

dataanalyse_Zephyr

December 16, 2023

```
[33]: import numpy as np
import pandas as pd
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.graph_objects as go
import statsmodels.api as sm

#
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False #

import warnings
warnings.filterwarnings("ignore", category=UserWarning)

import matplotlib.pyplot as plt
from matplotlib.font_manager import FontProperties
#
font = FontProperties(fname=r'C:\WINDOWS\Fonts\simhei.ttf') #

[2]: file_path = r'G:\data\Zephyr      \workfile.xlsx'
data = pd.read_excel(file_path)
#print(data.head())

[3]: #
for column in data.columns:
    print(f"      : {column},      : {data[column].dtype}")

      : Unnamed: 0,      : float64
      : Deal Number,      : int64
      : Acquiror name,      : object
      : Acquiror country code,      : object
      : Target name,      : object
      : Target country code,      : object
      : Deal type,      : object
      : Deal status,      : object
      : Deal value th EUR,      : object
```

```

: Target business description(s), : object
: Acquiror business description(s), : object
: Deal type.1, : object
: Deal sub-type, : object
: Deal financing, : object
: Deal method of payment, : object
: Deal method of payment value th EUR, : object
: Deal status.1, : object
: Rumour date, : datetime64[ns]
: Announced date, : datetime64[ns]
: Expected completion date, : datetime64[ns]
: Assumed completion date, : datetime64[ns]
: Completed date, : datetime64[ns]
: Postponed date, : datetime64[ns]
: Withdrawn date, : datetime64[ns]
: Last deal status date, : datetime64[ns]
: Last deal value, offer price, bid premium update date, : datetime64[ns]
: Last deal status update date, : datetime64[ns]
: Last % of stake update date, : datetime64[ns]
: Last acquiror, target, vendor update date, : datetime64[ns]
: Last advisor update date, : datetime64[ns]
: Last deal comment, rationale update date, : datetime64[ns]
: Last update, : datetime64[ns]
: Deal value th EUR.1, : object
: Deal value (Native currency) th LCU, : object
: Deal equity value th EUR, : object
: Deal equity value (Native currency) th LCU, : object
: Deal enterprise value th EUR, : object
: Deal enterprise value (Native currency) th LCU, : object
: Deal modelled enterprise value th EUR, : object
: Deal modelled enterprise value (Native currency) th LCU, : object
: Deal total target value th EUR, : object
: Deal total target value (Native currency) th LCU, : object
: Modelled Fee Income th EUR, : object
: As Reported Fee Income th EUR, : object
: Initial stake (%), : object
: Acquired stake (%), : object
: Final stake (%), : object
: IRR (%), : float64
: Native currency, : object

```

```

[4]: # "Acquiror country code"
acquiror_country_code_counts = data['Acquiror country code'].value_counts()

#
print("Acquiror country code")
for code, count in acquiror_country_code_counts.items():

