nam', 'Thai Nguyen', 'Thanh Hoa', 'Thua
Thien Hue', 'Da Nang', 'Ninh Binh', 'Hau Giang', 'Kon Tum', 'Nam Dinh', 'Tien
Giang', 'Binh Dinh', 'Quang Tri', 'Ninh Thuan', 'Quang Binh', 'Yen Bai', 'Vinh
Long', 'Dong Thap', 'Quang Ngai', 'Quang Nam', 'Khanh Hoa', 'Ca Mau', 'Binh
Thuan', 'Lam Dong', 'Tra Vinh', 'An Giang', 'Ha Tinh', 'Lao Cai', 'Dak Nong',
'Kien Giang', 'Lang Son', 'Gia Lai', 'Phu Yen', 'Cao Bang', 'Lai Chau', 'Soc
Trang', 'Bac Lieu', 'Hoa Binh', 'Tuyen Quang', 'Ben Tre', 'Bac Kan', 'Dien
Bien', 'Ha Giang', 'Son La', 'TP. Ho Chi Minh', 'Binh Duong', 'Quang Ninh', 'Bac
Ninh', 'Hai Phong', 'Ha Noi', 'Thai Nguyen', 'Dong Nai', 'Bac Giang', 'Ba Ria -
Vung Tau', 'Nghe An', 'Long An', 'Hung Yen', 'Phu Tho', 'Tay Ninh', 'Ha Nam',
'Hai Duong', 'Thai Binh', 'Ha Tinh', 'Vinh Phuc', 'Binh Phuoc', 'Tien Giang',
'Thua Thien Hue', 'Can Tho', 'Vinh Long', 'Da Nang', 'Soc Trang', 'Thanh Hoa',
'Ninh Thuan', 'Quang Ngai', 'Quang Nam', 'Ninh Binh', 'Nam Dinh', 'Binh Dinh',
'Binh Thuan', 'An Giang', 'Dak Lak', 'Yen Bai', 'Khanh Hoa', 'Kien Giang', 'Tra
Vinh', 'Ben Tre', 'Phu Yen', 'Lang Son', 'Hoa Binh', 'Tuyen Quang', 'Dak Nong',
'Kon Tum', 'Quang Tri', 'Hau Giang', 'Ca Mau', 'Gia Lai', 'Lam Dong', 'Bac
Lieu', 'Bac Kan', 'Cao Bang', 'Dien Bien', 'Dong Thap', 'Ha Giang', 'Lai Chau',
'Lao Cai', 'Quang Binh', 'Son La']
```

[185]:
```
