In [ ]:

```python
# Library cell
import pandas as pd
import regex as re
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import panel as pn
#to ignore warnings
import warnings
warnings.filterwarnings('ignore')
from sklearn.linear_model import LinearRegression
```

In [ ]:

```python
# 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
```

Load data

In [ ]:

```python
file_path = r'D:\Repo-train\Jnotebook\FDI_Analytics\dataset\fdi_industry_en.csv'
df = pd.read_csv(file_path)
```

In [ ]:

```python
df.head()
```

Out[ ]:

	Order	Industry	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	Value of capital contribution, share purchase\n(million USD)	Year
0	1	Manufacturing and processing industry	1020	9812.57	861	5132.55	290	593.51	2016
1	2	Wholesale and retail; repair cars, motorbikes,...	505	367.04	99	320.72	1269	1211.45	2016
2	3	Real estate business	59	1522.67	12	-559.05	80	722.55	2016
3	4	Professional activities, science and technology	282	436.45	65	316.95	212	179.68	2016
4	5	Warehousing transportation	88	703.94	22	-29	119	207.19	2016

In [ ]:

```python
df.tail()
```

Out[ ]:

	Order	Industry	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	Value of capital contribution, share purchase\n(million USD)	Year
126	127	Extractive	1	2	-	-	3	17.09	2022
127	128	Accommodation and food services	33	8	18	-59.82	240	63.71	2022
128	129	Other service activities	2	0	4	3.37	17	2.24	2022
129	130	Art, play and entertainment	1	0	1	0.15	11	3.5	2022
130	131	Employment activities in households	-	-	-	-	1	0.55	2022

Analyze the data

In [ ]:

```
# 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')
```

```
(131, 8)
 ----------------
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 131 entries, 0 to 130
Data columns (total 8 columns):
 #   Column                                                       Non-Null Count  Dtype 
---  ------                                                       --------------  ----- 
 0   Industry                                                     131 non-null    object
 1   Number of new projects                                       131 non-null    object
 2   Newly registered capital (million USD)                       131 non-null    object
 3   Adjusted project number                                      127 non-null    object
 4   Adjusted capital (million USD)                               127 non-null    object
 5   Number of times of capital contribution to buy shares        124 non-null    object
 6   Value of capital contribution, share purchase
(million USD)  124 non-null    object
 7   Year                                                         131 non-null    int64 
dtypes: int64(1), object(7)
memory usage: 8.3+ KB
None
 ----------------
Industry                                                         19
Number of new projects                                           84
Newly registered capital (million USD)                          127
Adjusted project number                                          56
Adjusted capital (million USD)                                  123
Number of times of capital contribution to buy shares            99
Value of capital contribution, share purchase\n(million USD)    123
Year                                                              7
dtype: int64
 ----------------
Industry                                                        False
Number of new projects                                          False
Newly registered capital (million USD)                          False
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
 ----------------
Industry                                                        0
Number of new projects                                          0
Newly registered capital (million USD)                          0
Adjusted project number                                         4
Adjusted capital (million USD)                                  4
Number of times of capital contribution to buy shares           7
Value of capital contribution, share purchase\n(million USD)    7
Year                                                            0
dtype: int64
 ----------------
```

Observations

- The shape of dataset `fdi_industry_en.csv` is 131 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 duplicates to get total number of `Industr`, `Year`
- Check all columns to get boolean values indicating if missing values exist and determine which columns have missing values

`Adjusted project number` => `Adjusted capital (million USD)`

`Number of times of capital contribution to buy shares` => `Value of capital contribution, share purchase\n(million USD)`)

Data Cleaning

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 `Industry` 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

In [ ]:
```
## 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 'Industry' have values not numeric:
['Manufacturing and processing industry', 'Wholesale and retail; repair cars, motorbikes, motorbikes', 'Real estate busines
s', 'Professional activities, science and technology', 'Warehousing transportation', 'Construction', 'Financial, banking and
insurance activities', 'Water supply and waste treatment', 'Accommodation and food services', 'Information and communicatio
n', 'Art, play and entertainment', 'Administrative activities and support services', 'Producing and distributing electricity,
gas, water, air conditioning', 'Agriculture, forestry and fisheries', 'Extractive', 'Other service activities', 'Education an
d training', 'Health and social assistance activities', 'Employment activities in households', 'Manufacturing and processing
industry', 'Producing and distributing electricity, gas, water, air conditioning', 'Real estate business', 'Wholesale and ret
ail; repair cars, motorbikes, motorbikes', 'Extractive', 'Construction', 'Professional activities, science and technology',
'Water supply and waste treatment', 'Accommodation and food services', 'Health and social assistance activities', 'Warehousin
g transportation', 'Information and communication', 'Agriculture, forestry and fisheries', 'Education and training', 'Adminis
trative activities and support services', 'Other service activities', 'Financial, banking and insurance activities', 'Art, pl
ay and entertainment', 'Employment activities in households', 'Manufacturing and processing industry', 'Real estate busines
s', 'Wholesale and retail; repair cars, motorbikes, motorbikes', 'Professional activities, science and technology', 'Producin
g and distributing electricity, gas, water, air conditioning', 'Construction', 'Art, play and entertainment', 'Accommodation
and food services', 'Information and communication', 'Warehousing transportation', 'Water supply and waste treatment', 'Admin
istrative activities and support services', 'Agriculture, forestry and fisheries', 'Health and social assistance activities',
'Education and training', 'Financial, banking and insurance activities', 'Extractive', 'Other service activities', 'Manufactu
ring and processing industry', 'Real estate business', 'Wholesale and retail; repair cars, motorbikes, motorbikes', 'Professi
onal activities, science and technology', 'Financial, banking and insurance activities', 'Producing and distributing electric
ity, gas, water, air conditioning', 'Construction', 'Information and communication', 'Accommodation and food services', 'Ware
housing transportation', 'Water supply and waste treatment', 'Health and social assistance activities', 'Administrative activ
ities and support services', 'Agriculture, forestry and fisheries', 'Education and training', 'Art, play and entertainment',
'Other service activities', 'Extractive', 'Employment activities in households', 'Manufacturing and processing industry', 'Pr
oducing and distributing electricity, gas, water, air conditioning', 'Real estate business', 'Wholesale and retail; repair ca
rs, motorbikes, motorbikes', 'Professional activities, science and technology', 'Warehousing transportation', 'Construction',
'Accommodation and food services', 'Financial, banking and insurance activities', 'Information and communication', 'Agricultu
re, forestry and fisheries', 'Education and training', 'Water supply and waste treatment', 'Other service activities', 'Admin
istrative activities and support services', 'Health and social assistance activities', 'Extractive', 'Art, play and entertain
ment', 'Employment activities in households', 'Manufacturing and processing industry', 'Producing and distributing electricit
y, gas, water, air conditioning', 'Real estate business', 'Wholesale and retail; repair cars, motorbikes, motorbikes', 'Profe
ssional activities, science and technology', 'Warehousing transportation', 'Construction', 'Information and communication',
'Accommodation and food services', 'Agriculture, forestry and fisheries', 'Water supply and waste treatment', 'Financial, ban
king and insurance activities', 'Education and training', 'Administrative activities and support services', 'Health and socia
l assistance activities', 'Other service activities', 'Art, play and entertainment', 'Extractive', 'Manufacturing and process
ing industry', 'Real estate business', 'Producing and distributing electricity, gas, water, air conditioning', 'Professional
activities, science and technology', 'Wholesale and retail; repair cars, motorbikes, motorbikes', 'Information and communicat
ion', 'Warehousing transportation', 'Education and training', 'Construction', 'Agriculture, forestry and fisheries', 'Adminis
trative activities and support services', 'Financial, banking and insurance activities', 'Water supply and waste treatment',
'Health and social assistance activities', 'Extractive', 'Accommodation and food services', 'Other service activities', 'Art,
play and entertainment', 'Employment activities in households']
Column 'Number of new projects' have values not numeric:
[' -   ', ' -   ']
Column 'Newly registered capital (million USD)' have values not numeric:
['6,860.36', '8,369.30', '2,238.93', '1,279.02', '9,067.46', '5,216.78', '1,631.33', '12,093.14', '1,817.97', '7,190.77', '5,
080.81', '7,251.98', '5,316.16', '1,390.03', ' -   ', '7,213', '1,816', '2,101', ' -   ']
Column 'Adjusted project number' have values not numeric:
[' -   ', ' -   ', ' -   ', ' -   ', ' -   ']
Column 'Adjusted capital (million USD)' have values not numeric:
['7,271.27', '5,093.78', '1,125.00', '5,381.98', '4,593.86', '1,256.08', ' -   ', ' -   ', '7,346.30', ' -   ', '7,977.90',
'1,059.28', ' -   ', ' -   ']
Column 'Number of times of capital contribution to buy shares' have values not numeric:
['1,365', '1,945', '1,528', '2,829', '2,261', '3,292', '1,129', '1,268', '2,264', ' -   ', '1,338', '1,417']
Column 'Value of capital contribution, share purchase
(million USD)' have values not numeric:
['1,744.36', '1,555.86', '2,426.80', '2,863.11', '1,820.00', '7,086.66', '2,751.79', '1,427.98', '1,091.52', '1,816.46', '1,9
41.46', '1,062.96', ' -   ', '3,522.60', '1,000.73', '1,611.06', '1,576.55']
```

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

In [ ]:
```
# 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.]
### 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, rege
### 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'
### 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
```

In [ ]:

```
## 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 'Industry' have values not numeric:
['Manufacturing and processing industry', 'Wholesale and retail; repair cars motorbikes motorbikes', 'Real estate business',
'Professional activities science and technology', 'Warehousing transportation', 'Construction', 'Financial banking and insura
nce activities', 'Water supply and waste treatment', 'Accommodation and food services', 'Information and communication', 'Art
play and entertainment', 'Administrative activities and support services', 'Producing and distributing electricity gas water
air conditioning', 'Agriculture forestry and fisheries', 'Extractive', 'Other service activities', 'Education and training',
'Health and social assistance activities', 'Employment activities in households', 'Manufacturing and processing industry', 'P
roducing and distributing electricity gas water air conditioning', 'Real estate business', 'Wholesale and retail; repair cars
motorbikes motorbikes', 'Extractive', 'Construction', 'Professional activities science and technology', 'Water supply and was
te treatment', 'Accommodation and food services', 'Health and social assistance activities', 'Warehousing transportation', 'I
nformation and communication', 'Agriculture forestry and fisheries', 'Education and training', 'Administrative activities and
support services', 'Other service activities', 'Financial banking and insurance activities', 'Art play and entertainment', 'E
mployment activities in households', 'Manufacturing and processing industry', 'Real estate business', 'Wholesale and retail;
repair cars motorbikes motorbikes', 'Professional activities science and technology', 'Producing and distributing electricity
gas water air conditioning', 'Construction', 'Art play and entertainment', 'Accommodation and food services', 'Information an
d communication', 'Warehousing transportation', 'Water supply and waste treatment', 'Administrative activities and support se
rvices', 'Agriculture forestry and fisheries', 'Health and social assistance activities', 'Education and training', 'Financia
l banking and insurance activities', 'Extractive', 'Other service activities', 'Manufacturing and processing industry', 'Real
estate business', 'Wholesale and retail; repair cars motorbikes motorbikes', 'Professional activities science and technolog
y', 'Financial banking and insurance activities', 'Producing and distributing electricity gas water air conditioning', 'Const
ruction', 'Information and communication', 'Accommodation and food services', 'Warehousing transportation', 'Water supply and
waste treatment', 'Health and social assistance activities', 'Administrative activities and support services', 'Agriculture f
orestry and fisheries', 'Education and training', 'Art play and entertainment', 'Other service activities', 'Extractive', 'Em
ployment activities in households', 'Manufacturing and processing industry', 'Producing and distributing electricity gas wate
r air conditioning', 'Real estate business', 'Wholesale and retail; repair cars motorbikes motorbikes', 'Professional activit
ies science and technology', 'Warehousing transportation', 'Construction', 'Accommodation and food services', 'Financial bank
ing and insurance activities', 'Information and communication', 'Agriculture forestry and fisheries', 'Education and trainin
g', 'Water supply and waste treatment', 'Other service activities', 'Administrative activities and support services', 'Health
and social assistance activities', 'Extractive', 'Art play and entertainment', 'Employment activities in households', 'Manufa
cturing and processing industry', 'Producing and distributing electricity gas water air conditioning', 'Real estate busines
s', 'Wholesale and retail; repair cars motorbikes motorbikes', 'Professional activities science and technology', 'Warehousing
transportation', 'Construction', 'Information and communication', 'Accommodation and food services', 'Agriculture forestry an
d fisheries', 'Water supply and waste treatment', 'Financial banking and insurance activities', 'Education and training', 'Ad
ministrative activities and support services', 'Health and social assistance activities', 'Other service activities', 'Art pl
ay and entertainment', 'Extractive', 'Manufacturing and processing industry', 'Real estate business', 'Producing and distribu
ting electricity gas water air conditioning', 'Professional activities science and technology', 'Wholesale and retail; repair
cars motorbikes motorbikes', 'Information and communication', 'Warehousing transportation', 'Education and training', 'Constr
uction', 'Agriculture forestry and fisheries', 'Administrative activities and support services', 'Financial banking and insur
ance activities', 'Water supply and waste treatment', 'Health and social assistance activities', 'Extractive', 'Accommodation
and food services', 'Other service activities', 'Art play and entertainment', 'Employment activities in households']
```

In [ ]:

```
## Drop missing value fill with 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'
### 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
```

In [ ]:

```
n_df.sample(n=10)
```

Out[ ]:

	Industry	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	Value of capital contribution, share purchase\n(million USD)	Year
71	Art play and entertainment	6	8.27	1	2.65	26	51.81	2019
117	Information and communication	241	183	52	310.73	305	161.64	2022
15	Other service activities	5	55.76	7	9.77	6	2.23	2016
45	Accommodation and food services	102	27.36	21	59.82	311	491.34	2018
8	Accommodation and food services	97	278.14	11	58.02	135	70.53	2016
26	Water supply and waste treatment	12	566.7	2	1.3	0	0	2017
70	Education and training	71	24.96	12	9.36	172	30.3	2019
61	Producing and distributing electricity gas wat...	15	722.6	2	0	62	302.42	2019
41	Professional activities science and technology	386	183.37	88	144.05	584	1820.00	2018
126	Extractive	1	2	0	0	3	17.09	2022

In [ ]:

```
## Data consistency
cols_to_convert = n_df.columns.drop(['Industry', '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
Industry                                                        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
```

In [ ]:

```
# Summary statistics
n_df.describe().T
```

Out[ ]:

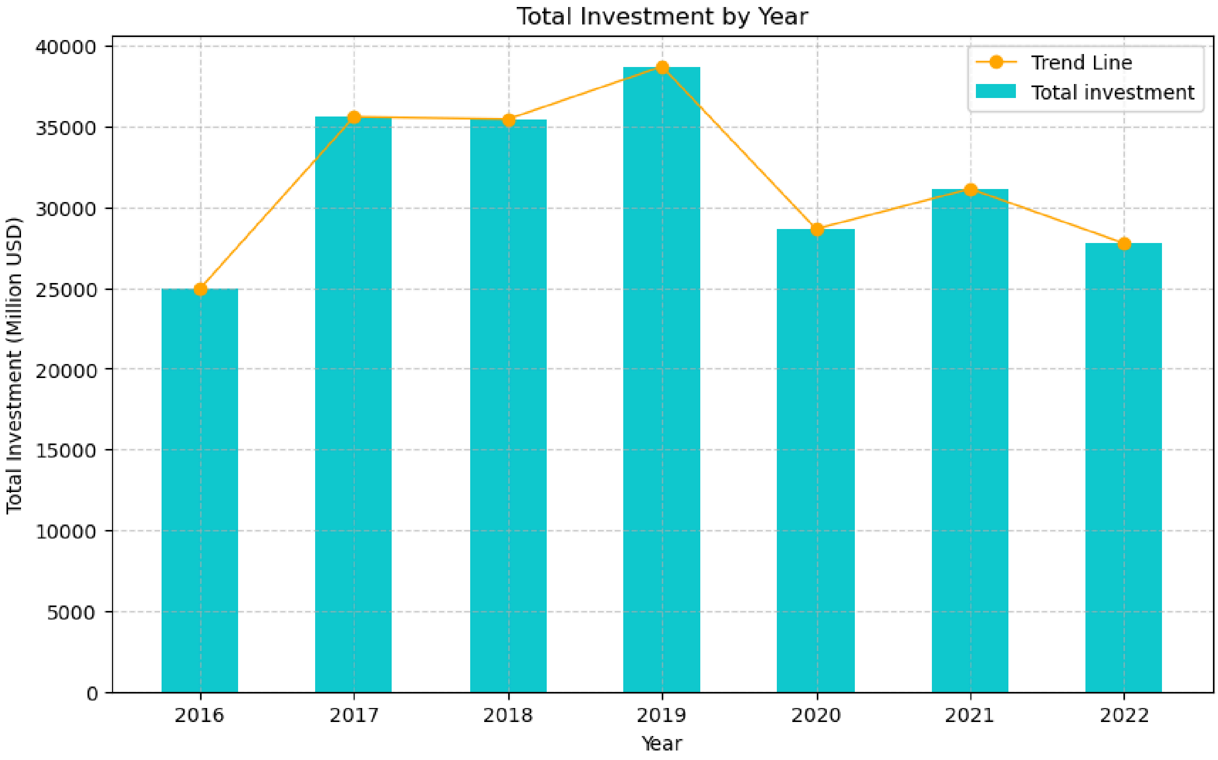
	count	mean	std	min	25%	50%	75%	max
Number of new projects	131.0	140.251908	254.032413	0.0	6.000	33.00	110.000	1314.00
Newly registered capital (million USD)	131.0	866.555344	2170.593023	0.0	13.405	94.00	377.775	12093.14
Adjusted project number	131.0	62.557252	166.067834	0.0	3.000	13.00	31.500	861.00
Adjusted capital (million USD)	131.0	417.013893	1403.382507	0.0	2.620	30.88	116.490	7977.90
Number of times of capital contribution to buy shares	131.0	285.427481	557.047188	0.0	12.500	68.00	250.500	3292.00
Value of capital contribution, share purchase\n(million USD)	131.0	413.856107	871.367984	0.0	11.615	74.03	397.190	7086.66
Year	131.0	2018.992366	2.013402	2016.0	2017.000	2019.00	2021.000	2022.00

Univariate Analysis

Total investment over the years

In [ ]:

```python
# Caculate the total investment in each row (add column 'Total investment')
n_df['Total investment'] = n_df['Newly registered capital (million USD)'] + n_df['Adjusted capital (million USD)'] + n_df['V
# Caculate the total investment in each year (group by year)
total_investment_by_year = n_df.groupby('Year')['Total investment'].sum().reset_index()
# plot the total investment by year
plt.figure(figsize=(10, 6))
plt.bar(total_investment_by_year['Year'], total_investment_by_year['Total investment'], color='#10c8ce',width= 0.5 ,  label=
plt.plot(total_investment_by_year['Year'], total_investment_by_year['Total investment'], color='orange', marker='o', linewid
plt.xlabel('Year')
plt.ylabel('Total Investment (Million USD)')
plt.title('Total Investment by Year')
plt.legend()
plt.grid(True, linestyle='--', alpha=0.6)
plt.show()
```



In [ ]:

```python
# Statistical analysis
total_investment_by_year['Change'] = total_investment_by_year['Total investment'].diff()
total_investment_by_year['Percentage Change'] = total_investment_by_year['Change']/total_investment_by_year['Total investmen
total_investment_by_year['Percentage Change'] = total_investment_by_year['Percentage Change'].round(2).astype(str) + '%'
total_investment_by_year
```

Out[ ]:

	Year	Total investment	Change	Percentage Change
0	2016	24960.96	NaN	nan%
1	2017	35605.15	10644.19	42.64%
2	2018	35469.20	-135.95	-0.38%
3	2019	38727.77	3258.57	9.19%
4	2020	28667.57	-10060.20	-25.98%
5	2021	31153.35	2485.78	8.67%
6	2022	27778.72	-3374.63	-10.83%

From the bar/line chart and `total_investment_by_year` df, we can infer the following

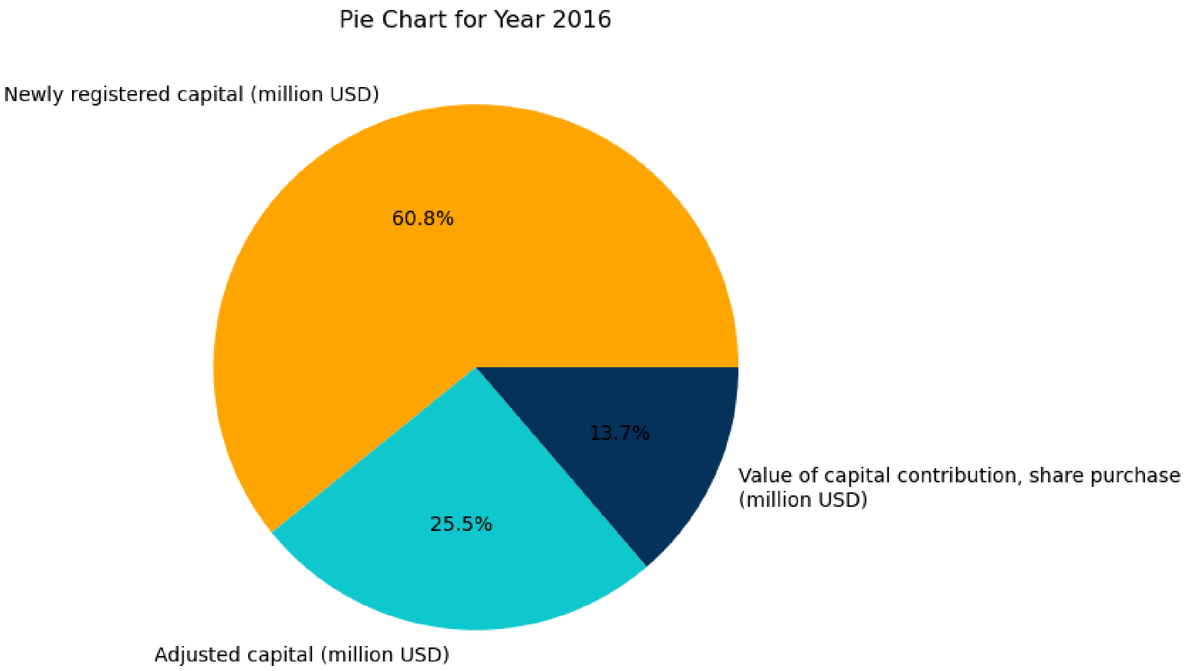
- **Significant Growth in 2017**: The year 2017 saw a remarkable increase in total investment, with an increase of *$10,644.19 million*, representing a *42.64%* rise compared to 2016. This could be due to a surge in new projects or adjustments in investment policies that attracted more foreign capital.
- **Decline in 2020**: The year 2020 was challenging, with total investment decreasing by *25.98%* compared to 2019. This significant drop could be attributed to the impact of the *COVID-19* pandemic, leading to a global economic slowdown.

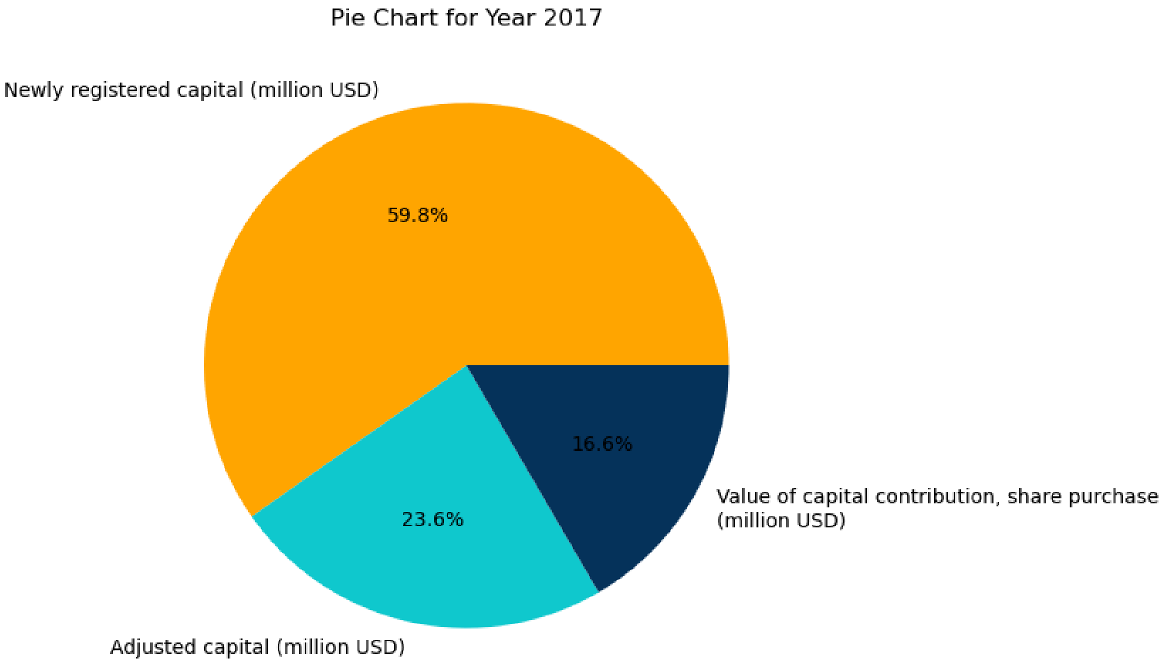
- **Recovery and Mild Volatility**: After the decline in 2020, investment saw a slight recovery in 2021, but again decreased in 2022. This may be the result of a "*recession/economic downturn*" on the world.

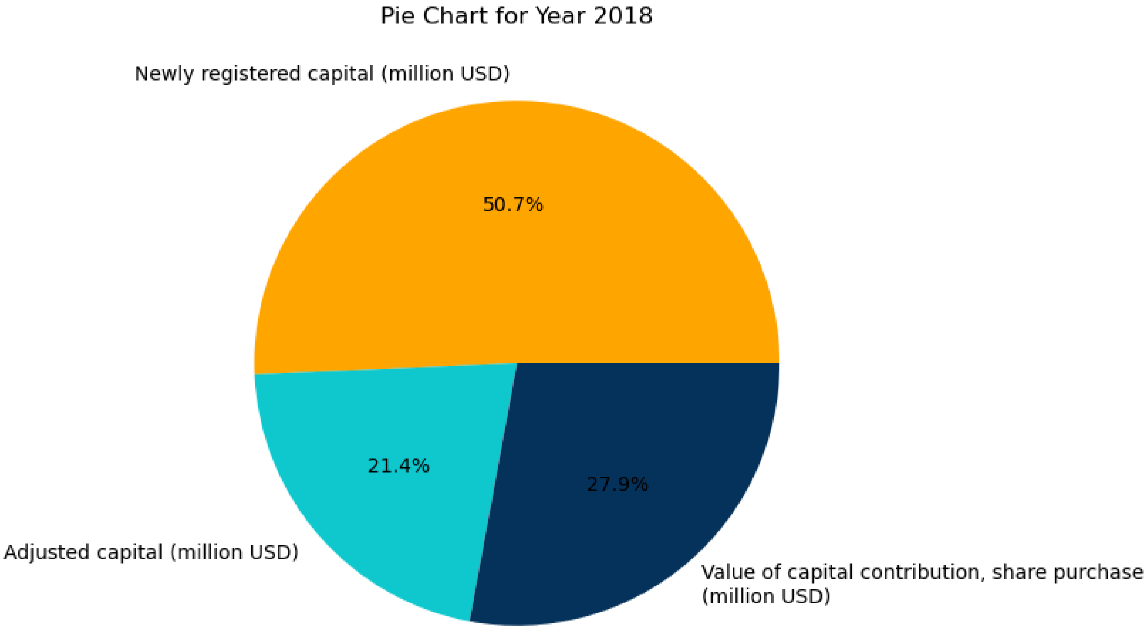
Proportion in total investment

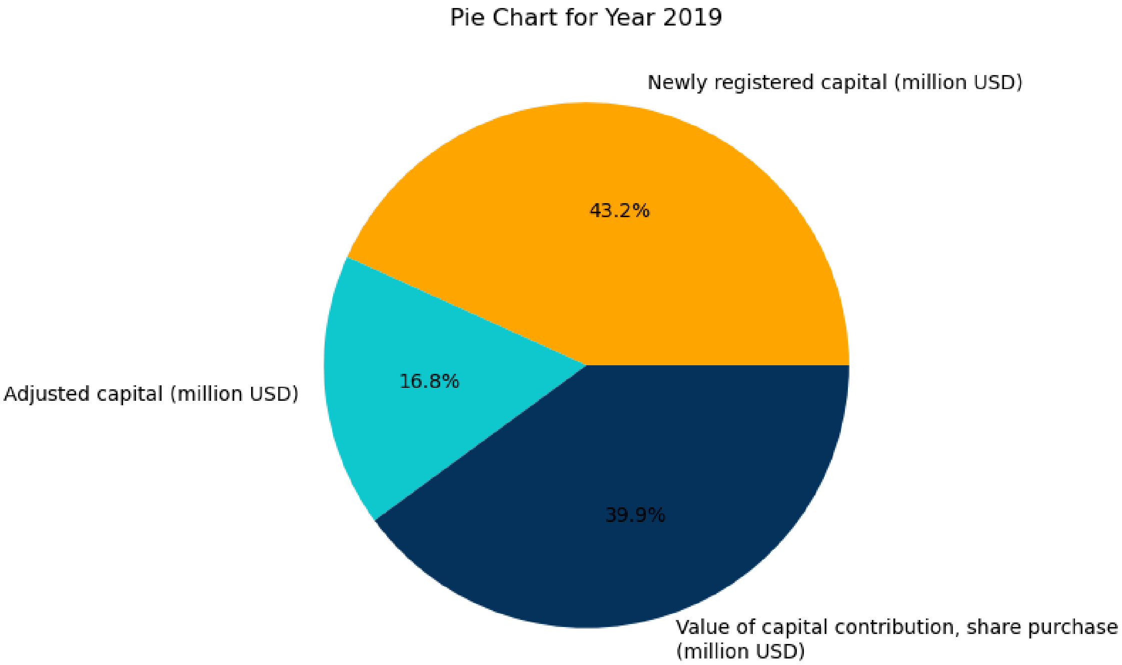
In [ ]:

```
years = n_df['Year'].unique()
for year in years:
    # Filter data for each year
    df_year = n_df[n_df['Year'] == year]
    # Calculate the total each column
    pie_data = df_year[['Newly registered capital (million USD)',
                        'Adjusted capital (million USD)',
                        'Value of capital contribution, share purchase\n(million USD)']].sum()
    # Plot
    plt.figure(figsize=(8, 6))
    plt.pie(pie_data, labels=pie_data.index, autopct='%1.1f%%',
            colors=['orange', '#10c8ce', '#08355e'])
    plt.title(f'Pie Chart for Year {year}')
    plt.show()
```