```

```
print(f" : {code},      : {count}")
```

Acquiror country code

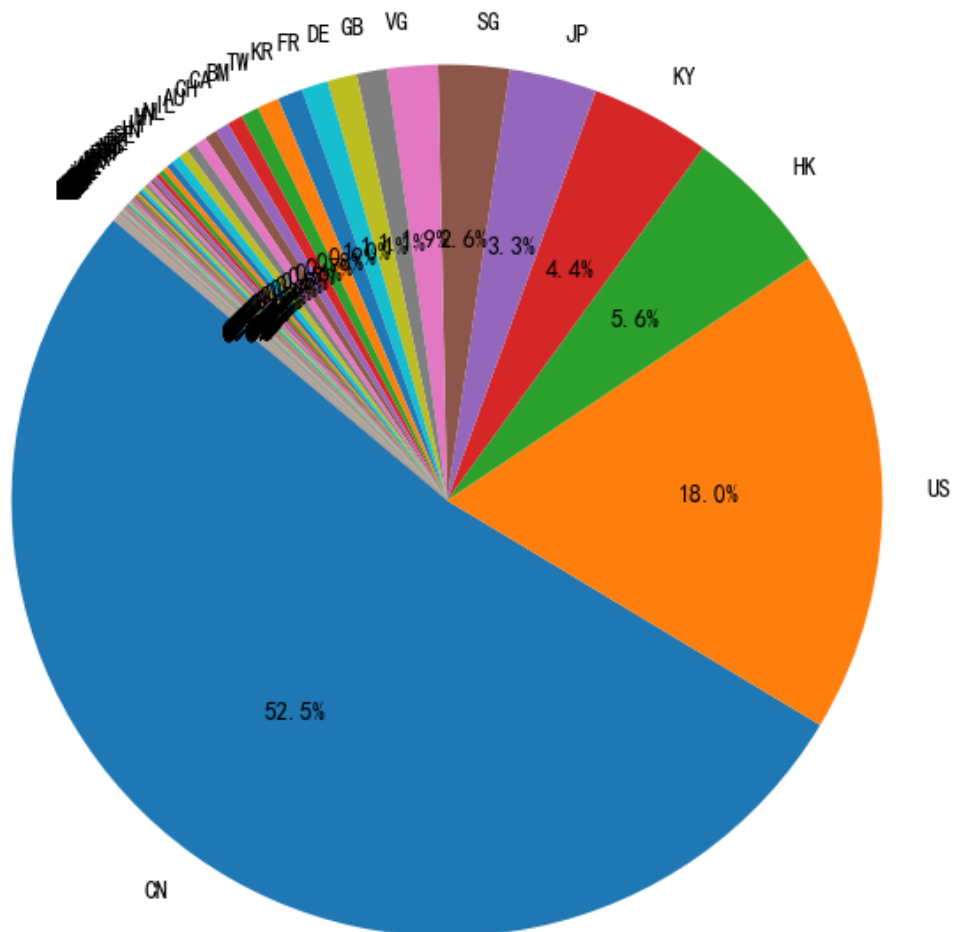
```
: CN,      : 12288
: US,      : 4212
: HK,      : 1315
: KY,      : 1041
: JP,      : 761
: SG,      : 616
: VG,      : 441
: GB,      : 262
: DE,      : 257
: FR,      : 230
: KR,      : 218
: TW,      : 190
: BM,      : 155
: CA,      : 136
: CH,      : 118
: AU,      : 109
: IL,      : 99
: NL,      : 89
: MY,      : 84
: IT,      : 80
: IN,      : 59
: SE,      : 45
: FI,      : 44
: BE,      : 39
: ES,      : 35
: NO,      : 29
: LU,      : 29
: TH,      : 29
: RU,      : 28
: BR,      : 27
: MU,      : 25
: DK,      : 21
: II,      : 19
: UZ,      : 15
: KZ,      : 14
: BY,      : 14
: CZ,      : 14
: WS,      : 13
: IE,      : 13
: AT,      : 13
: VN,      : 13
: SA,      : 9
: ID,      : 8
: ZA,      : 8
: HU,      : 8
```

: GR,	: 7
: MO,	: 6
: PT,	: 6
: PL,	: 6
: PK,	: 6
: AR,	: 6
: CL,	: 6
: CO,	: 5
: EE,	: 5
: ZW,	: 5
: EG,	: 4
: NG,	: 4
: AE,	: 4
: PH,	: 3
: CU,	: 3
: BG,	: 3
: AM,	: 3
: NZ,	: 3
: CW,	: 3
: TR,	: 3
: UA,	: 2
: LT,	: 2
: IR,	: 2
: CY,	: 2
: CD,	: 2
: SC,	: 2
: KH,	: 2
: QA,	: 2
: DZ,	: 2
: MA,	: 2
: SI,	: 2
: HR,	: 2
: SK,	: 1
: UY,	: 1
: OM,	: 1
: TJ,	: 1
: GN,	: 1
: GE,	: 1
: MD,	: 1
: BH,	: 1
: MX,	: 1
: PA,	: 1
: KW,	: 1
: KN,	: 1
: VE,	: 1
: SL,	: 1
: ZM,	: 1
: LI,	: 1

```
: MN, : 1
: UG, : 1
: PE, : 1
: RS, : 1
: BD, : 1
```

```
[5]: #
plt.figure(figsize=(8, 8))
plt.pie(acquiror_country_code_counts, labels=acquiror_country_code_counts.
        index, autopct='%1.1f%%', startangle=140)
plt.title('Acquiror country code ')
plt.show()
```