## Drop missing value fill with 0
### Number of new projects
```

```
n_df['Number of new projects'] = n_df['Number of new projects'].fillna(0)
### Newly registered capital (million USD)
n_df['Newly registered capital (million USD)'] = n_df['Newly registered capital␣
 ↪(million USD)'].fillna(0)
### Adjusted project number
n_df['Adjusted project number'] = n_df['Adjusted project number'].fillna(0)
### Adjusted capital (million USD)
n_df['Adjusted capital (million USD)'] = n_df['Adjusted capital (million USD)'].
 ↪fillna(0)
### Number of times of capital contribution to buy shares
n_df['Number of times of capital contribution to buy shares'] = n_df['Number of␣
 ↪times of capital contribution to buy shares'].fillna(0)
### Value of capital contribution, share purchase\n(million USD)
n_df['Value of capital contribution, share purchase\n(million USD)'] =␣
 ↪n_df['Value of capital contribution, share purchase\n(million USD)'].
 ↪fillna(0)
```

[186]:
```
n_df.sample(n=10)
```

[186]:

	Provinces	Number of new projects	Newly registered capital (million USD)	Adjusted project number	Adjusted capital (million USD)	Number of times of capital contribution to buy shares
217	Ninh Binh	7	120.01	3	23.04	10
171	Binh Thuan	0	0	1	0.41	7
5	Bac Giang	53	937.51	20	71.06	10
112	An Giang	1	0.2	0	0	
432	Bac Kan	0	0	0	0	
254	Ha Noi	496	711.81	158	1261.91	