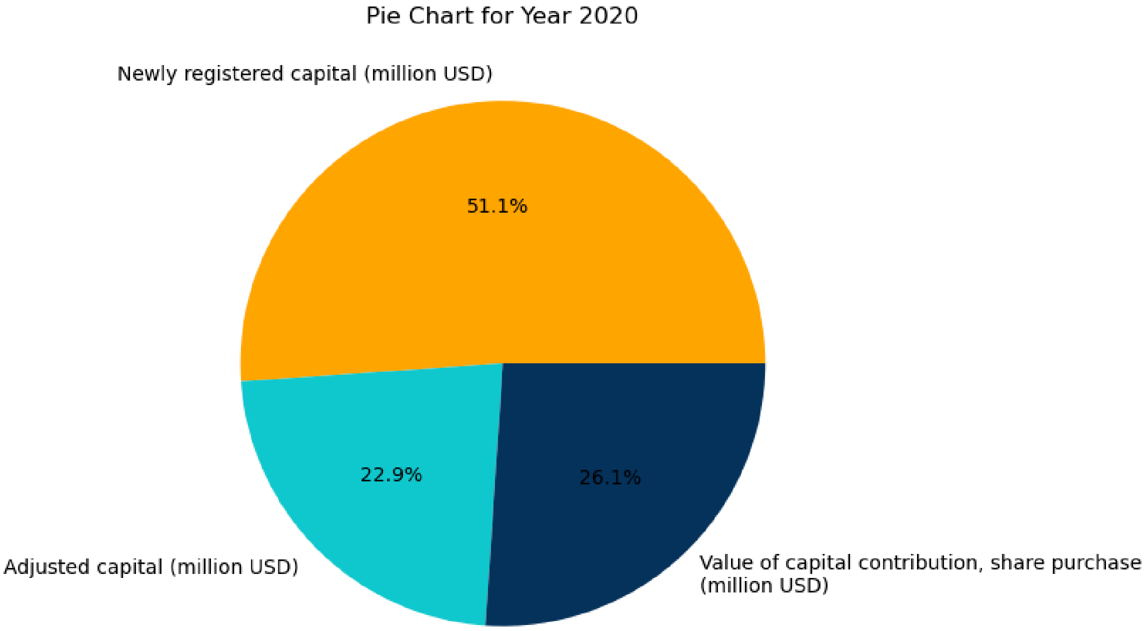



Pie Chart for Year 2017
Newly registered capital (million USD)
59.8%
16.6%
Value of capital contribution, share purchase (million USD)
23.6%
Adjusted capital (million USD)

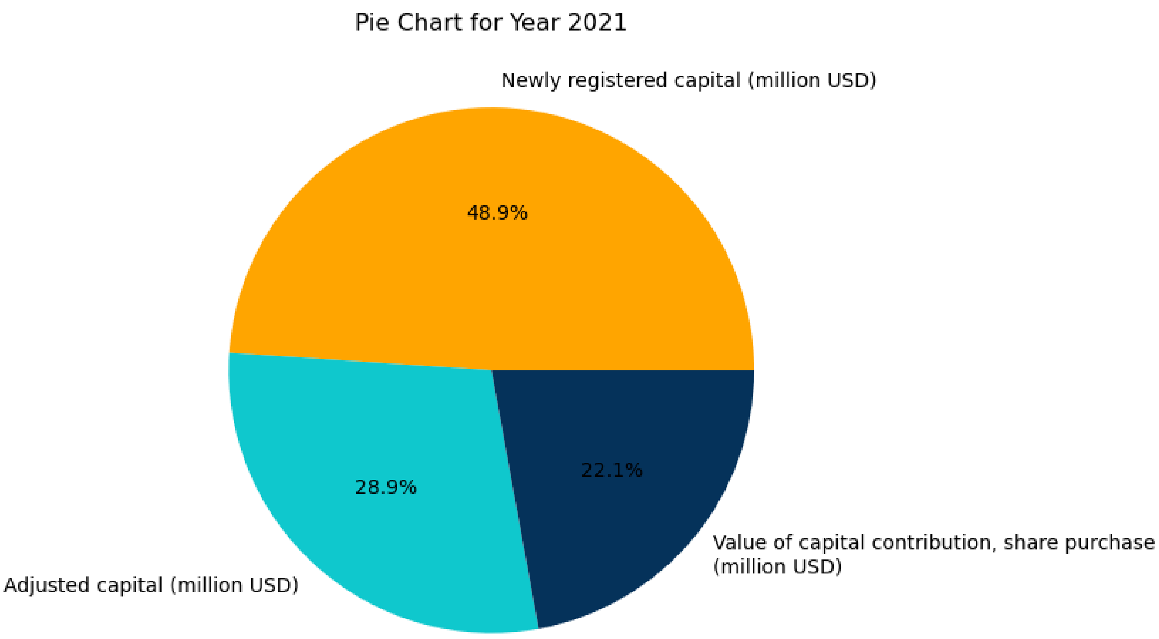

Pie Chart for Year 2018
Newly registered capital (million USD)
50.7%
21.4%
27.9%
Adjusted capital (million USD)
Value of capital contribution, share purchase (million USD)



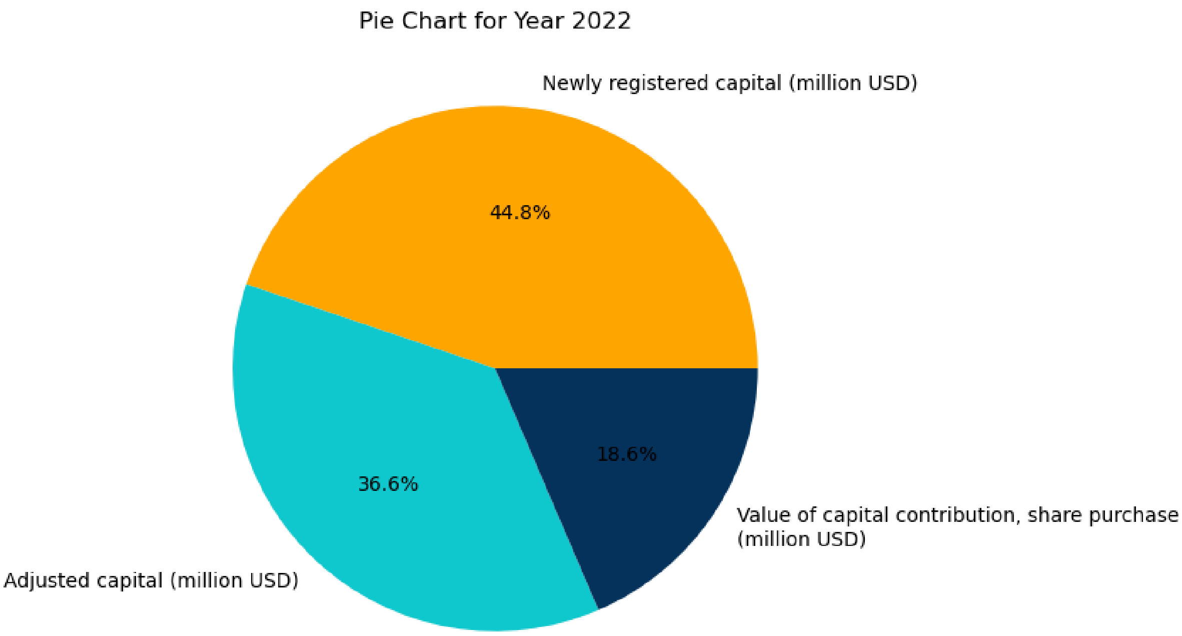
Pie Chart for Year 2019



Pie Chart for Year 2020



Pie Chart for Year 2021



Pie Chart for Year 2022

From the pie charts of the years 2016-2022, we can infer the following:

- The values have not fluctuated significantly in terms of proportion, with `Newly registered capital` consistently holding the largest share.
- `Although Adjusted project number and Adjusted capital` do not account for the majority, they have maintained stability. This demonstrates the effective and long-term collaboration in ongoing projects.
- The `Value of capital contribution, share purchase` saw a strong increase in investment from 2016 to 2019. By 2019, the investment nearly matched the Newly registered capital, showcasing *Vietnam's development potential*. However, due to the impact of *COVID-19*, there was a regression during the 2020-2022 period.

In [ ]:

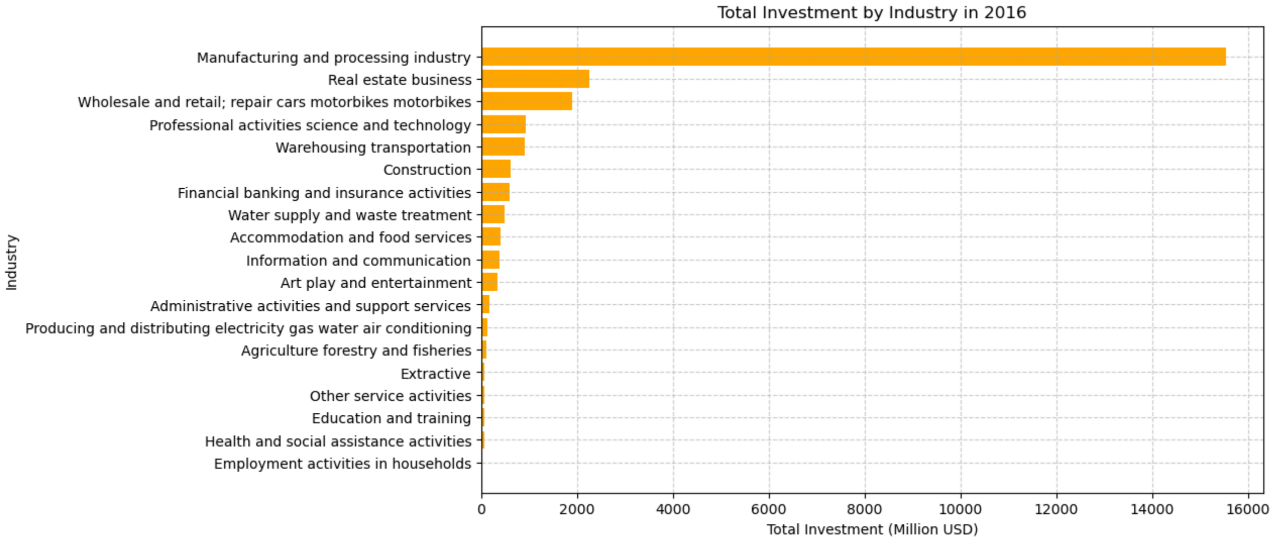
```
'''
This cell is used to create a heatmap of the correlation matrix for the selected year. The heatmap is interactive, allowing
But it just in enviroment with runtime download and run it in your local machine.

'''
# Get the unique years in the dataset
```

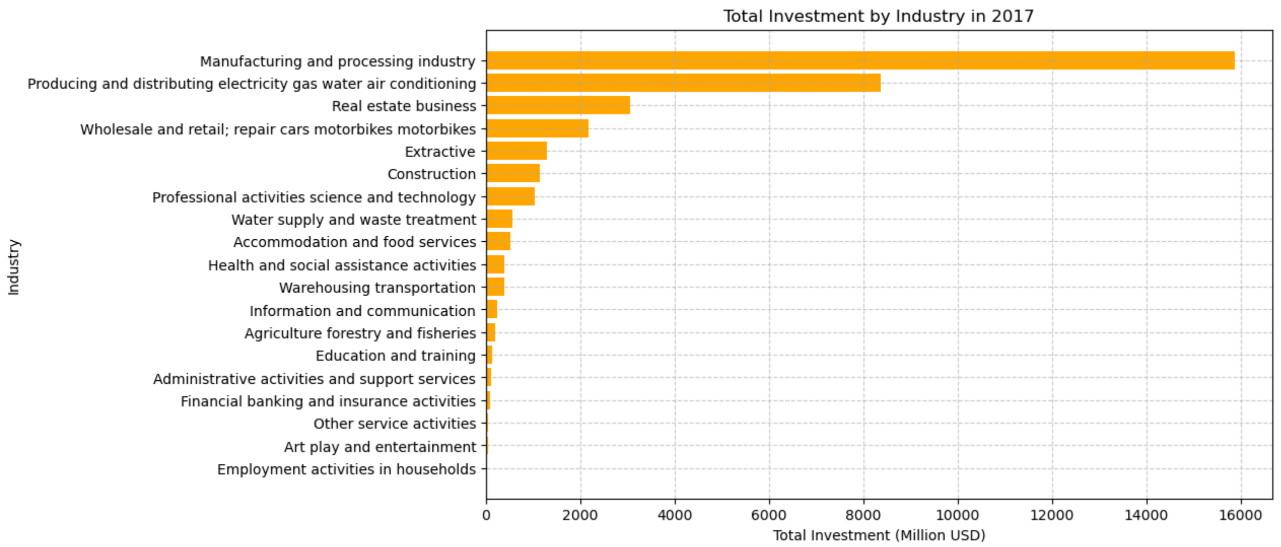
```
years = n_df['Year'].unique()
# Initialize the Panel extension
pn.extension('plotly')
# Initialize the Panel widgets
year_slider = pn.widgets.IntSlider(name='Select Year', start=years[0], end=years[-1], step=1, value=years[0])
# Create a heatmap of the correlation matrix for the selected year
def create_pie_chart(year):
    df_year = n_df[n_df['Year'] == year]
    # Select the columns for the pie chart
    pie_data = df_year[['Number of new projects', 'Adjusted project number', 'Number of times of capital contribution to buy
    # Create the pie chart
    fig = px.pie(pie_data, values=pie_data, names=pie_data.index, title=f'Pie Chart for Year {year}',
                 color_discrete_sequence=['orange', '#10c8ce', '#08355e'])
    fig.update_layout(width=800, height=700)
    return fig
# Update the pie chart based on the selected year
@pn.depends(year_slider)
def update_pie_chart(year):
    return create_pie_chart(year)
# Show the pie chart
pn.Column(year_slider, update_pie_chart).servable()
```

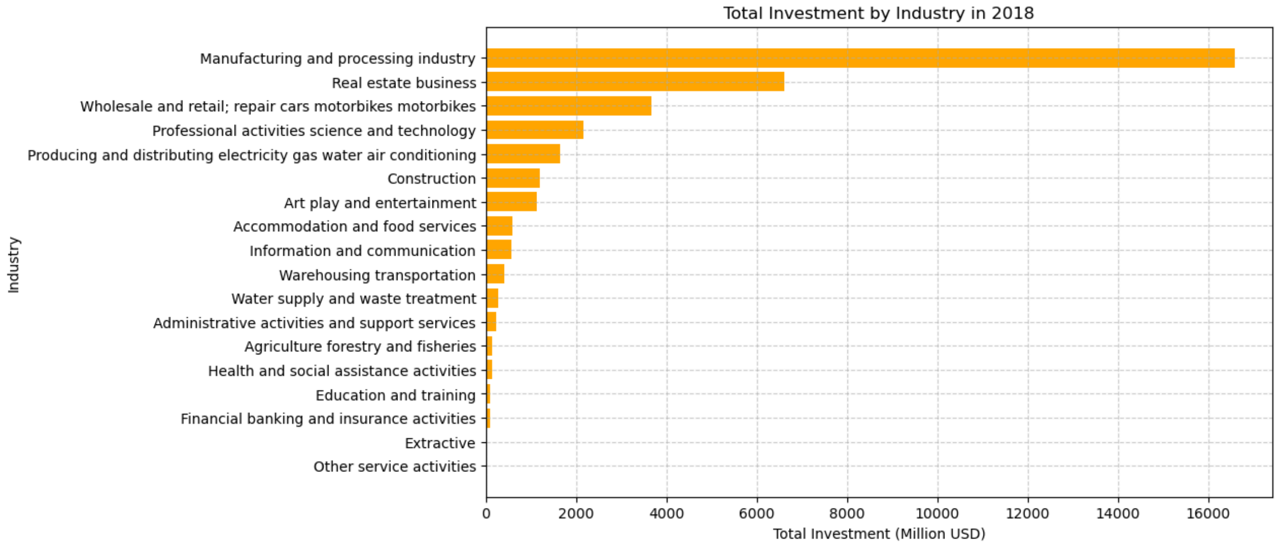
```
Out[ ]:  BokehModel(combine_events=True, render_bundle={'docs_json': {'c3f45c21-7ece-4abe-b6dd-e0314232fe46': {'version…
```