Acquiror country code 分布饼状图



```

[6]: # "Acquiror country code"
acquiror_country_code_counts = data['Acquiror country code'].value_counts()

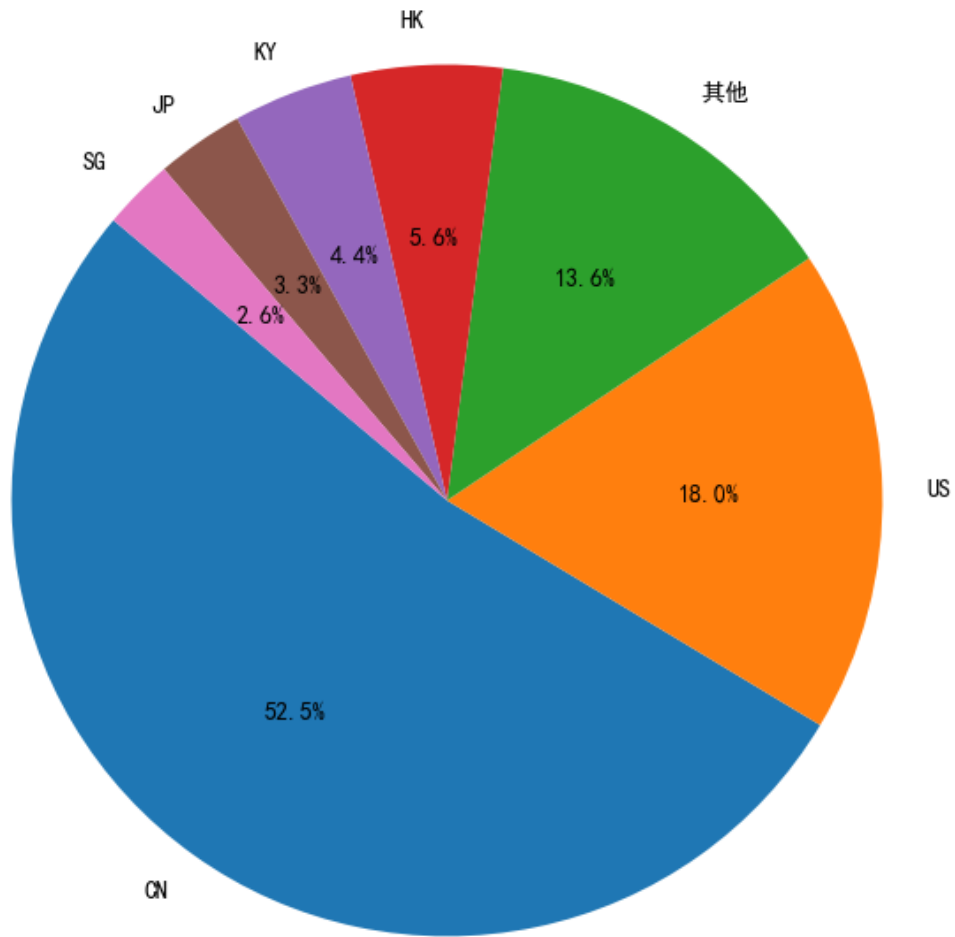
# 2.6%
threshold = 0.026
other_codes = acquiror_country_code_counts[acquiror_country_code_counts /
↳acquiror_country_code_counts.sum() < threshold].index
data['Acquiror country code'] = data['Acquiror country code'].
↳replace(other_codes, ' ')

#
acquiror_country_code_counts = data['Acquiror country code'].value_counts()

#
plt.figure(figsize=(8, 8))
plt.pie(acquiror_country_code_counts, labels=acquiror_country_code_counts.
↳index, autopct='%1.1f%%', startangle=140)
plt.title('Acquiror country code')
plt.show()

```

Acquiror country code 分布饼状图



```
[7]: deal_value_column = data['Deal value th EUR']  
print(f"Deal value th EUR      {deal_value_column.dtype}")
```

Deal value th EUR object

```
[8]: data['Deal value th EUR'] = pd.to_numeric(data['Deal value th EUR'],  
      ↪errors='coerce')  
#  
print(f"Deal value th EUR      {data['Deal value th EUR'].dtype}")
```

Deal value th EUR float64

```
[9]: deal_value_column = data['Rumour date']
print(f"Rumour date      {deal_value_column.dtype}")
```