277	Tien Giang	9	108.19	11	55.28	
181	Lao Cai	0	0	0	0	
396	Ha Tinh	1	275	0	0	
101	Dak Lak	2	49.5	1	9	

```
112                                                    4
432                                                    0
254                                                  751
277                                                    6
181                                                    2
396                                                    2
101                                                    2

    Value of capital contribution, share purchase\n(million USD)  Year
217                                               4.92            2019
171                                              18.71            2018
5                                                 1.46            2016
112                                               7.73            2017
432                                                  0            2022
254                                            1611.82            2020
277                                               3.69            2020
181                                               0.84            2018
396                                               1.13            2022
101                                               1.75            2017
```

[187]:
```
## Data consistency
cols_to_convert = n_df.columns.drop(['Provinces', 'Year'])
n_df[cols_to_convert] = n_df[cols_to_convert].apply(pd.to_numeric,␣
 ↪errors='coerce')
## Check for missing value
print(n_df.isnull().values.any())
print(n_df.isna().sum())
```

```
False
Provinces                                                          0
Number of new projects                                             0
Newly registered capital (million USD)                             0
Adjusted project number                                            0
Adjusted capital (million USD)                                     0
Number of times of capital contribution to buy shares              0
Value of capital contribution, share purchase\n(million USD)       0
Year                                                               0
dtype: int64
```

0.4 Visualiztion

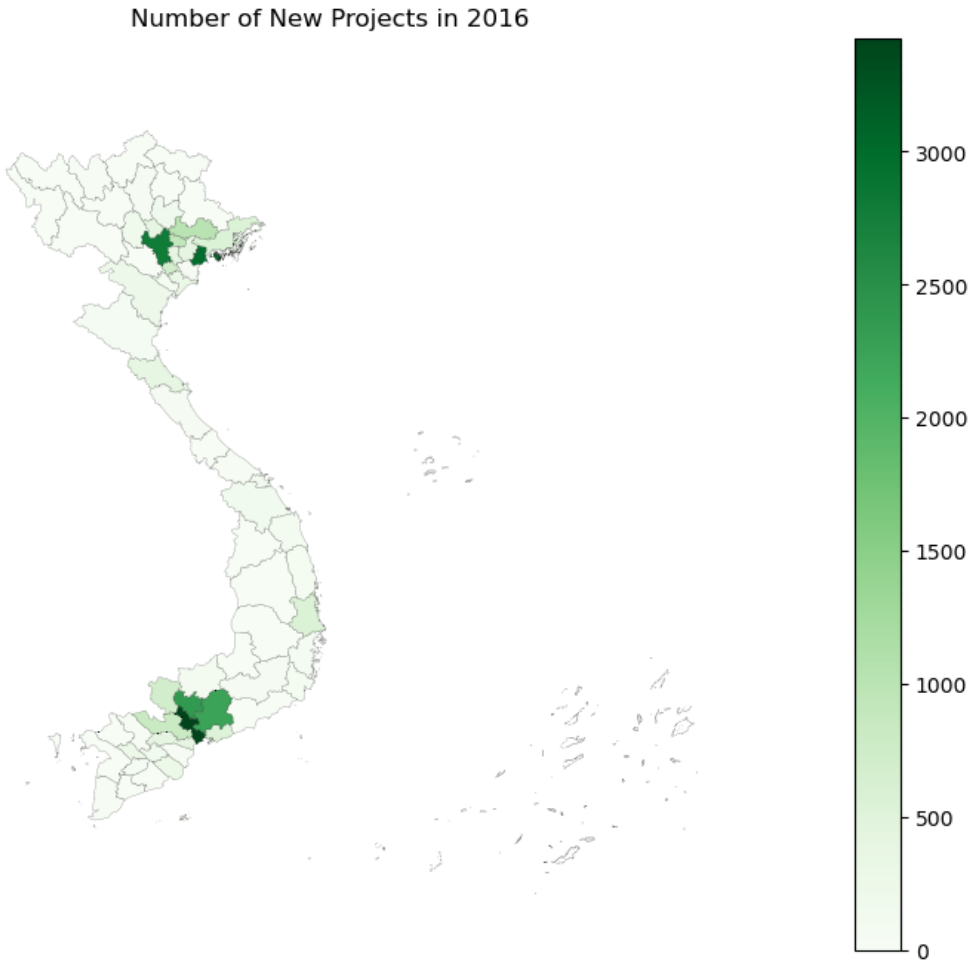
[188]:
```
# Create a new column for the total FDI
n_df['Total FDI'] = n_df['Newly registered capital (million USD)'] +␣
 ↪n_df['Adjusted capital (million USD)'] + n_df['Value of capital␣
 ↪contribution, share purchase\n(million USD)']
# Merge data with geodata
fullData = data.merge(
```

```
    n_df,
    left_on=['Name'],  # identifier from geodataframe
    right_on=['Provinces']  # identifier from dataframe
)
```

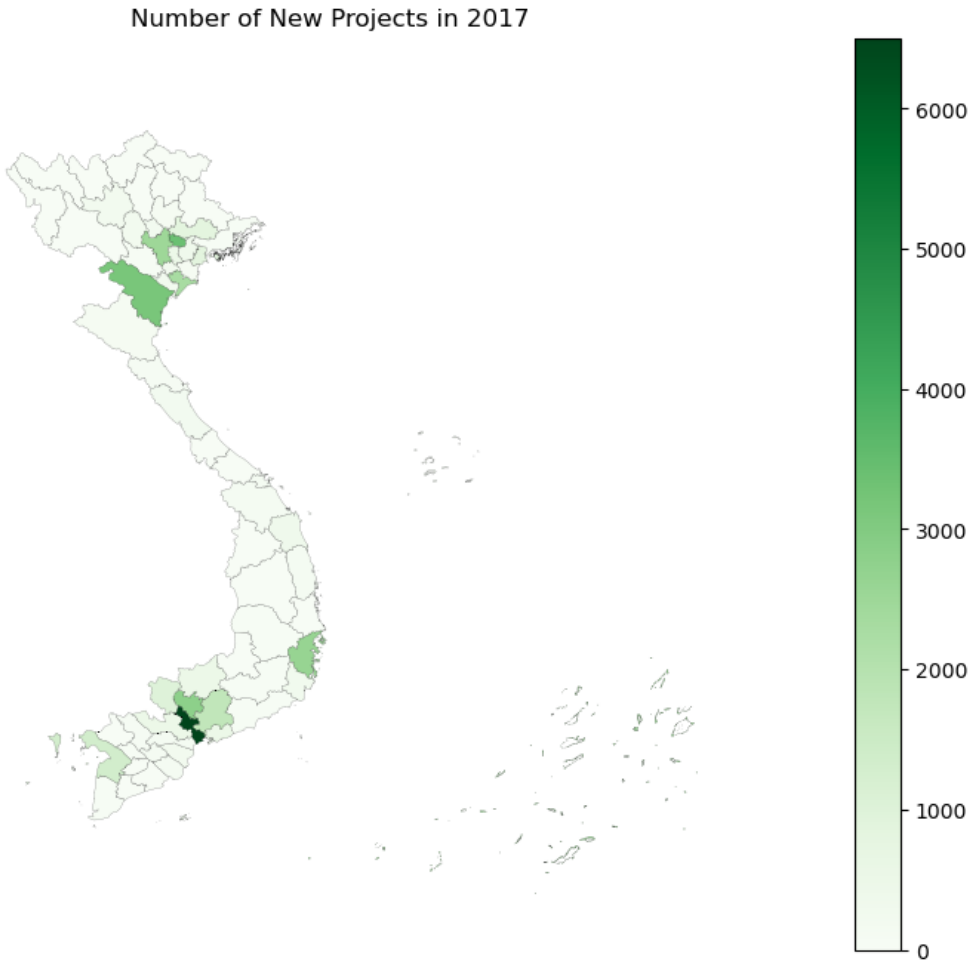
0.4.1 Plot map chart of dataset

[189]:
```
years = fullData['Year'].unique()
fullData['Total FDI'] = pd.to_numeric(fullData['Total FDI'], errors='coerce')
for year in years:
    # Filter data for each year
    data_year = fullData[fullData['Year'] == year]
    # Plot with geoplot for each year
    plt.figure(figsize=(12, 8))
    geoplot.choropleth(
        data_year,
        projection=gcrs.AlbersEqualArea(),
        hue="Total FDI",
        cmap='Greens',
        linewidth=0.1,
        edgecolor='black',
        legend=True,
        figsize=(12, 8)
    )
    plt.title(f"Number of New Projects in {year}")
    plt.show()
```

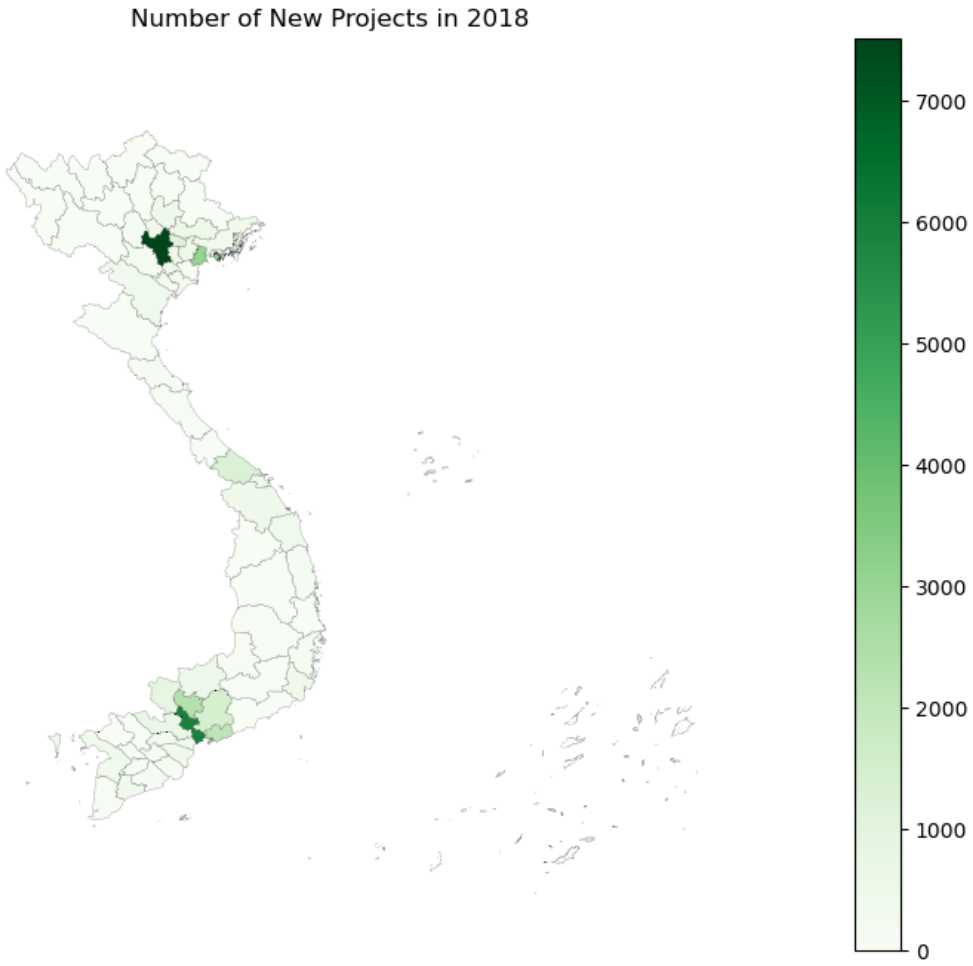
```
<Figure size 1200x800 with 0 Axes>
```



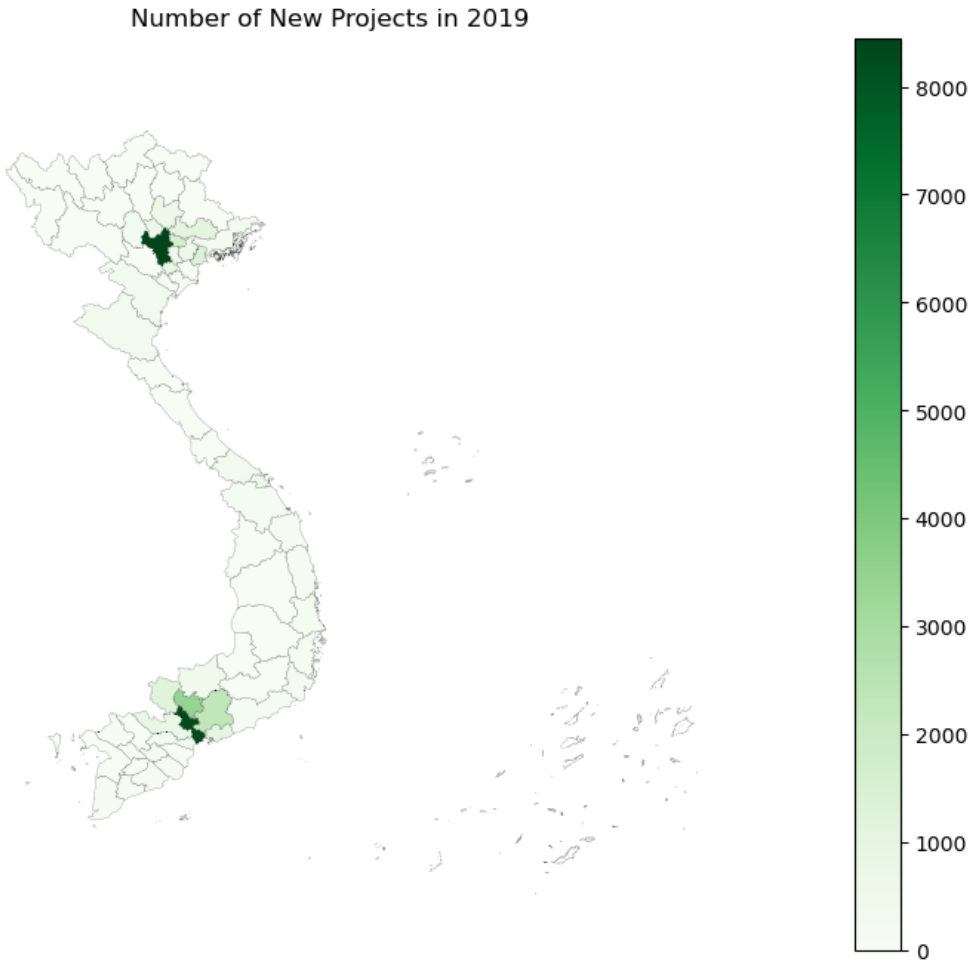

```
<Figure size 1200x800 with 0 Axes>
```



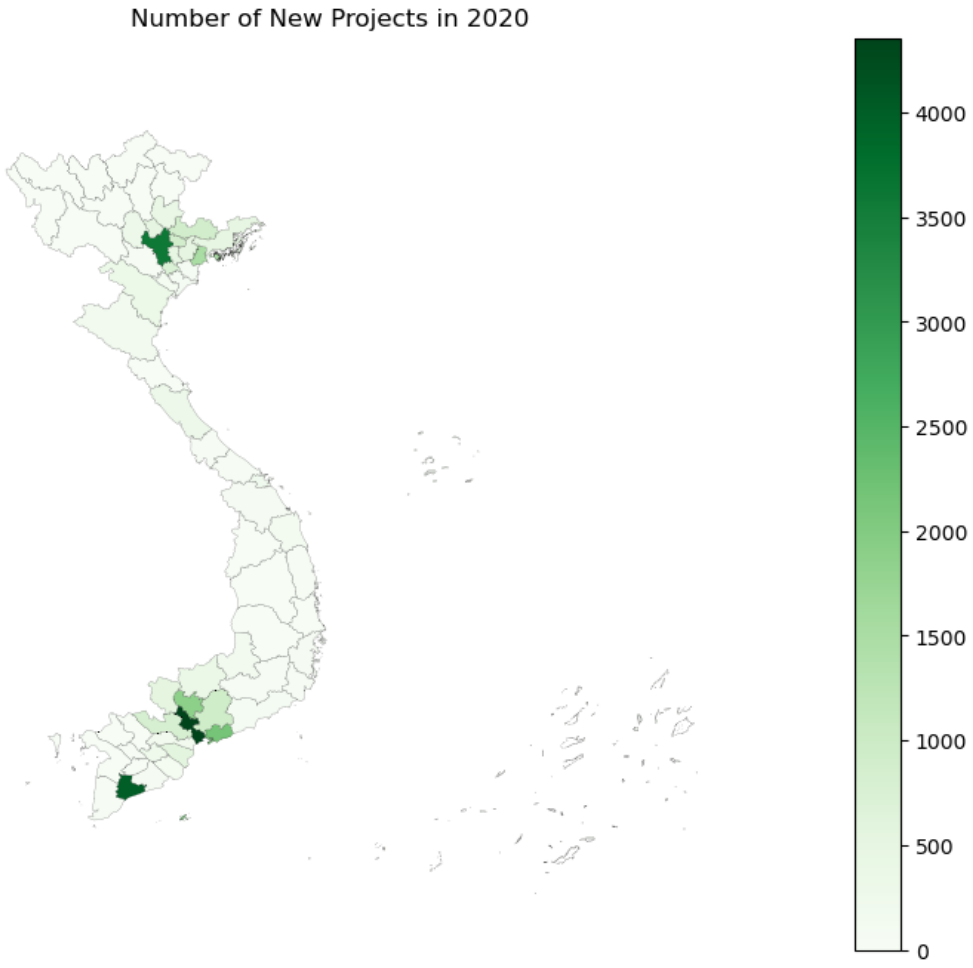

```
<Figure size 1200x800 with 0 Axes>
```



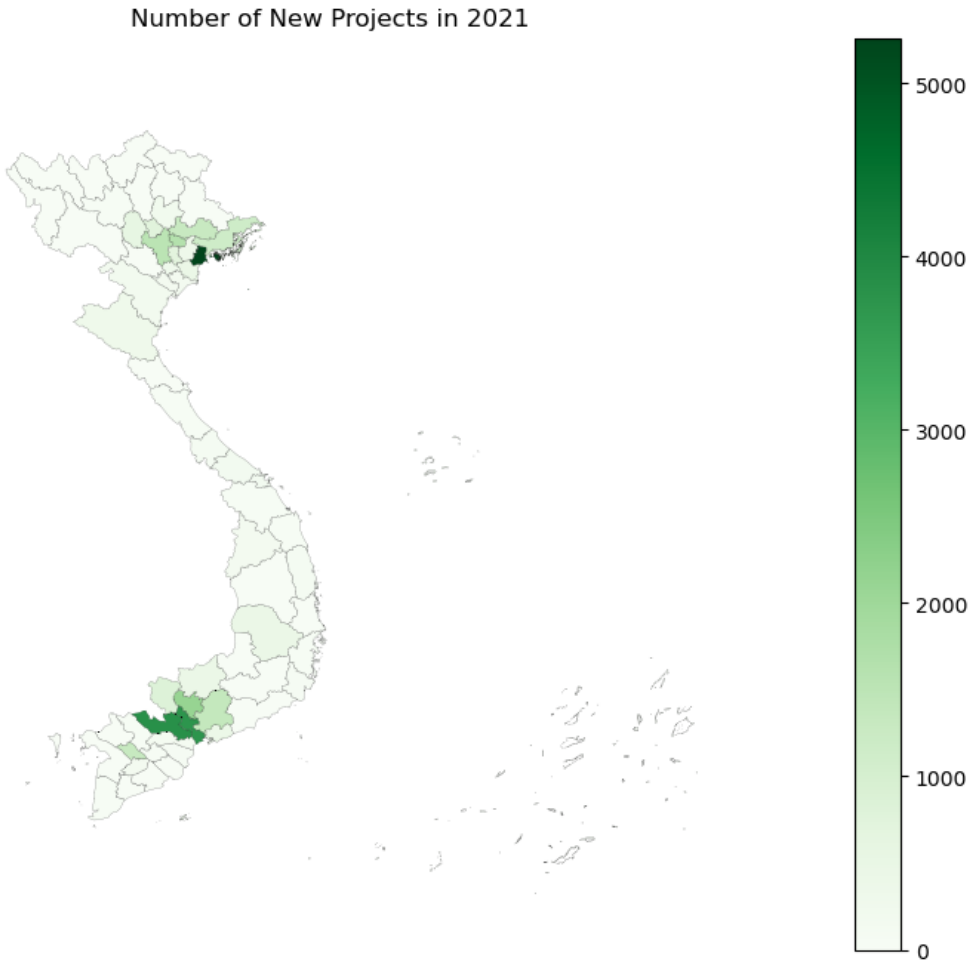

```
<Figure size 1200x800 with 0 Axes>
```



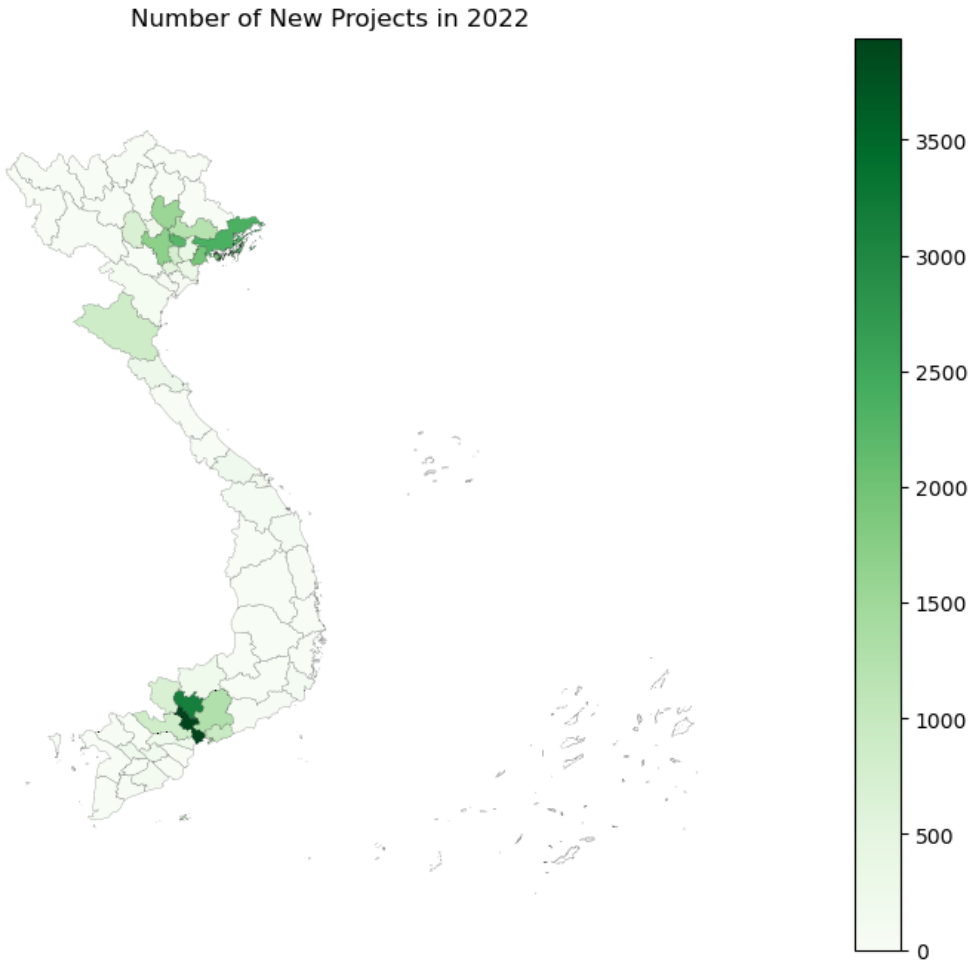
<Figure size 1200x800 with 0 Axes>



<Figure size 1200x800 with 0 Axes>



<Figure size 1200x800 with 0 Axes>



Note: From the chart above, we can see that foreign investment in Vietnam is concentrated in major cities such as Ho Chi Minh City, Hanoi, Hai Phong,etc. However, a positive sign is that there is also investment spread across various provinces.