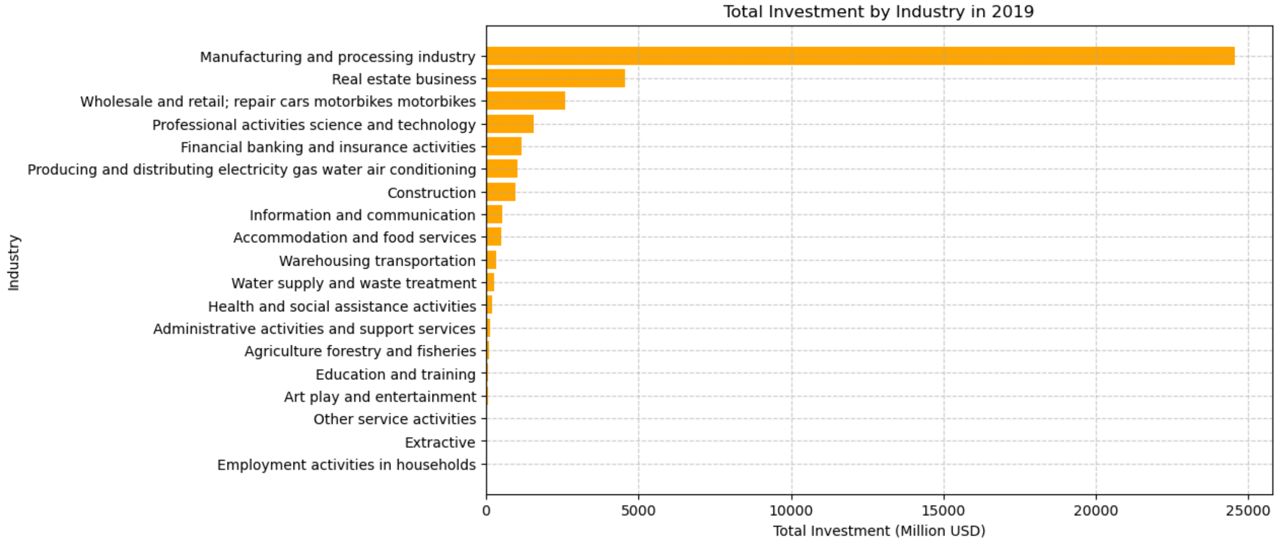
Ranking Industry get total investment each year

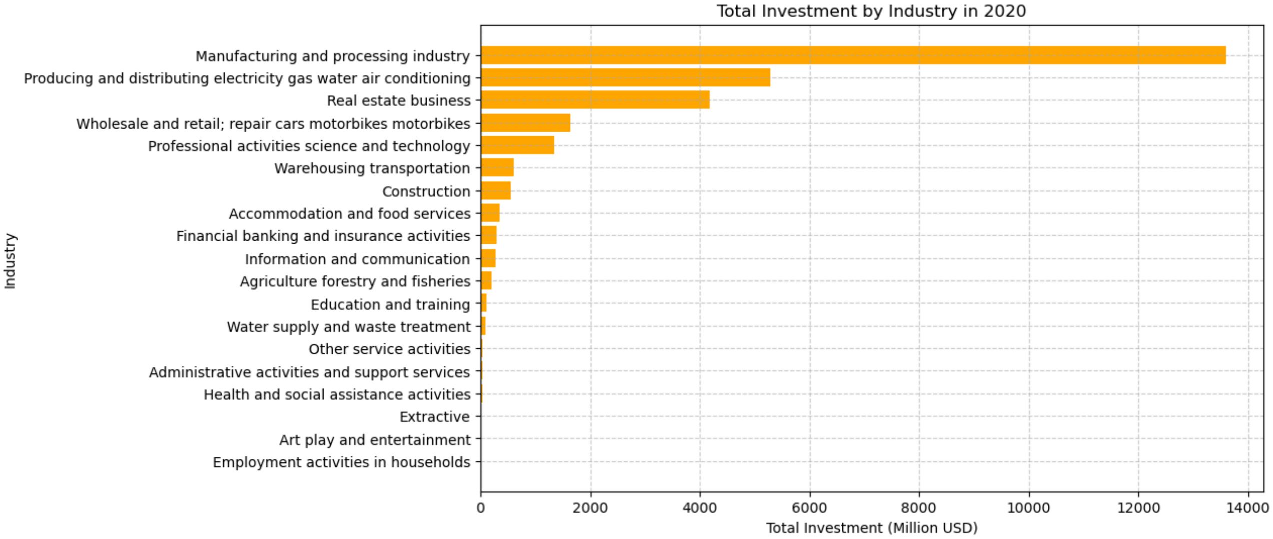
```
In [ ]: years = n_df['Year'].unique()
        # Loop through each year and plot the total investment by industry
        for year in years:
            # Filter data by year
            df_year = n_df[n_df['Year'] == year]
            # Sort the data by Total investment
            df_year.sort_values('Total investment', ascending=True, inplace=True)
            # Set axis values
            x = df_year['Industry'].values
            y = df_year['Total investment'].values
            # Plot
            plt.figure(figsize=(10, 6))
            plt.barh(x, y, color='orange')
            plt.xlabel('Total Investment (Million USD)')
            plt.ylabel('Industry')
            plt.title(f'Total Investment by Industry in {year}')
            plt.grid(True, linestyle='--', alpha=0.6)
            plt.show()
```

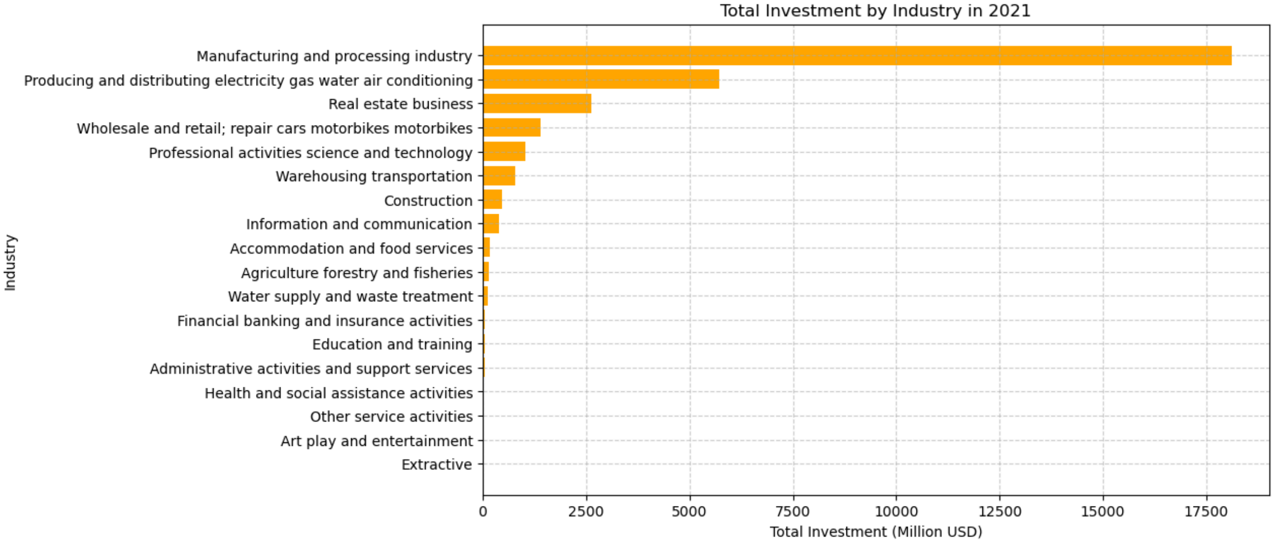


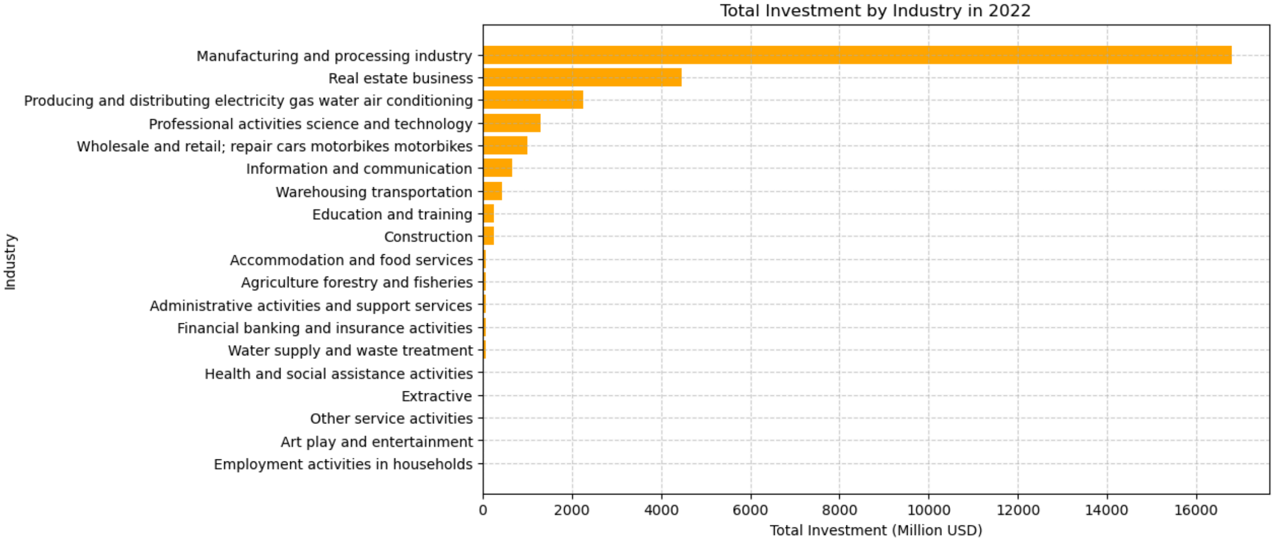

Total Investment by Industry in 2017


Total Investment by Industry in 2018


Total Investment by Industry in 2019







In [ ]:

```
# Create a pivot table
pivot_df = n_df.pivot_table(index='Industry', columns='Year', values='Total investment', aggfunc='sum')
# Sort the data by the total investment in 2020
pivot_df = pivot_df.sort_values(by=2020, ascending=True)
pivot_df.fillna(0, inplace=True)
pivot_df
```

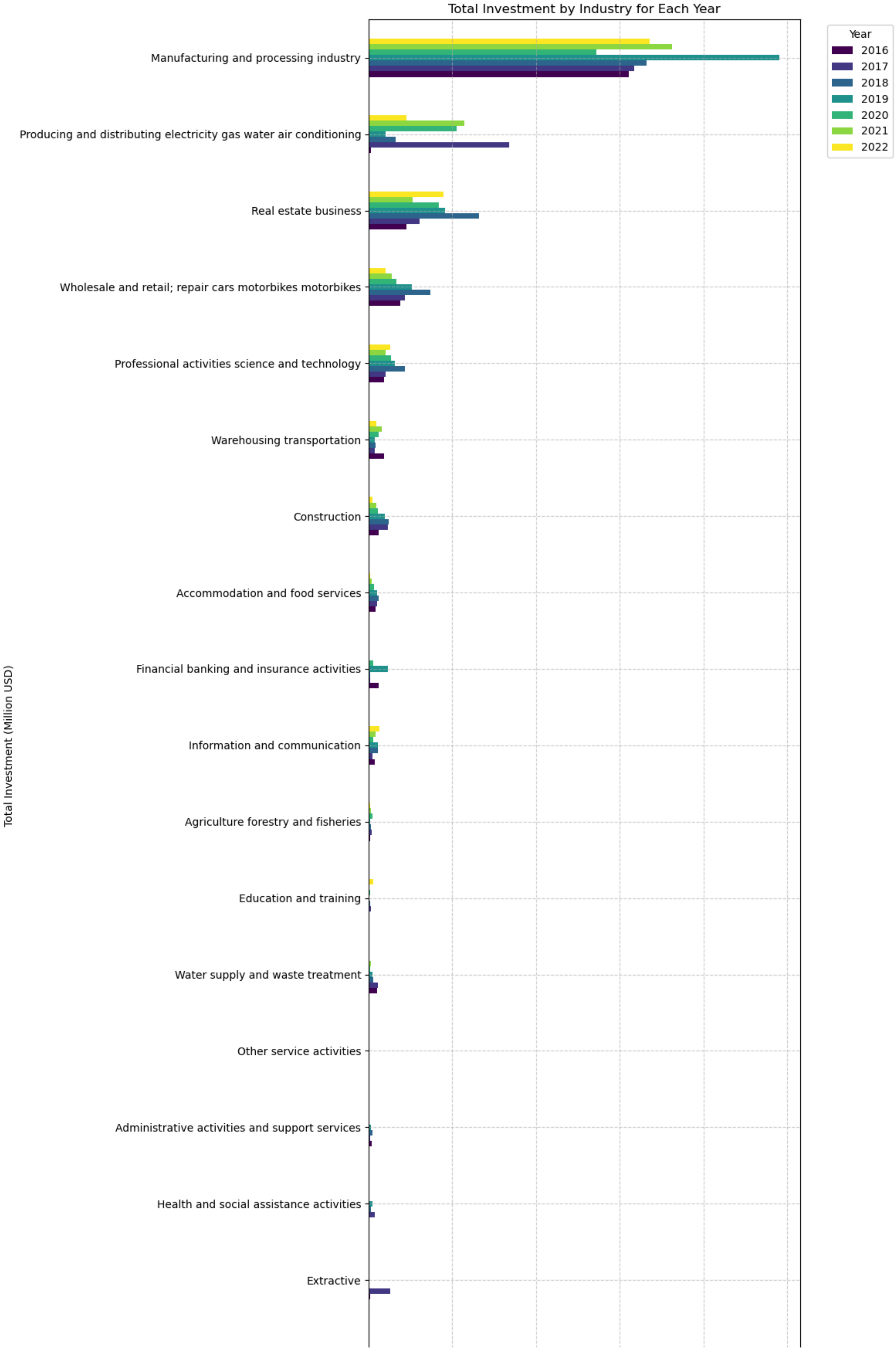
Out[ ]:

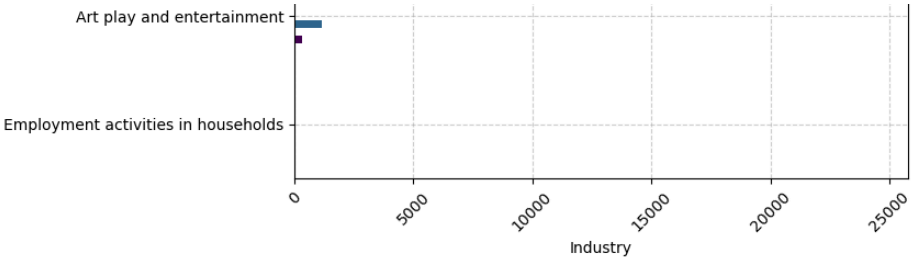
Year	2016	2017	2018	2019	2020	2021	2022
Industry							
Employment activities in households	4.05	0.50	0.00	0.43	2.70	0.00	0.55
Art play and entertainment	329.80	37.72	1133.64	62.73	5.16	1.68	3.65
Extractive	70.02	1288.90	25.40	35.59	6.37	1.48	19.09
Health and social assistance activities	52.08	387.53	132.81	211.45	32.57	4.52	20.73
Administrative activities and support services	160.36	109.03	213.97	123.61	40.38	46.53	64.15
Other service activities	67.76	48.74	7.76	47.07	43.77	3.60	5.61
Water supply and waste treatment	488.26	568.00	259.20	249.27	88.01	116.93	57.44
Education and training	60.67	119.97	90.72	64.62	108.34	51.08	253.48
Agriculture forestry and fisheries	99.48	191.55	140.84	99.32	210.64	156.78	68.37
Information and communication	369.28	236.69	560.88	536.60	271.28	404.39	655.37
Financial banking and insurance activities	582.41	88.22	81.84	1171.86	286.84	59.63	57.50
Accommodation and food services	406.69	513.20	578.52	488.89	341.47	167.60	71.71
Construction	610.40	1133.05	1183.07	979.03	559.85	457.28	247.44
Warehousing transportation	911.13	386.58	405.53	346.06	611.93	783.80	438.59
Professional activities science and technology	933.08	1028.07	2147.42	1566.57	1346.56	1023.98	1289.33
Wholesale and retail; repair cars motorbikes motorbikes	1899.21	2163.71	3672.90	2588.11	1645.63	1404.01	1010.21
Real estate business	2245.22	3053.63	6615.33	4569.76	4184.95	2637.42	4451.83
Producing and distributing electricity gas water air conditioning	132.43	8374.07	1631.33	1025.02	5280.03	5711.76	2261.71
Manufacturing and processing industry	15538.63	15875.99	16588.04	24561.78	13601.09	18120.88	16801.96