Rumour date datetime64[ns]

```
[10]: # "Rumour date"
rumour_date_summary = data['Rumour date'].describe(datetime_is_numeric=True)
# rumour_date_summary = data['Rumour date'].describe()

#
print("Rumour date      ")
print(rumour_date_summary)
```

```
Rumour date
count                25367
mean    2014-01-22 00:27:11.473962240
min          1995-12-31 00:00:00
25%          2011-01-11 00:00:00
50%          2015-10-27 00:00:00
75%          2017-10-05 00:00:00
max          2019-08-27 00:00:00
Name: Rumour date, dtype: object
```

```
[11]: # "Rumour date" "Acquiror country code" "Deal value th EUR"
deal_value_sum_by_group = data.groupby(['Rumour date', 'Acquiror country_
↵code'])['Deal value th EUR'].sum()

#
print(" Rumour date  Acquiror country code  Deal value th EUR  ")
print(deal_value_sum_by_group)
```

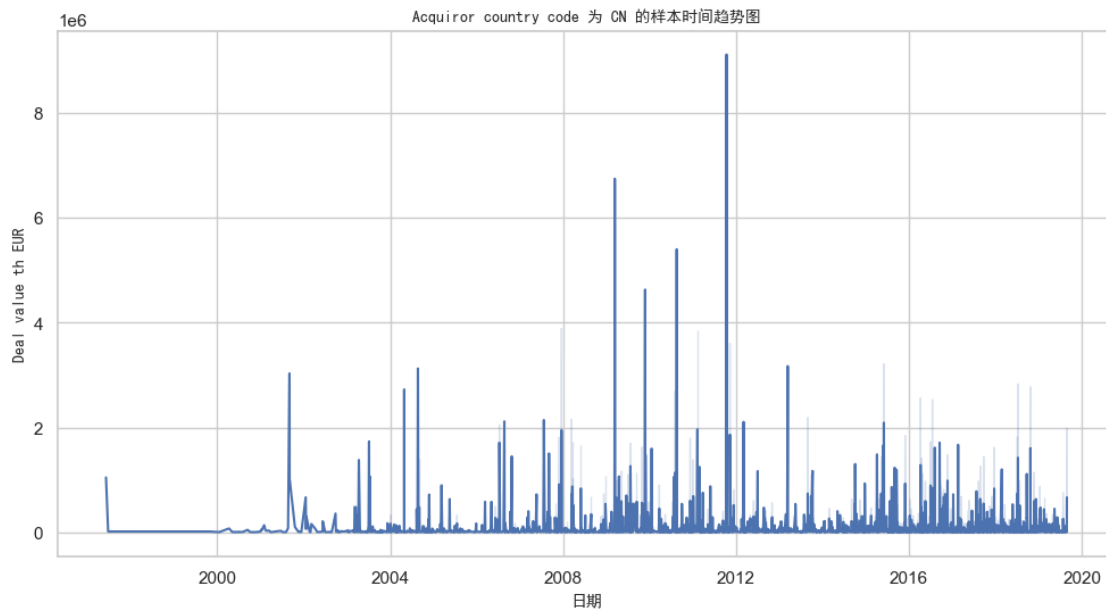
```
 Rumour date  Acquiror country code  Deal value th EUR
Rumour date  Acquiror country code
1995-12-31    US                      0.00
1997-02-28    CN                      0.00
1997-06-06    CN          1043196.93
1997-06-24    CN          12158.26
1998-02-04    CN                      0.00
...
2019-08-24    US                      0.00
2019-08-26    CN          25983.52
              US                      0.00
2019-08-27    CN          1991362.14
              97825.48