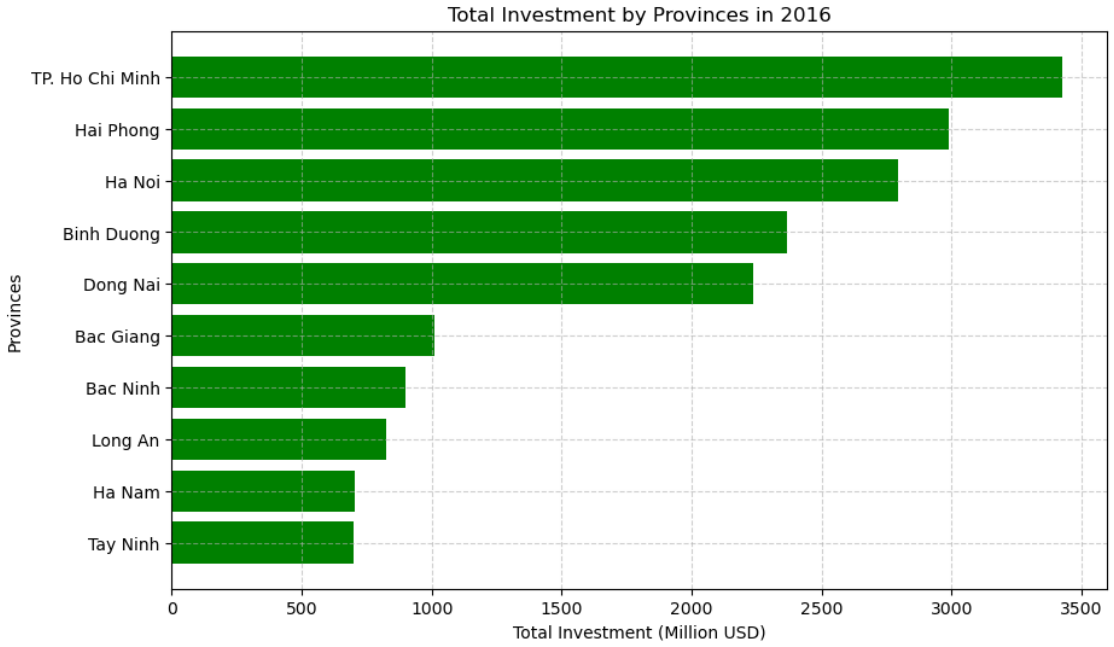
0.4.2 Detail with buble chart

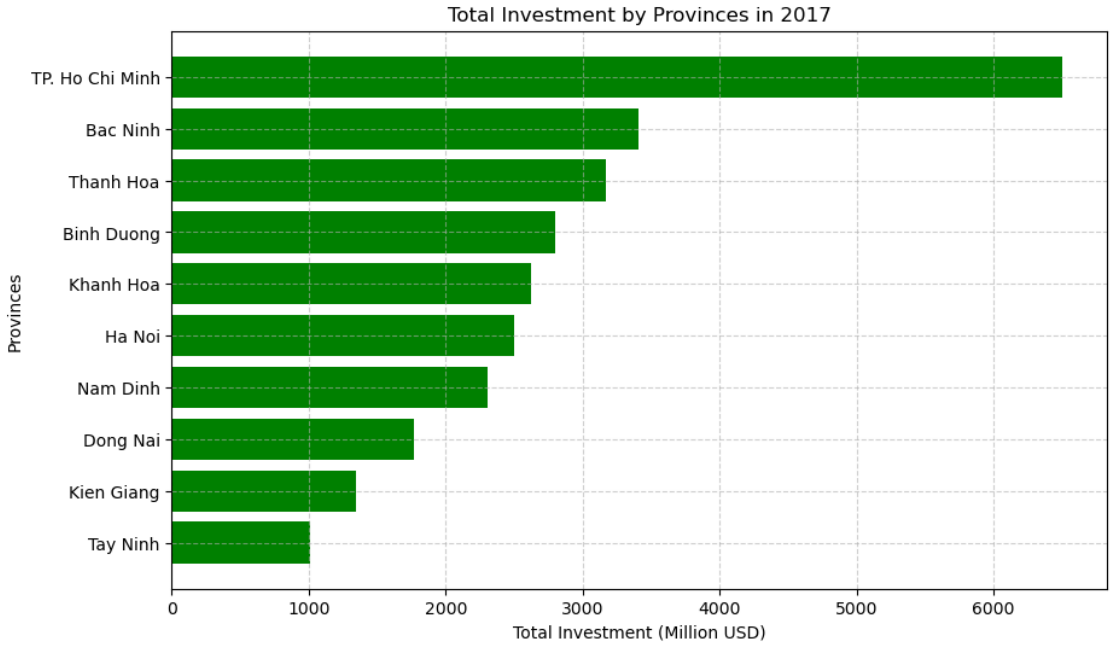
[190]:
```
years = n_df['Year'].unique()
for year in years:
    # Filter data for each year
    data_year = n_df[n_df['Year'] == year]
    # Plot with Plotly Express
    fig = px.scatter(data_year, x='Provinces', y='Total FDI', size='Total␣
 ↪FDI',color='Provinces',
                     title=f'Total FDI by Provinces in {year}',
                     labels={'Total FDI': 'Total Investment (Million USD)',␣
 ↪'Provinces': 'Provinces'},
                     size_max=60)
```

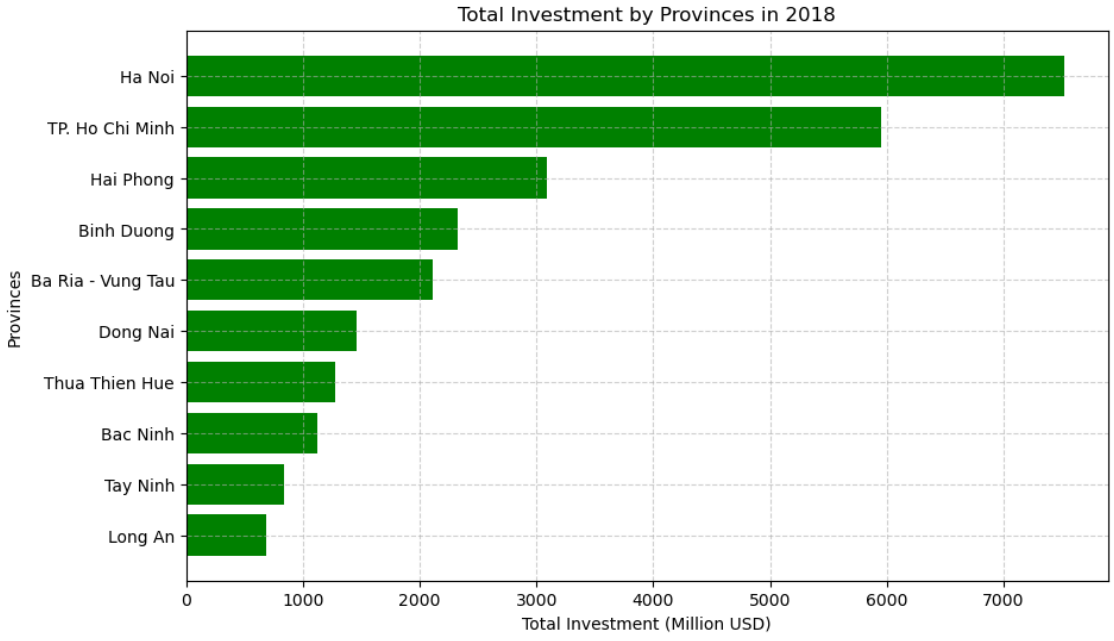
```
    fig.update_layout(yaxis_title='Total Investment (Million USD)',
                      xaxis_title='Provinces',
                      title=f'Total FDI by Provinces in {year}')
    fig.show()
```

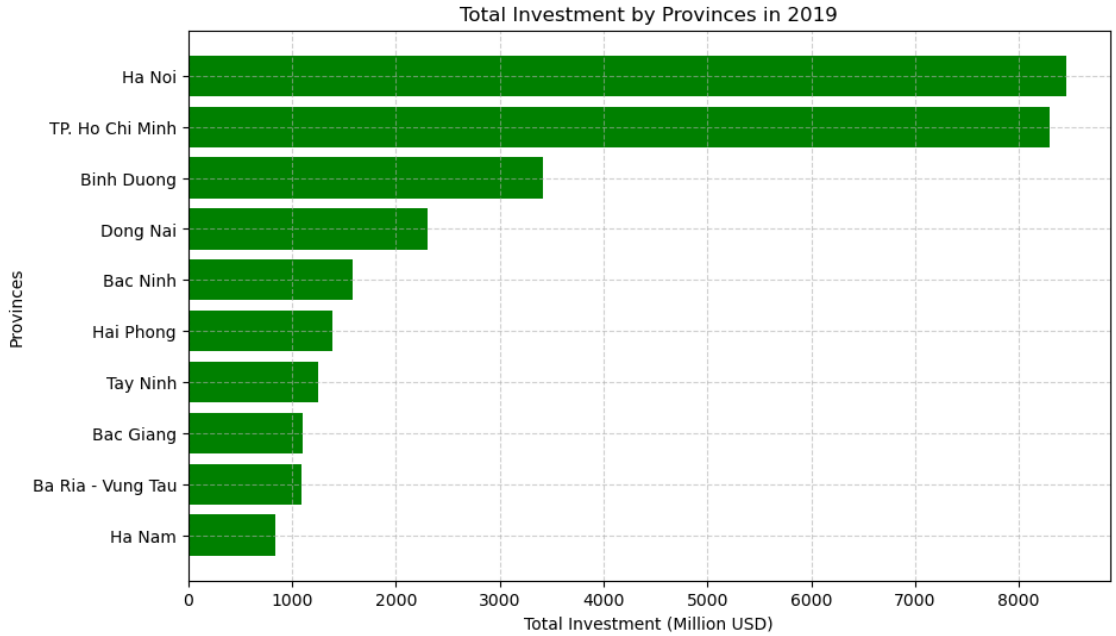
0.4.3 Top 10 provinces with the largest total FDI

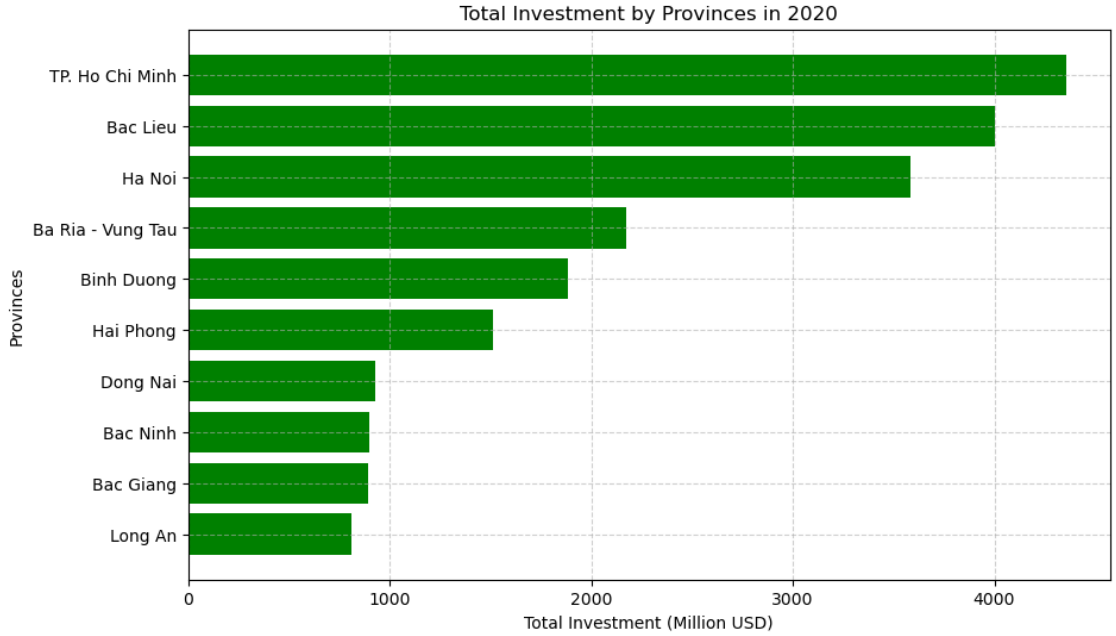
[191]:
```
years = n_df['Year'].unique()
# Loop through each year and plot the total investment by industry
for year in years:
    # Filter data by year
    df_year = n_df[n_df['Year'] == year]
    # Sort the data by Total FDI
    df_year.sort_values('Total FDI', ascending=False, inplace=True)
    # Select the top 10 provinces with the highest Total FDI
    df_year_top10 = df_year.head(10)
    # Set axis values
    x = df_year_top10['Provinces'].values
    y = df_year_top10['Total FDI'].values
    # Plot
    plt.figure(figsize=(10, 6))
    plt.barh(x, y, color='Green')
    plt.xlabel('Total Investment (Million USD)')
    plt.ylabel('Provinces')
    plt.title(f'Total Investment by Provinces in {year}')
    plt.gca().invert_yaxis()
    plt.grid(True, linestyle='--', alpha=0.6)
    plt.show()
```

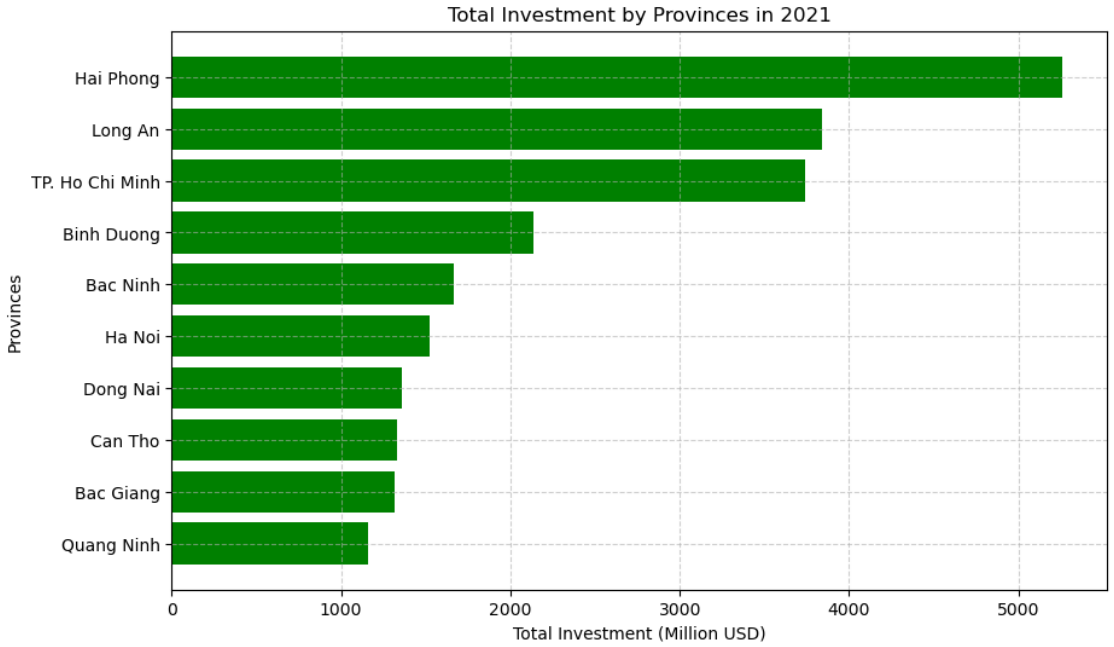


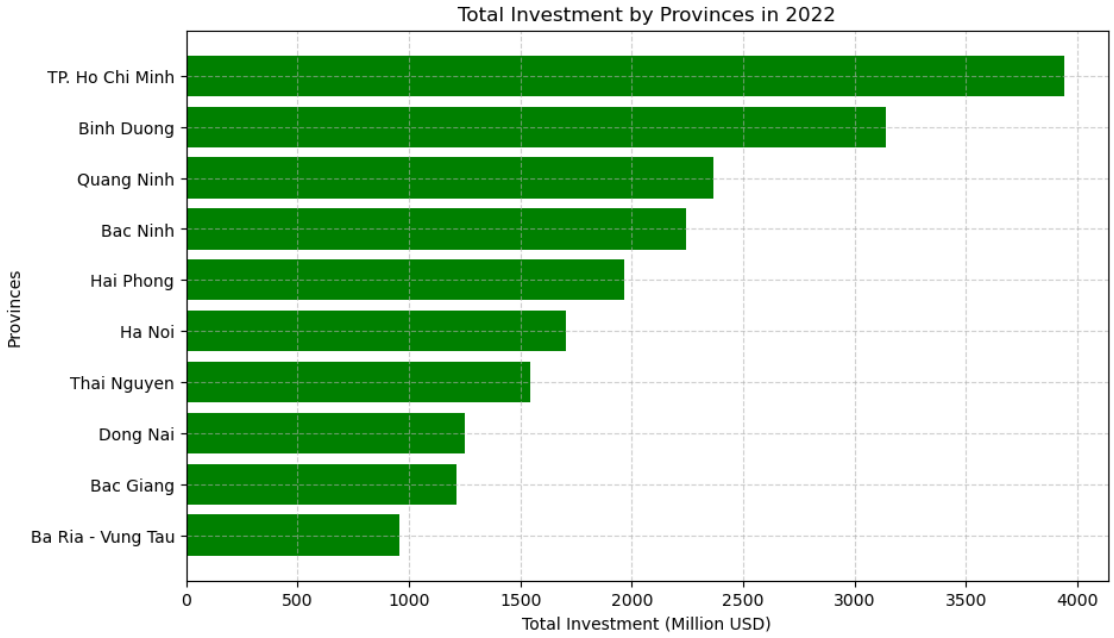

Total Investment by Provinces in 2017
TP. Ho Chi Minh
Bac Ninh
Thanh Hoa
Binh Duong
Khanh Hoa
Ha Noi
Nam Dinh
Dong Nai
Kien Giang
Tay Ninh
Provinces
0
1000
2000
3000
4000
5000
6000
Total Investment (Million USD)


Total Investment by Provinces in 2018
Ha Noi
TP. Ho Chi Minh
Hai Phong
Binh Duong
Ba Ria - Vung Tau
Dong Nai
Thua Thien Hue
Bac Ninh
Tay Ninh
Long An
Provinces
0
1000
2000
3000
4000
5000
6000
7000
Total Investment (Million USD)


Total Investment by Provinces in 2019
Ha Noi
TP. Ho Chi Minh
Binh Duong
Dong Nai
Bac Ninh
Hai Phong
Tay Ninh
Bac Giang
Ba Ria - Vung Tau
Ha Nam
Provinces
0
1000
2000
3000
4000
5000
6000
7000
8000
Total Investment (Million USD)


Total Investment by Provinces in 2020
TP. Ho Chi Minh
Bac Lieu
Ha Noi
Ba Ria - Vung Tau
Binh Duong
Hai Phong
Dong Nai
Bac Ninh
Bac Giang
Long An
Provinces
0
1000
2000
3000
4000
Total Investment (Million USD)





From the ranking plot and multiple barh , we can have below observations: - There is still volatility and changes in the rankings: this indicates the diversity of investment sectors, opportunities, and potential existing across different provinces in Vietnam. - There are `new names` appearing in some years: this suggests that the development is on a significant upward trend and continues to attract investors.