Multiple horizontal bar ranking of Industry

In [ ]:

```python
# Plot the grouped bar chart
fig, ax = plt.subplots(figsize=(12, 20))  # Set the figure size
colors = plt.cm.viridis(np.linspace(0, 1, len(years)))
# Plot the bar chart
pivot_df.plot(kind='barh', width=0.5, ax=ax, color=colors )
plt.xlabel('Industry')
plt.ylabel('Total Investment (Million USD)')
plt.title('Total Investment by Industry for Each Year')
plt.legend(title='Year', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.grid(True, linestyle='--', alpha=0.6)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```


Total Investment by Industry for Each Year
Total Investment (Million USD)
Manufacturing and processing industry
Producing and distributing electricity gas water air conditioning
Real estate business
Wholesale and retail; repair cars motorbikes motorbikes
Professional activities science and technology
Warehousing transportation
Construction
Accommodation and food services
Financial banking and insurance activities
Information and communication
Agriculture forestry and fisheries
Education and training
Water supply and waste treatment
Other service activities
Administrative activities and support services
Health and social assistance activities
Extractive
Year
2016
2017
2018
2019
2020
2021
2022



In [ ]:

```python
per_of_total = pivot_df.div(pivot_df.sum(axis=0), axis=1) * 100
# Sort the data of all years by the total percentage of investment
by = per_of_total.columns[-1]
per_of_total.sort_values(by, ascending = False, inplace = True)
per_of_total
```

Out[ ]:

Year	2016	2017	2018	2019	2020	2021	2022
Industry							
Manufacturing and processing industry	62.251732	44.589027	46.767449	63.421622	47.444168	58.166714	60.485004
Real estate business	8.994926	8.576372	18.650914	11.799698	14.598203	8.465927	16.026044
Producing and distributing electricity gas water air conditioning	0.530549	23.519266	4.599286	2.646731	18.418129	18.334336	8.141880
Professional activities science and technology	3.738158	2.887419	6.054323	4.045082	4.697154	3.286902	4.641431
Wholesale and retail; repair cars motorbikes motorbikes	7.608722	6.076958	10.355181	6.682827	5.740389	4.506771	3.636633
Information and communication	1.479430	0.664763	1.581316	1.385569	0.946296	1.298063	2.359252
Warehousing transportation	3.650220	1.085742	1.143330	0.893571	2.134572	2.515941	1.578870
Education and training	0.243060	0.336946	0.255771	0.166857	0.377918	0.163963	0.912497
Construction	2.445419	3.182264	3.335485	2.527979	1.952904	1.467836	0.890754
Accommodation and food services	1.629304	1.441365	1.631049	1.262376	1.191137	0.537984	0.258147
Agriculture forestry and fisheries	0.398542	0.537984	0.397077	0.256457	0.734768	0.503252	0.246124
Administrative activities and support services	0.642443	0.306220	0.603256	0.319177	0.140856	0.149358	0.230932
Financial banking and insurance activities	2.333284	0.247773	0.230735	3.025891	1.000573	0.191408	0.206993
Water supply and waste treatment	1.956095	1.595275	0.730775	0.643647	0.307002	0.375337	0.206777
Health and social assistance activities	0.208646	1.088410	0.374438	0.545991	0.113613	0.014509	0.074625
Extractive	0.280518	3.619982	0.071611	0.091898	0.022220	0.004751	0.068722
Other service activities	0.271464	0.136890	0.021878	0.121541	0.152681	0.011556	0.020195
Art play and entertainment	1.321263	0.105940	3.196125	0.161977	0.017999	0.005393	0.013140
Employment activities in households	0.016225	0.001404	0.000000	0.001110	0.009418	0.000000	0.001980

From the ranking plot and multiple barh , we can have below observations:

- `Manufacturing and processing industry` from 2016-2022 has consistently been the industry with the largest investment proportion, ranging from *44-63%*.
- `Manufacturing and processing industry` was significantly impacted by COVID-19 in 2019-2020, with a decrease of *$11 billion*.
- `Real estate business` gradually recovered after 2020, with a continued high demand for real estate.
- `Professional activities science and technology` remained stable, showing that Vietnam's scientific and technological expertise continues to be highly trusted.
- `Warehousing transportation` experienced strong growth during the boom of e-commerce platforms, with the logistics industry driving significant growth in this sector.
- `Wholesale and retail; repair cars motorbikes motorbikes` accounted for a significant proportion, demonstrating that Vietnam consistently has a high demand for personal transportation.

Bivariate Analysis

Compare two trendlines

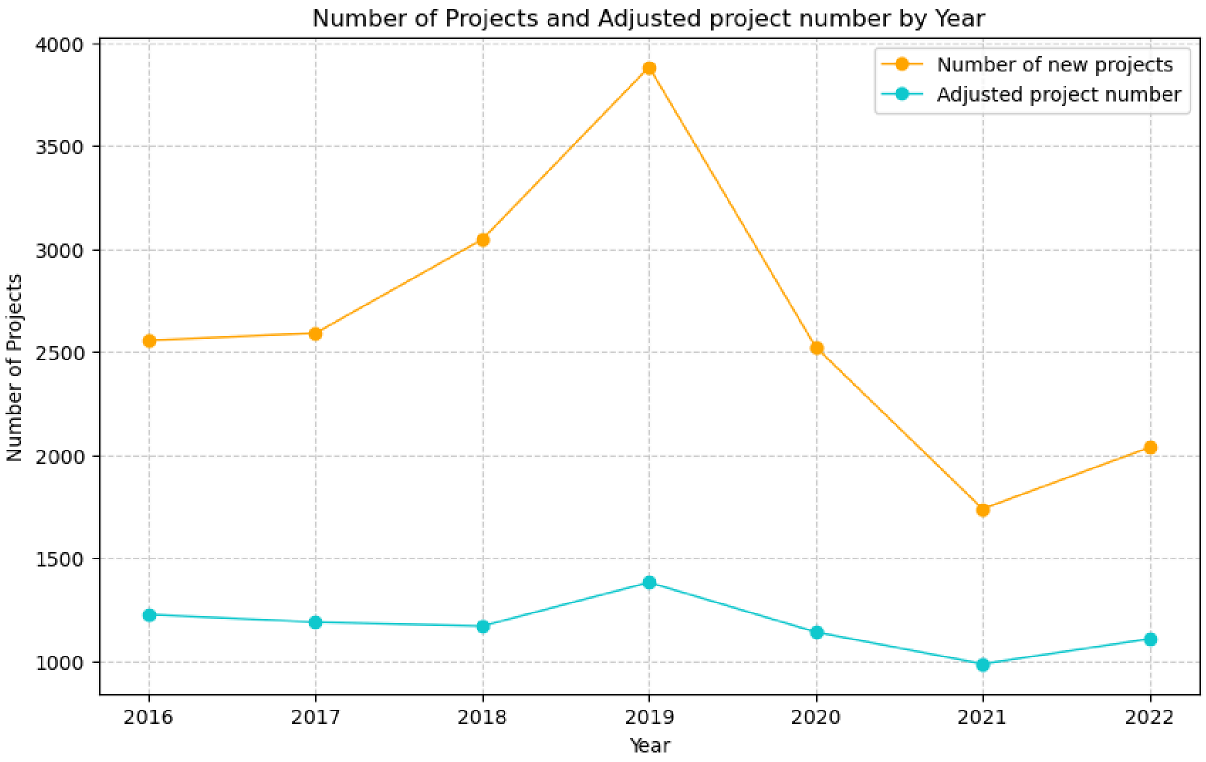
In [ ]:

```python
# Group by year
group_year_df = n_df.groupby('Year')
# Print the size of each group
print(group_year_df.size())
```

```
Year
2016    19
2017    19
2018    18
2019    19
2020    19
2021    18
2022    19
dtype: int64
```

In [ ]:

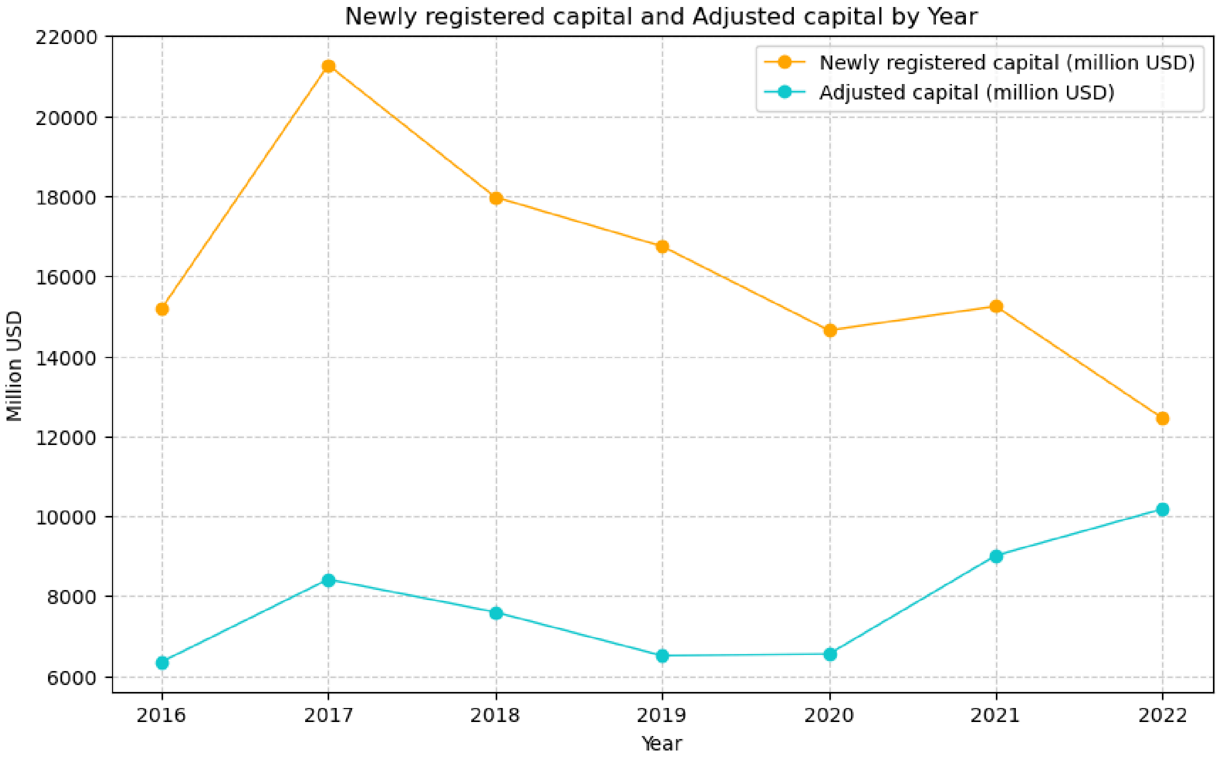
```python
# Set axis values
x = group_year_df['Year'].unique()
y1 = group_year_df['Number of new projects'].sum()
y2 = group_year_df['Adjusted project number'].sum()
# Plot
plt.figure(figsize=(10, 6))
# Line for Number of new projects
plt.plot(x, y1, marker='o', color='orange', label='Number of new projects',linewidth=1)
# Line for Adjusted project number
plt.plot(x, y2, marker='o', color='#10c8ce', label='Adjusted project number',linewidth=1)
plt.xlabel('Year')
plt.ylabel('Number of Projects')
plt.title('Number of Projects and Adjusted project number by Year')
plt.legend()
plt.grid(True, linestyle='--', alpha=0.6)
plt.show()
```



`Number of new projects` and `Adjusted project number` are directly proportional to each other, which reflects the growing trust in Vietnam from the international community, both from new and existing investment partners.

In [ ]:

```python
# Set axis values
x = group_year_df['Year'].unique()
y1 = group_year_df['Newly registered capital (million USD)'].sum()
y2 = group_year_df['Adjusted capital (million USD)'].sum()
# Plot
plt.figure(figsize=(10, 6))
# Line for Number of new projects
plt.plot(x, y1, marker='o', color='orange', label='Newly registered capital (million USD)',linewidth=1)
# Line for Adjusted project number
plt.plot(x, y2, marker='o', color='#10c8ce', label='Adjusted capital (million USD)',linewidth=1)
plt.xlabel('Year')
plt.ylabel('Million USD')
plt.title('Newly registered capital and Adjusted capital by Year')
plt.legend()
plt.grid(True, linestyle='--', alpha=0.6)
plt.show()
```

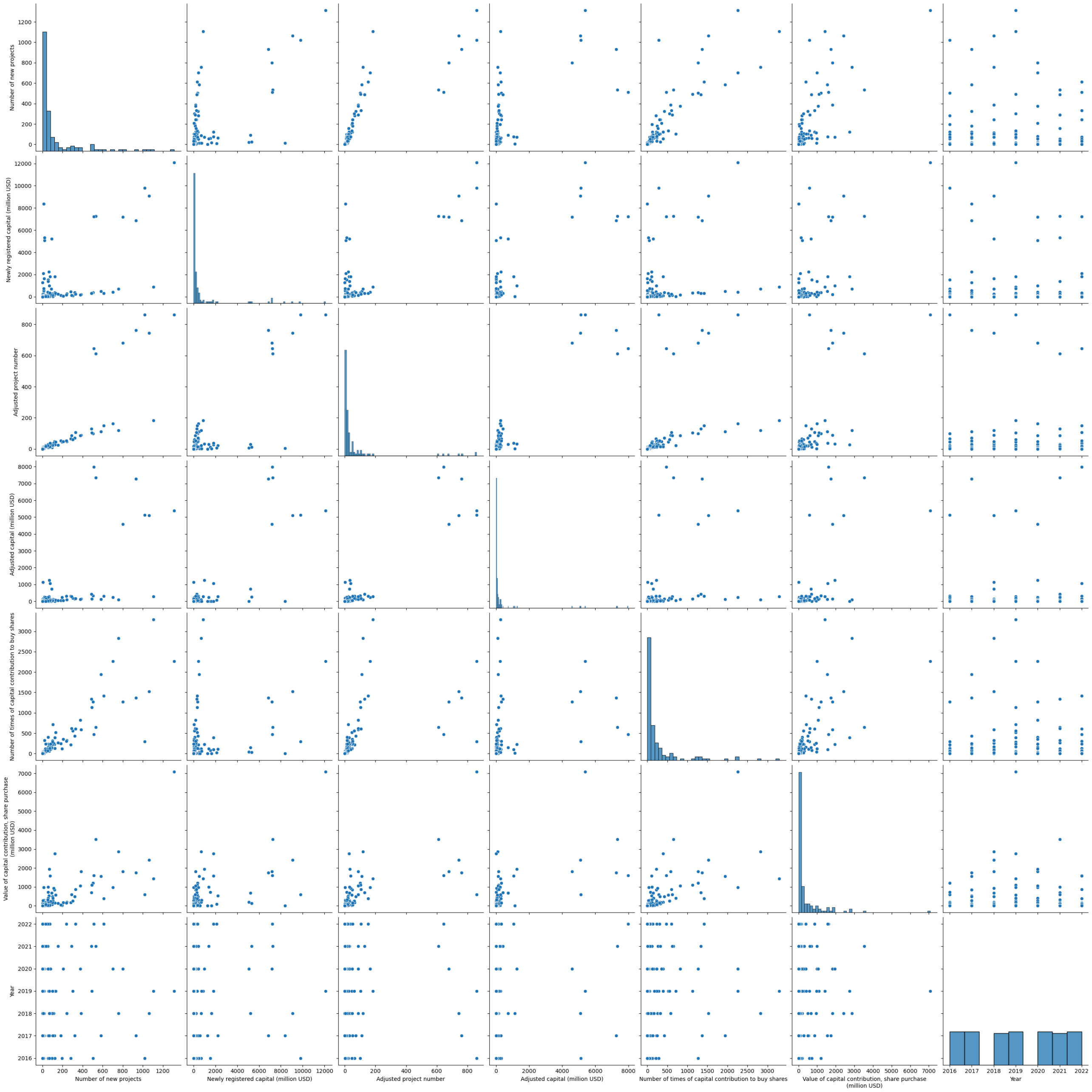


The global *recession/economic downturn* has caused a significant drop in `Newly registered capital (million USD)`. However, `Adjusted capital (million USD)` has still increased, demonstrating a strong and ongoing partnership with investment partners.

Pairplot correlated data

In [ ]:

```
sns.pairplot(data=n_df.drop(['Industry','Total investment'],axis=1), height=4)
plt.show()
```



Multivariate Analysis

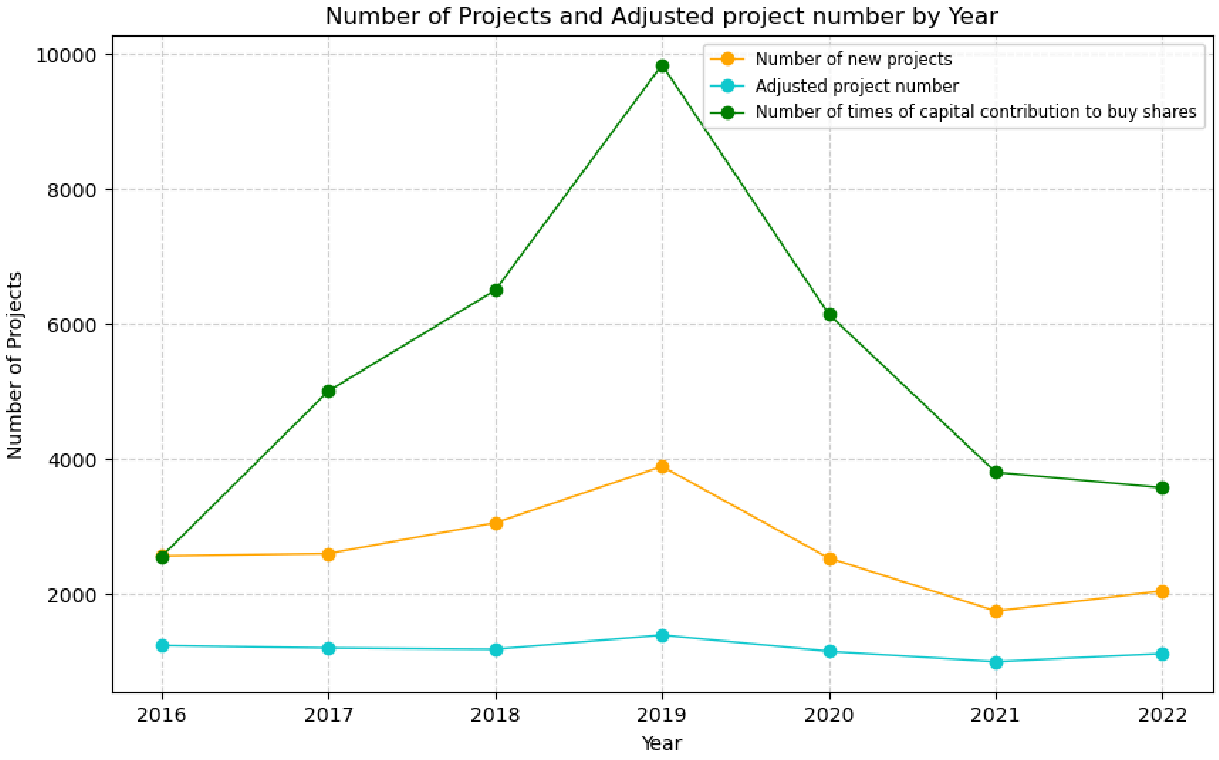
Compare three trendlines

In [ ]:
```
# Group by year
group_year_df = n_df.groupby('Year')
# Print the size of each group
print(group_year_df.size())
```

```
Year
2016    19
2017    19
2018    18
2019    19
2020    19
2021    18
2022    19
dtype: int64
```

In [ ]:
```
# Set axis values
x = group_year_df['Year'].unique()
y1 = group_year_df['Number of new projects'].sum()
y2 = group_year_df['Adjusted project number'].sum()
```

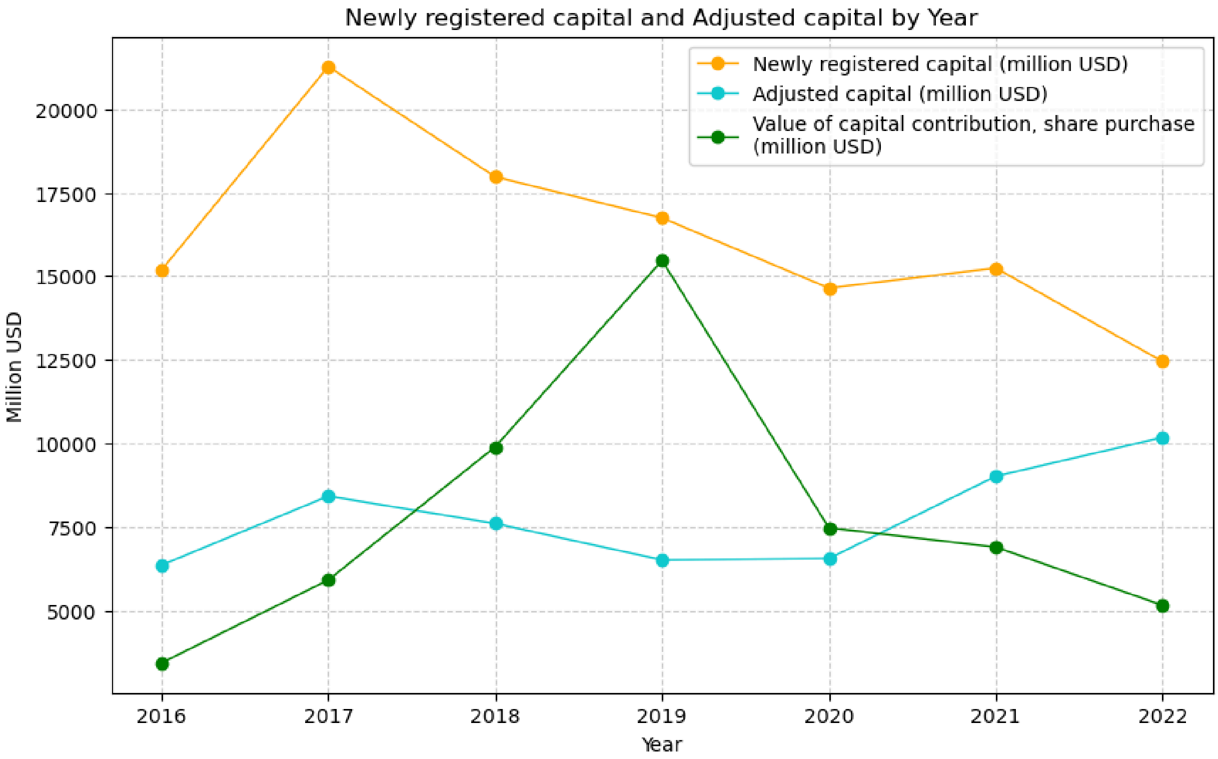
```
y3 = group_year_df['Number of times of capital contribution to buy shares'].sum()
# Plot
plt.figure(figsize=(10, 6))
# Line for Number of new projects
plt.plot(x, y1, marker='o', color='orange', label='Number of new projects',linewidth=1)
# Line for Adjusted project number
plt.plot(x, y2, marker='o', color='#10c8ce', label='Adjusted project number',linewidth=1)
# Line for Number of times of capital contribution to buy shares
plt.plot(x, y3, marker='o', color='green', label='Number of times of capital contribution to buy shares',linewidth=1)
plt.xlabel('Year')
plt.ylabel('Number of Projects')
plt.title('Number of Projects and Adjusted project number by Year')
plt.legend(loc='upper right', fontsize='small')
plt.grid(True, linestyle='--', alpha=0.6)
plt.show()
```



`Number of new projects`, `Adjusted project number` and `Number of times of capital contribution to buy shares` are a similar line pattern, indicating stability in both collaboration and investment.

In [ ]:

```
# Set axis values
x = group_year_df['Year'].unique()
y1 = group_year_df['Newly registered capital (million USD)'].sum()
y2 = group_year_df['Adjusted capital (million USD)'].sum()
y3 = group_year_df['Value of capital contribution, share purchase\n(million USD)'].sum()
# Plot
plt.figure(figsize=(10, 6))
# Line for Number of new projects
plt.plot(x, y1, marker='o', color='orange', label='Newly registered capital (million USD)',linewidth=1)
# Line for Adjusted project number
plt.plot(x, y2, marker='o', color='#10c8ce', label='Adjusted capital (million USD)',linewidth=1)
# Line for Adjusted project number
plt.plot(x, y3, marker='o', color='green', label='Value of capital contribution, share purchase\n(million USD)',linewidth=1)
plt.xlabel('Year')
plt.ylabel('Million USD')
plt.title('Newly registered capital and Adjusted capital by Year')
plt.legend()
plt.grid(True, linestyle='--', alpha=0.6)
plt.show()
```



`Newly registered capital` continues to hold the largest proportion of Total investment, with `Adjusted capital` showing similar trends as mentioned in the previous chart. The *recession/economic downturn* has led to a significant decline in `Value of capital contribution, share purchase` in Vietnam during the 2020-2022 period.

In [ ]:

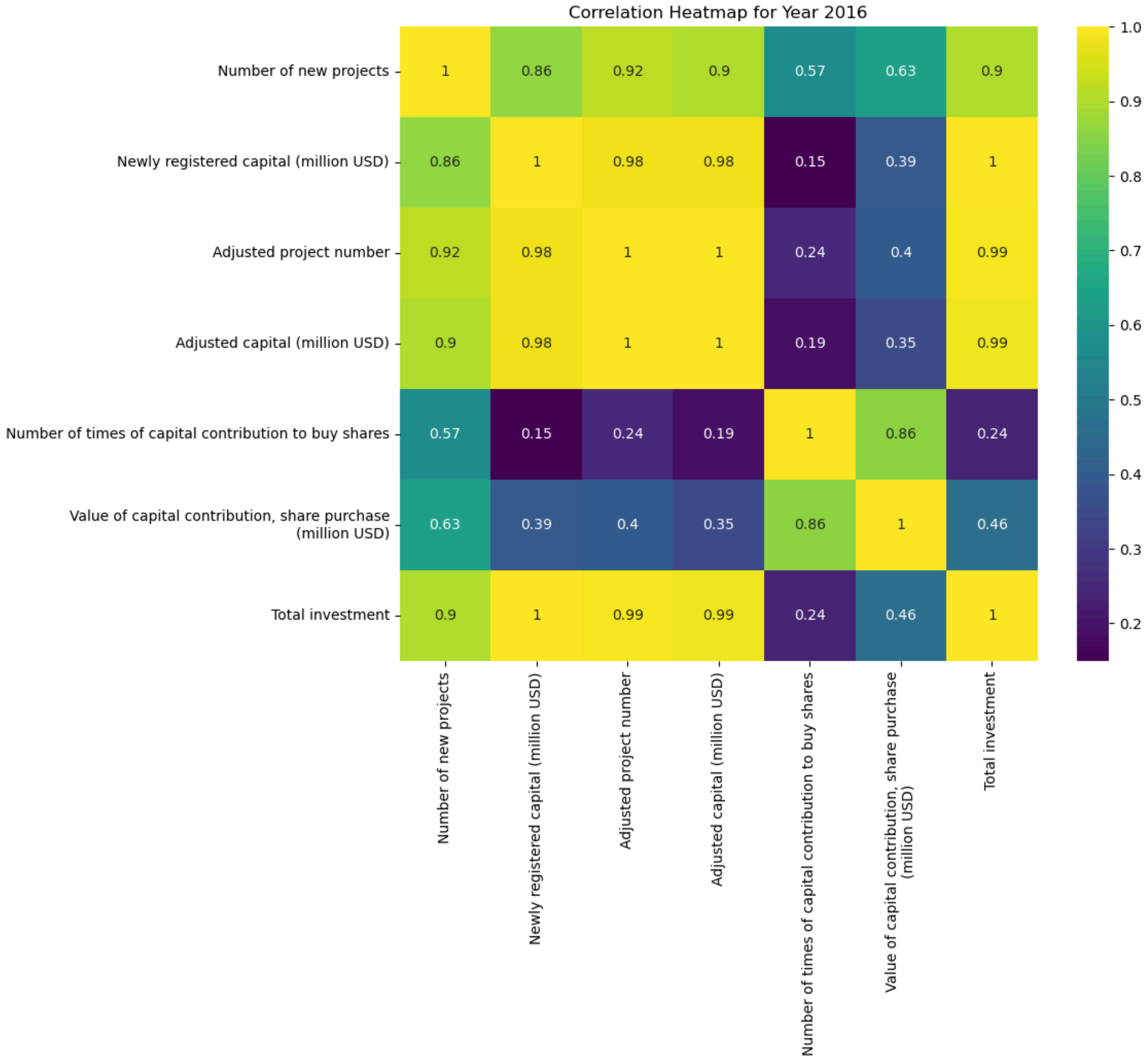
```
'''
This cell is used to create a heatmap of the correlation matrix for the selected year. The heatmap is interactive, allowing
But it just in enviroment with runtime download and run it in your local machine.

'''
# Get the unique years in the dataset
years = n_df['Year'].unique()
# Initialize the Panel extension
pn.extension('plotly')
# Initialize slider widget
year_slider = pn.widgets.IntSlider(name='Select Year', start=years[0], end=years[-1], step=1, value=years[0])
# Create a function to generate a heatmap for a given year
def create_heatmap(year):
    df_year = n_df[n_df['Year'] == year].drop(columns=['Industry', 'Year'])
    corr_matrix = df_year.corr()
    # Create the heatmap
    fig = px.imshow(corr_matrix, text_auto=True, aspect="auto", color_continuous_scale='Viridis')
    fig.update_layout(title=f'Correlation Heatmap for Year {year}', width=800, height=700)
    return fig
# Update the heatmap based on the selected year
@pn.depends(year_slider)
def update_heatmap(year):
    return create_heatmap(year)
# Show the heatmap
pn.Column(year_slider, update_heatmap).servable()
```

Out[ ]:

```
BokehModel(combine_events=True, render_bundle={'docs_json': {'b83e0dbf-7f79-4f2c-b71c-936e8799c9b8': {'version…
```

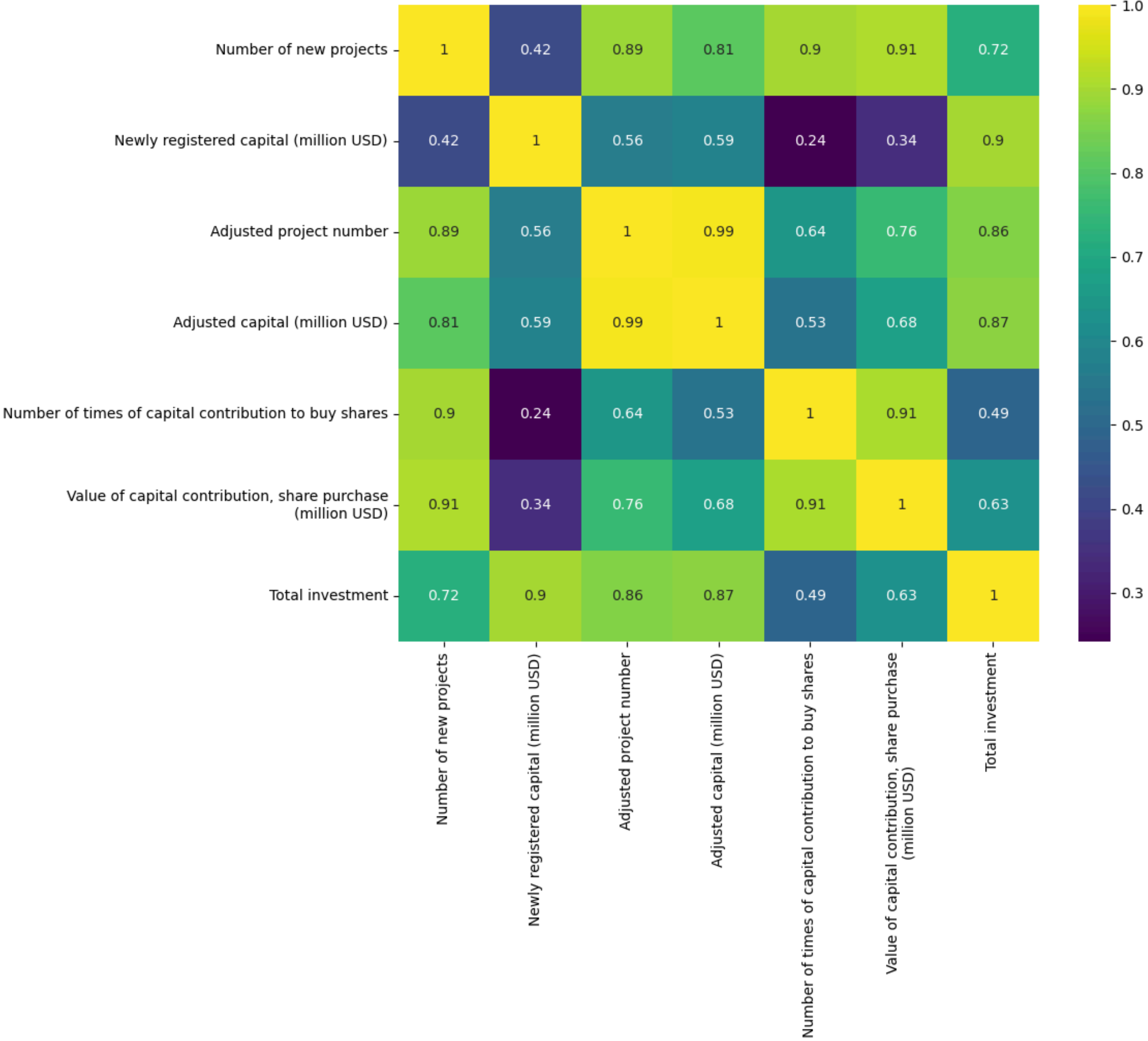
In [ ]:

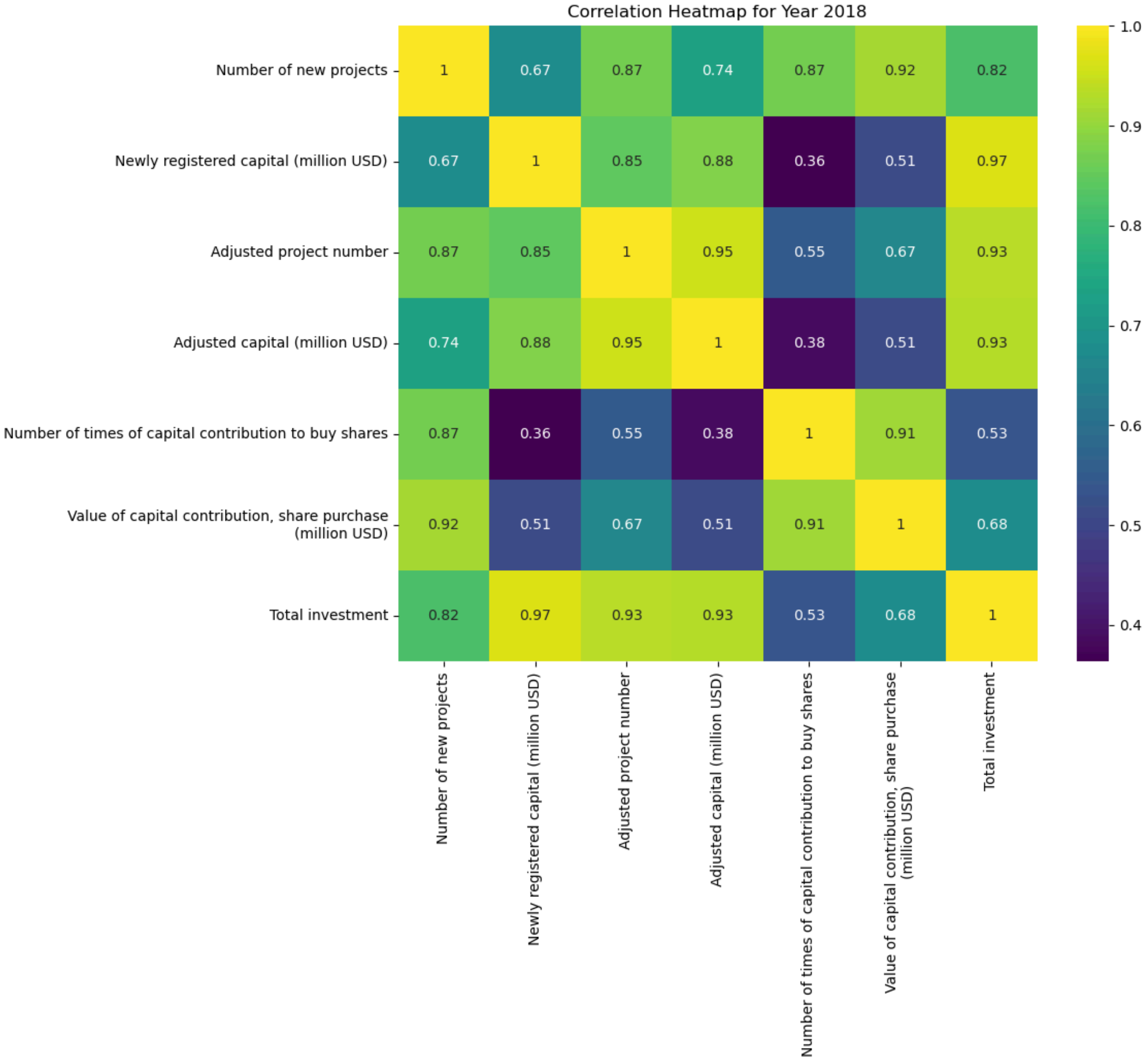
```
years = n_df['Year'].unique()
for year in years:
    # Filter data by year
    df_year = n_df[n_df['Year'] == year].drop(columns=['Industry', 'Year'])
    # Create the correlation matrix
    corr_matrix = df_year.corr()
    # Plot
    plt.figure(figsize=(10, 8))
    sns.heatmap(corr_matrix, annot=True, cmap='viridis', cbar=True)
    plt.title(f'Correlation Heatmap for Year {year}')
    # Show the plot
    plt.show()
```

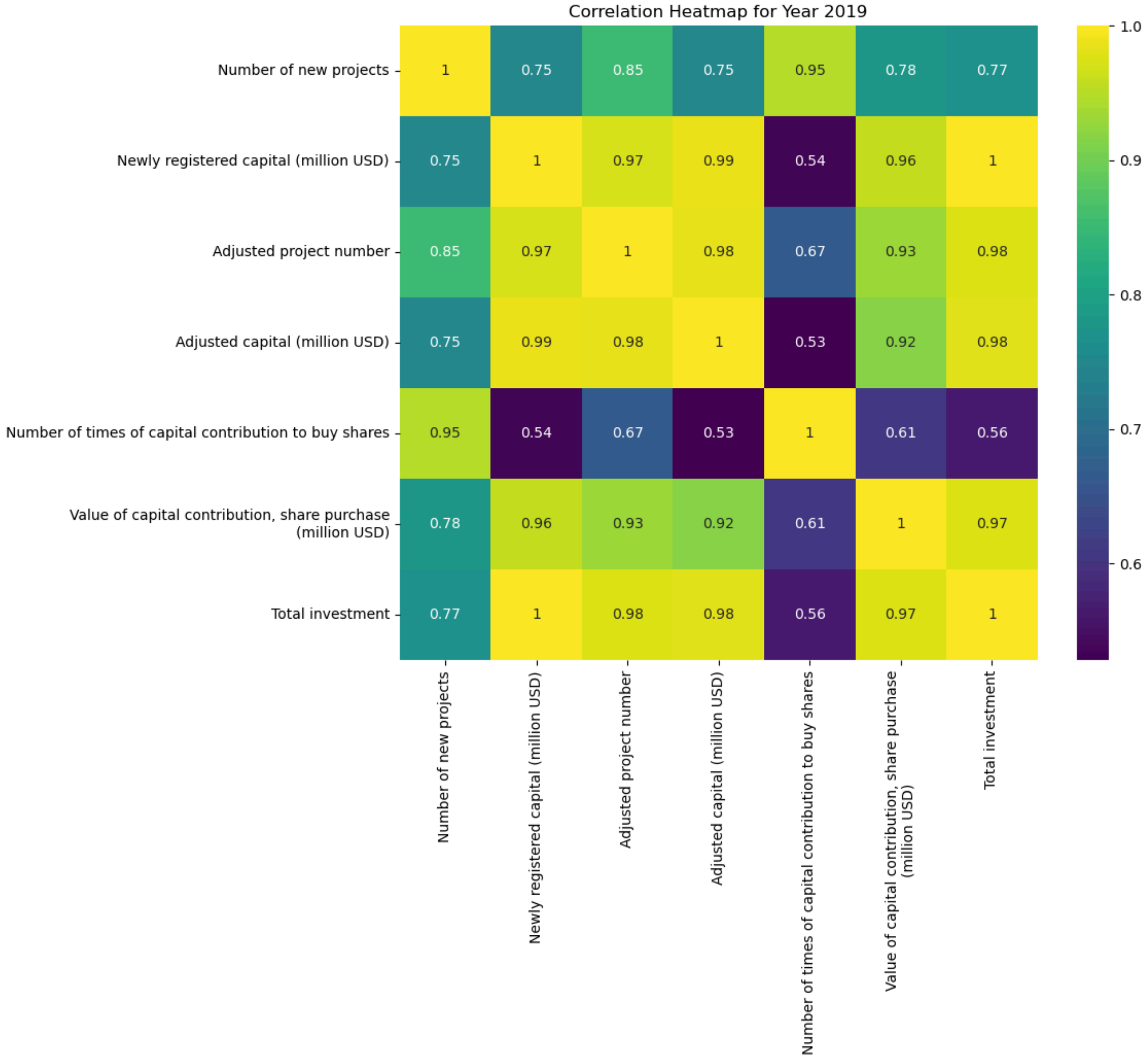


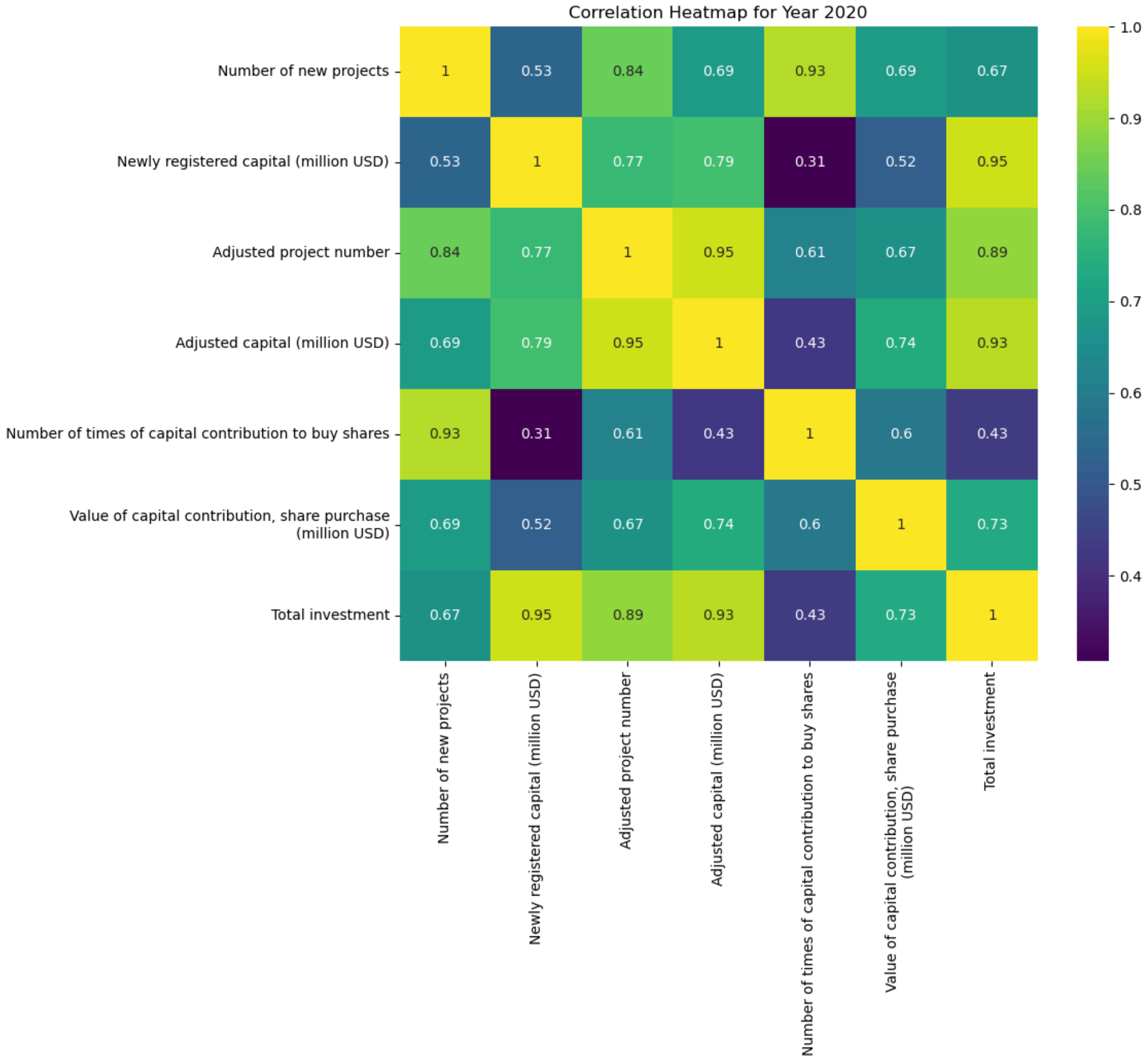
Correlation Heatmap for Year 2016

Correlation Heatmap for Year 2017

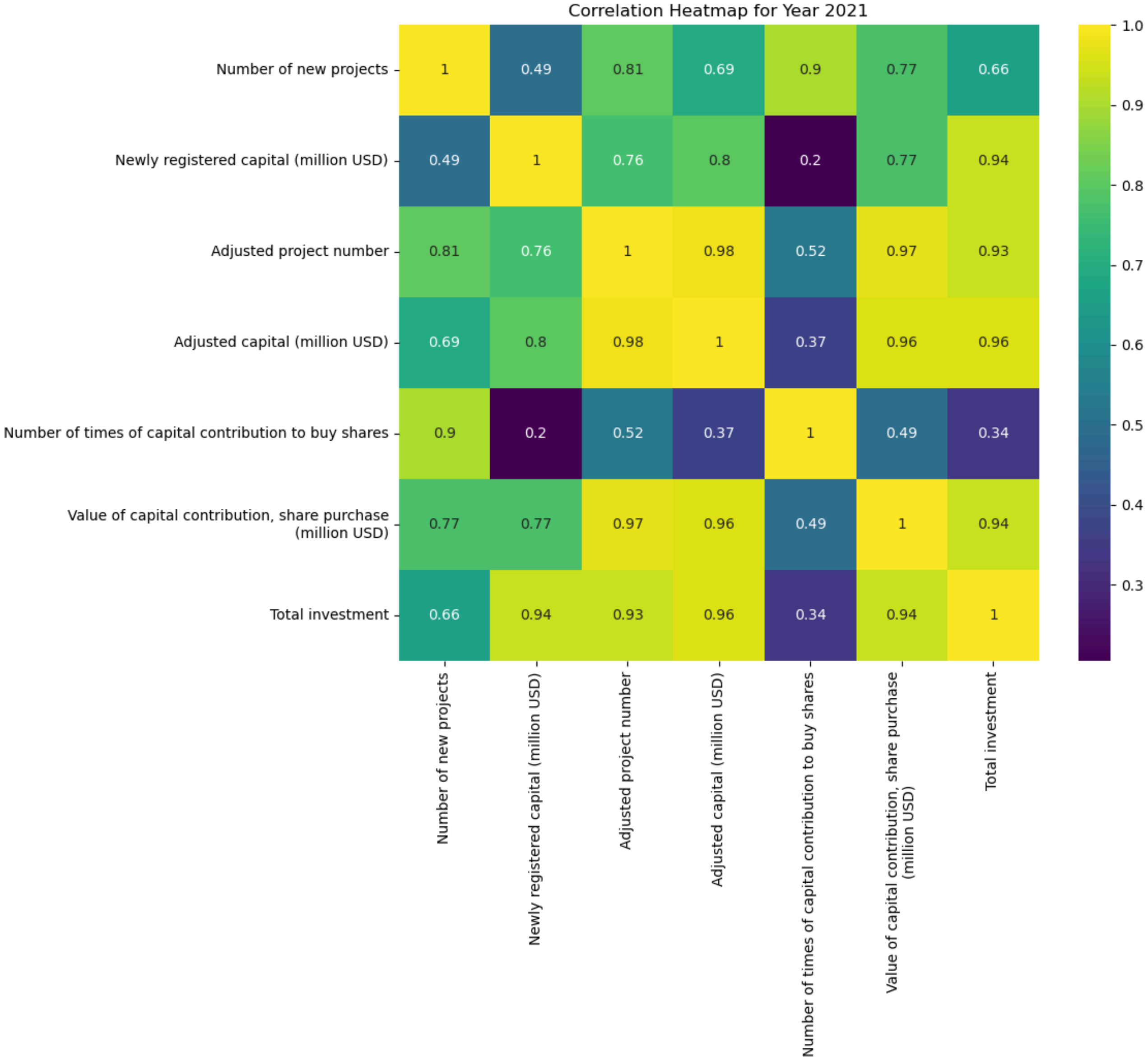


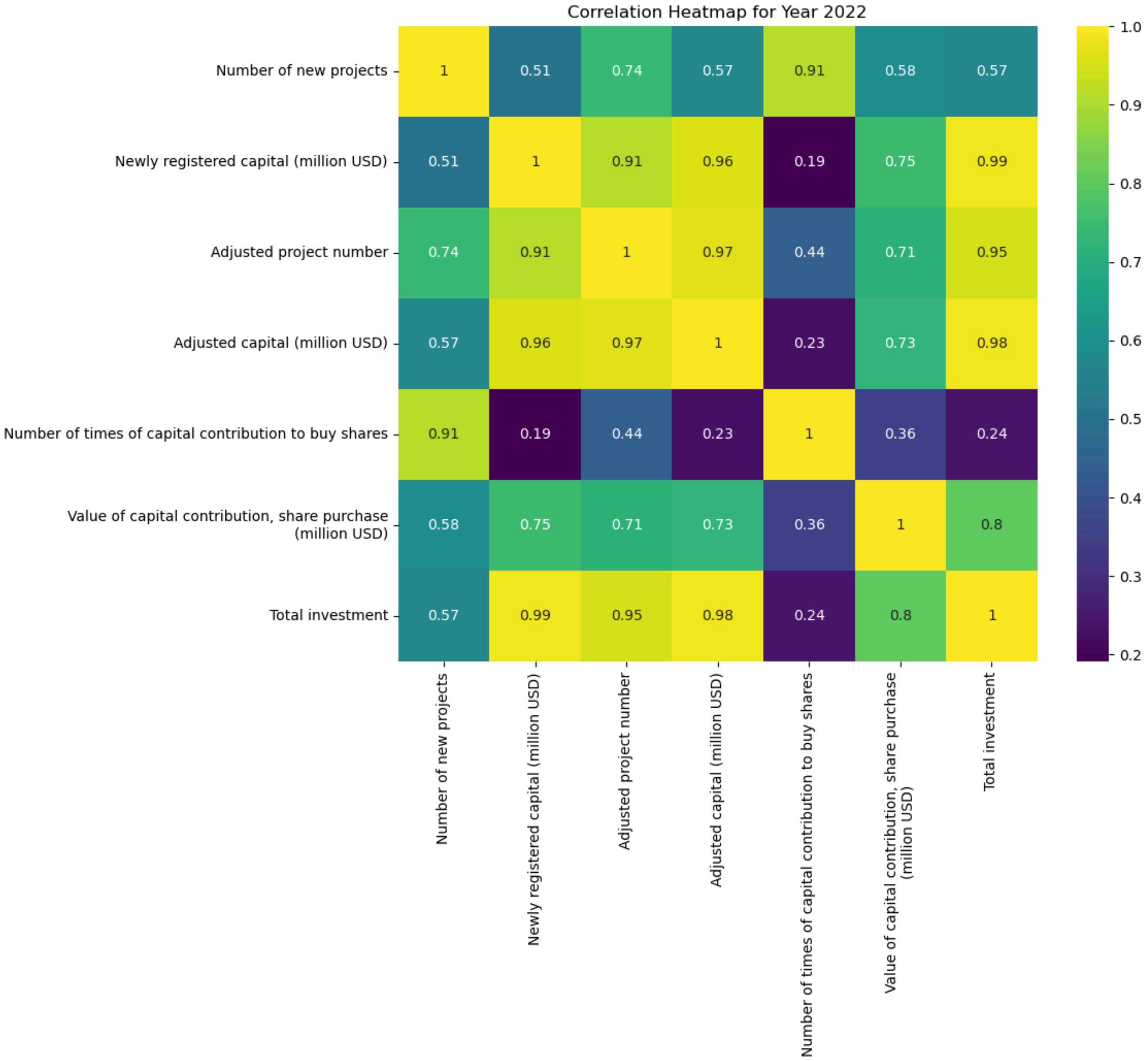

Correlation Heatmap for Year 2018
Number of new projects
Newly registered capital (million USD)
Adjusted project number
Adjusted capital (million USD)
Number of times of capital contribution to buy shares
Value of capital contribution, share purchase (million USD)
Total investment
1 0.67 0.87 0.74 0.87 0.92 0.82
0.67 1 0.85 0.88 0.36 0.51 0.97
0.87 0.85 1 0.95 0.55 0.67 0.93
0.74 0.88 0.95 1 0.38 0.51 0.93
0.87 0.36 0.55 0.38 1 0.91 0.53
0.92 0.51 0.67 0.51 0.91 1 0.68
0.82 0.97 0.93 0.93 0.53 0.68 1
1.0
0.9
0.8
0.7
0.6
0.5
0.4


Correlation Heatmap for Year 2019
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
Value of capital contribution, share purchase (million USD)
Total investment
1 0.75 0.85 0.75 0.95 0.78 0.77
0.75 1 0.97 0.99 0.54 0.96 1
0.85 0.97 1 0.98 0.67 0.93 0.98
0.75 0.99 0.98 1 0.53 0.92 0.98
0.95 0.54 0.67 0.53 1 0.61 0.56
0.78 0.96 0.93 0.92 0.61 1 0.97
0.77 1 0.98 0.98 0.56 0.97 1
1.0
0.9
0.8
0.7
0.6



Correlation Heatmap for Year 2020


Correlation Heatmap for Year 2021



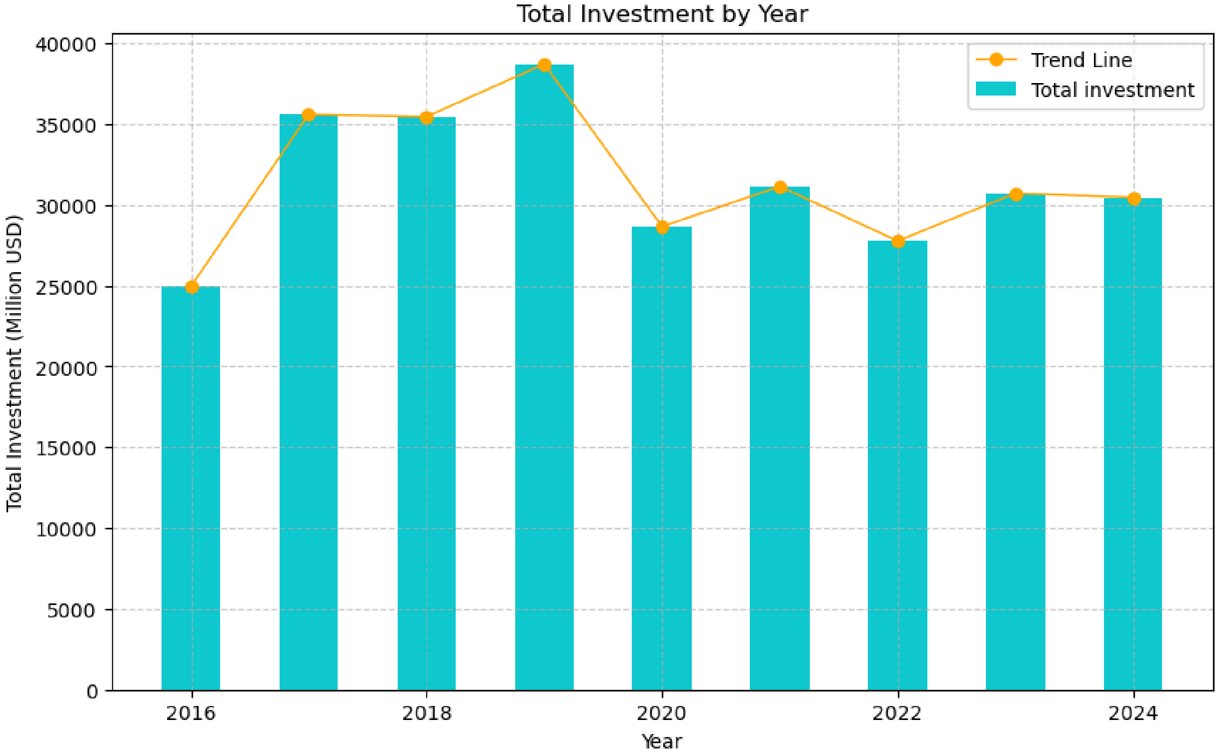
	Number of new projects	Newly registered capital (million USD)	Adjusted project number	Adjusted capital (million USD)	Number of times of capital contribution to buy shares	Value of capital contribution, share purchase (million USD)	Total investment
Number of new projects	1	0.51	0.74	0.57	0.91	0.58	0.57
Newly registered capital (million USD)	0.51	1	0.91	0.96	0.19	0.75	0.99
Adjusted project number	0.74	0.91	1	0.97	0.44	0.71	0.95
Adjusted capital (million USD)	0.57	0.96	0.97	1	0.23	0.73	0.98
Number of times of capital contribution to buy shares	0.91	0.19	0.44	0.23	1	0.36	0.24
Value of capital contribution, share purchase (million USD)	0.58	0.75	0.71	0.73	0.36	1	0.8
Total investment	0.57	0.99	0.95	0.98	0.24	0.8	1

Forecast

In [ ]:

```
# Create model
X = total_investment_by_year[['Year']].values
y = total_investment_by_year['Total investment'].values
model = LinearRegression()
model.fit(X, y)
# Predict the total investment for the next 2 years
future_years = np.array([[2023], [2024]])
predictions = model.predict(future_years)
# Create new df for forecast
forecast_df = pd.concat([
    total_investment_by_year,
    pd.DataFrame({'Year': [2023, 2024], 'Total investment': predictions})
], ignore_index=True)
# Print the forecast
for year, prediction in zip([2023, 2024], predictions):
    print(f'Total investment forecast for {year}: {prediction:.2f} million USD')
# Plot
plt.figure(figsize=(10, 6))
plt.bar(forecast_df['Year'], forecast_df['Total investment'], color='#10c8ce', width=0.5, label='Total investment')
plt.plot(forecast_df['Year'], forecast_df['Total investment'], color='orange', marker='o', linewidth=1, label='Trend Line')
plt.xlabel('Year')
plt.ylabel('Total Investment (Million USD)')
plt.title('Total Investment by Year')
plt.legend()
plt.grid(True, linestyle='--', alpha=0.6)
plt.show()
```

```
Total investment forecast for 2023: 30730.11 million USD
Total investment forecast for 2024: 30471.11 million USD
```



Without Sklearn

Fomula:

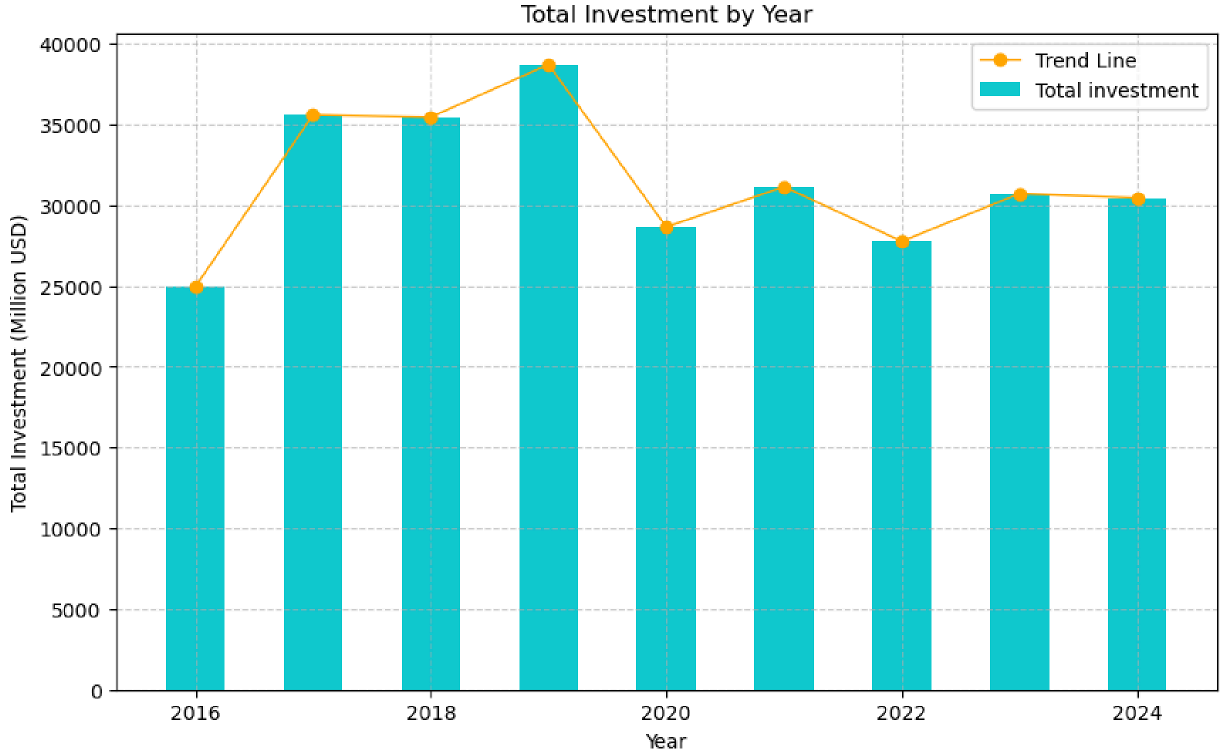
fomula

Detail fomula:

detail detail2

```
In [ ]: # Create X Matrix n × (d + 1)
X1 = np.array([[1, year] for year in total_investment_by_year['Year']])
# Vector Yan
y_value = total_investment_by_year['Total investment'].values
# Caculate X^T*X matrix
XtX = np.dot(X1.T, X1)
# Caculate (X^T*X)^(-1) matrix
XtX_inv = np.linalg.inv(XtX)
# β = (X'X)^(-1) * X'y
beta = np.dot(np.dot(XtX_inv, X1.T), y_value)
print("beta:", beta)
# Predict the total investment for the next 2 years
# Create new data for the next 2 years
new_data = np.array([[1, 2023], [1, 2024]])
# Caculate the predicted values
predicted_values = np.dot(new_data, beta)
# Create a DataFrame for the forecast
forecast_o = pd.DataFrame({'Year': [2023, 2024], 'Total investment': predicted_values})
# Combine the original data with the forecast data
combined_data = pd.concat([total_investment_by_year, forecast_o], ignore_index=True)
# Print the forecast values
for year, prediction in zip([2023, 2024], predicted_values):
    print(f'Total investment forecast for {year}: {prediction:.2f} million USD')
# Plot
plt.figure(figsize=(10, 6))
plt.bar(combined_data['Year'], combined_data['Total investment'], color='#10c8ce', width=0.5, label='Total investment')
plt.plot(combined_data['Year'], combined_data['Total investment'], color='orange', marker='o', linewidth=1, label='Trend Lin
plt.xlabel('Year')
plt.ylabel('Total Investment (Million USD)')
plt.title('Total Investment by Year')
plt.legend()
plt.grid(True, linestyle='--', alpha=0.6)
plt.show()
```

```
beta: [ 5.54683497e+05 -2.58998214e+02]
Total investment forecast for 2023: 30730.11 million USD
Total investment forecast for 2024: 30471.11 million USD
```



NOTE

- According to VIOIT, the actual Total investment figure for 2023 is $36.607,566 billion.
- The FDI in Vietnam for the first 6 months of 2024 is $15.2 billion, as reported by MINISTRY OF PLANNING AND INVESTMENT .
- There is a significant discrepancy of up to $6 billion in the forecast, and as 2024 has not yet concluded, no final conclusions can be drawn. The accuracy of the values and labels depends heavily on the number of samples; with fewer data points from different years, the precision may be compromised.