Name: Deal value th EUR, Length: 8937, dtype: float64
```

```
[12]: # "Acquiror country code" "CN"
cn_samples = data[data['Acquiror country code'] == 'CN']
```



```
#
sns.set(style="whitegrid")

#
plt.figure(figsize=(12, 6))
sns.lineplot(x='Rumour date', y='Deal value th EUR', data=cn_samples)
plt.title('Acquiror country code CN', fontproperties=font)
plt.xlabel('', fontproperties=font)
plt.ylabel('Deal value th EUR', fontproperties=font)
plt.show()
```



```
[13]: ### tabulate ###
# "Rumour date" "Deal value th EUR"
deal_value_sum_by_year = data.groupby([data['Rumour date'].dt.year, 'Acquiror_
↳country code'])['Deal value th EUR'].sum()
# : deal_value_sum_by_group = data.groupby(['Rumour date', 'Acquiror country_
↳code'])['Deal value th EUR'].sum()

#
print(" Rumour date Acquiror country code Deal value th EUR ")
print(deal_value_sum_by_year)

### ###
# "Acquiror country code" "CN"
```

```

cn_samples = data[data['Acquiror country code'] == 'CN']

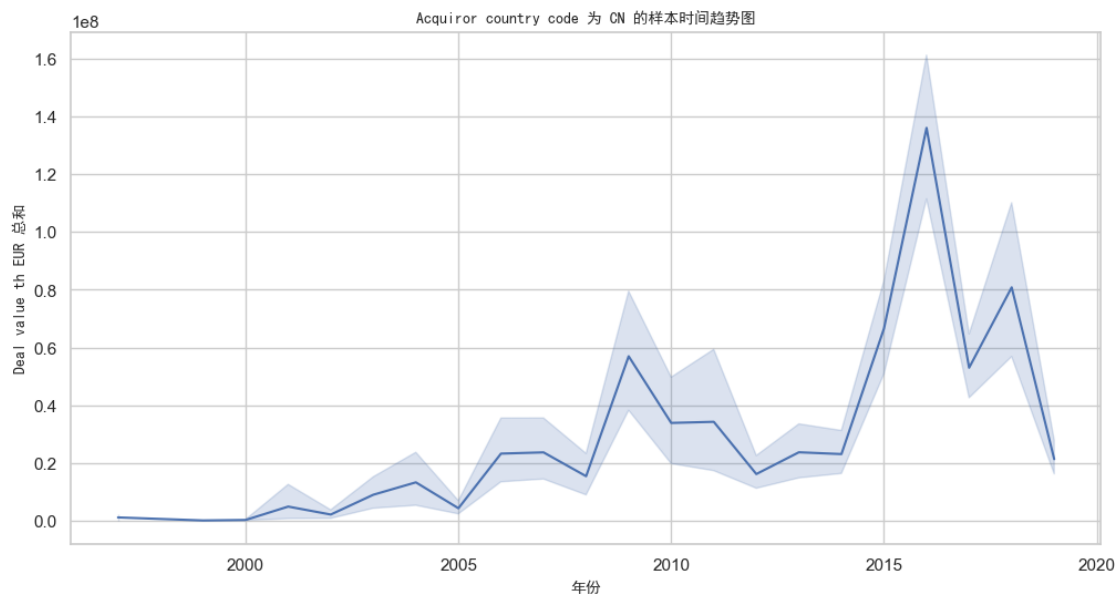
#
sns.set(style="whitegrid")

# "Acquiror country code" "CN"
plt.figure(figsize=(12, 6))
sns.lineplot(x=cn_samples['Rumour date'].dt.year, y=cn_samples['Deal value th_
EUR'], estimator='sum')
plt.title('Acquiror country code CN ', fontproperties=font)
plt.xlabel(' ', fontproperties=font)
plt.ylabel('Deal value th EUR ', fontproperties=font)
plt.show()

```

Rumour date	Acquiror country code	Deal value th EUR
Rumour date	Acquiror country code	
1995	US	0.00
1997	CN	1055355.19
1998	US	0.00
		0.00
1999	CN	11489.88
		...
2019	JP	1453527.09
	KY	135947.13
	SG	1986665.50
	US	4812972.23
		13771393.89

Name: Deal value th EUR, Length: 142, dtype: float64

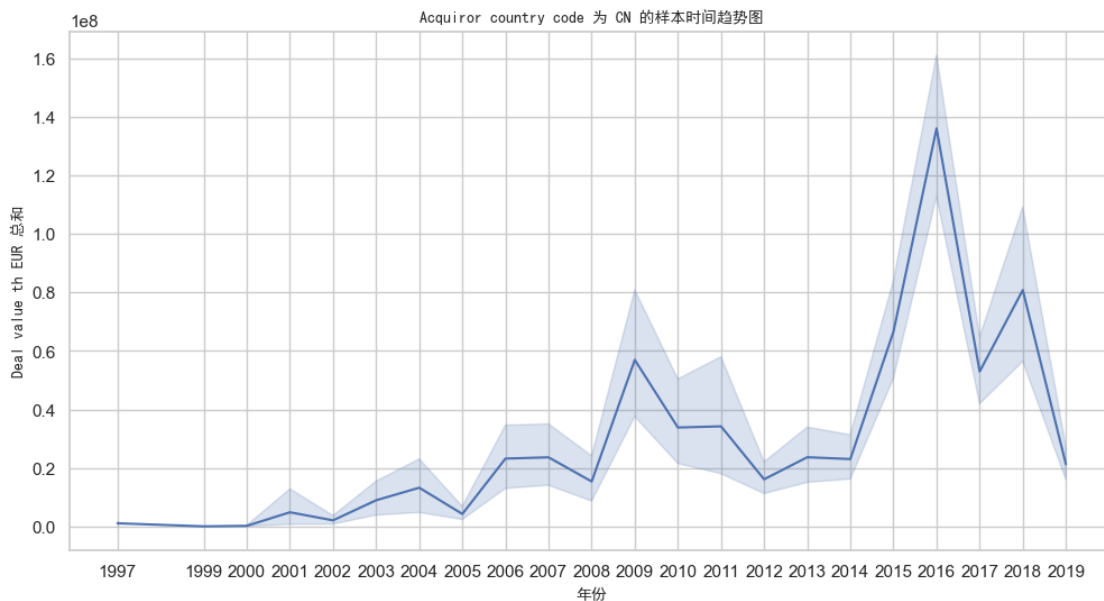


```
[14]: #
sns.set(style="whitegrid")

# "Acquiror country code" "CN"
plt.figure(figsize=(12, 6))
plot = sns.lineplot(x=cn_samples['Rumour date'].dt.year, y=cn_samples['Deal value th EUR'], estimator='sum')
plt.title('Acquiror country code CN ', fontproperties=font)
plt.xlabel(' ', fontproperties=font)
plt.ylabel('Deal value th EUR ', fontproperties=font)

# x
years = sorted(cn_samples['Rumour date'].dt.year.unique())
plot.set_xticks(years)

plt.show()
```



```
[15]: # "Acquiror country code" "CN" "US"
cn_samples = data[data['Acquiror country code'] == 'CN']
us_samples = data[data['Acquiror country code'] == 'US']

#
sns.set(style="whitegrid")

# "Acquiror country code" "CN" "US"
```

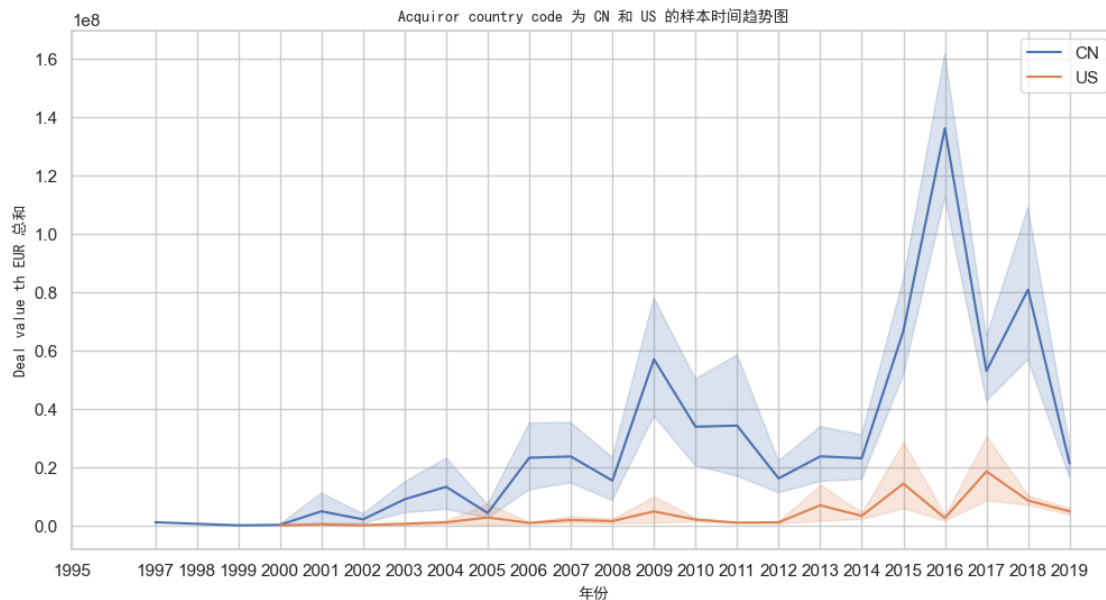
```

plt.figure(figsize=(12, 6))
plot = sns.lineplot(x=cn_samples['Rumour date'].dt.year, y=cn_samples['Deal value th EUR'], estimator='sum', label='CN')
sns.lineplot(x=us_samples['Rumour date'].dt.year, y=us_samples['Deal value th EUR'], estimator='sum', label='US')
plt.title('Acquiror country code CN US ', fontproperties=font)
plt.xlabel(' ', fontproperties=font)
plt.ylabel('Deal value th EUR ', fontproperties=font)
plt.legend()

# x
years = sorted(data['Rumour date'].dt.year.unique())
plot.set_xticks(years)

plt.show()

```



```

[16]: # "Acquiror country code" "CN" "US"
cn_samples = data[data['Acquiror country code'] == 'CN']
us_samples = data[data['Acquiror country code'] == 'US']

#
sns.set(style="whitegrid")

# "Acquiror country code" "CN"
plt.figure(figsize=(12, 6))

```

```

plot_cn = sns.lineplot(x=cn_samples['Rumour date'].dt.year, y=cn_samples['Deal_
↳value th EUR'], estimator='sum', label='CN', ci = None )

#           "Acquiror country code"   "US"
plot_us = plot_cn.twinx()
sns.lineplot(x=us_samples['Rumour date'].dt.year, y=us_samples['Deal value th_
↳EUR'], estimator='sum', color='orange', label='US', ci = None )

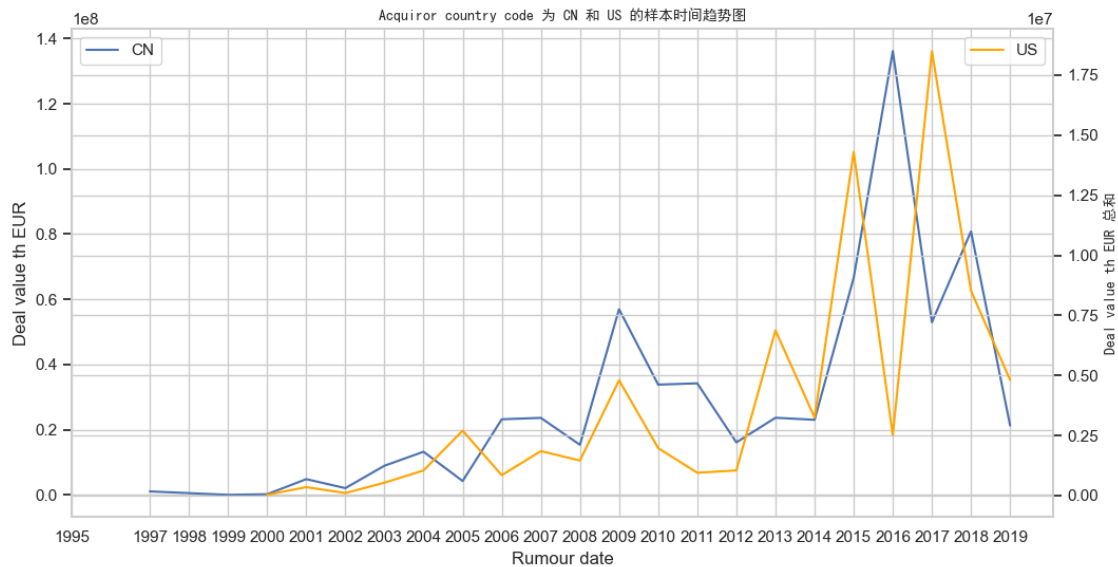
#
plt.title('Acquiror country code   CN   US   ', fontproperties=font)
plt.xlabel(' ', fontproperties=font)
plt.ylabel('Deal value th EUR ', fontproperties=font)

#   x
years = sorted(data['Rumour date'].dt.year.unique())
plot_cn.set_xticks(years)

#
plot_cn.legend(loc='upper left')
plot_us.legend(loc='upper right')

plt.show()

```



```

[17]: #
sns.set(style="whitegrid")

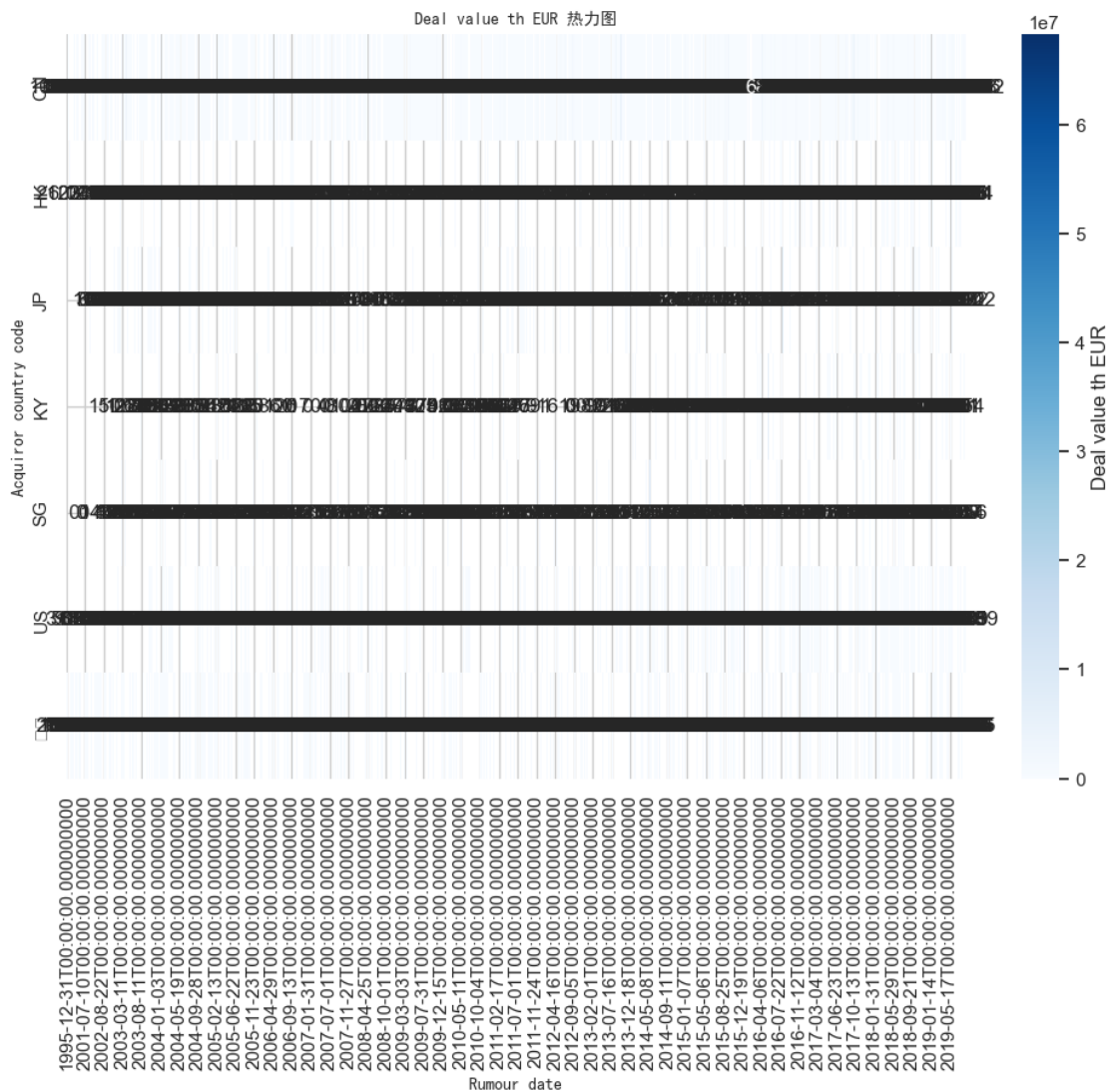
#

```

```
heatmap_data = data.pivot_table(values='Deal value th EUR', index='Acquiror_
↳country code', columns='Rumour date', aggfunc='sum')

#
plt.figure(figsize=(12, 8))
sns.heatmap(heatmap_data, cmap='Blues', annot=True, fmt='.0f',
↳cbar_kws={'label': 'Deal value th EUR'})
plt.title('Deal value th EUR ', fontproperties=font)
plt.xlabel('Rumour date', fontproperties=font)
plt.ylabel('Acquiror country code', fontproperties=font)

plt.show()
```



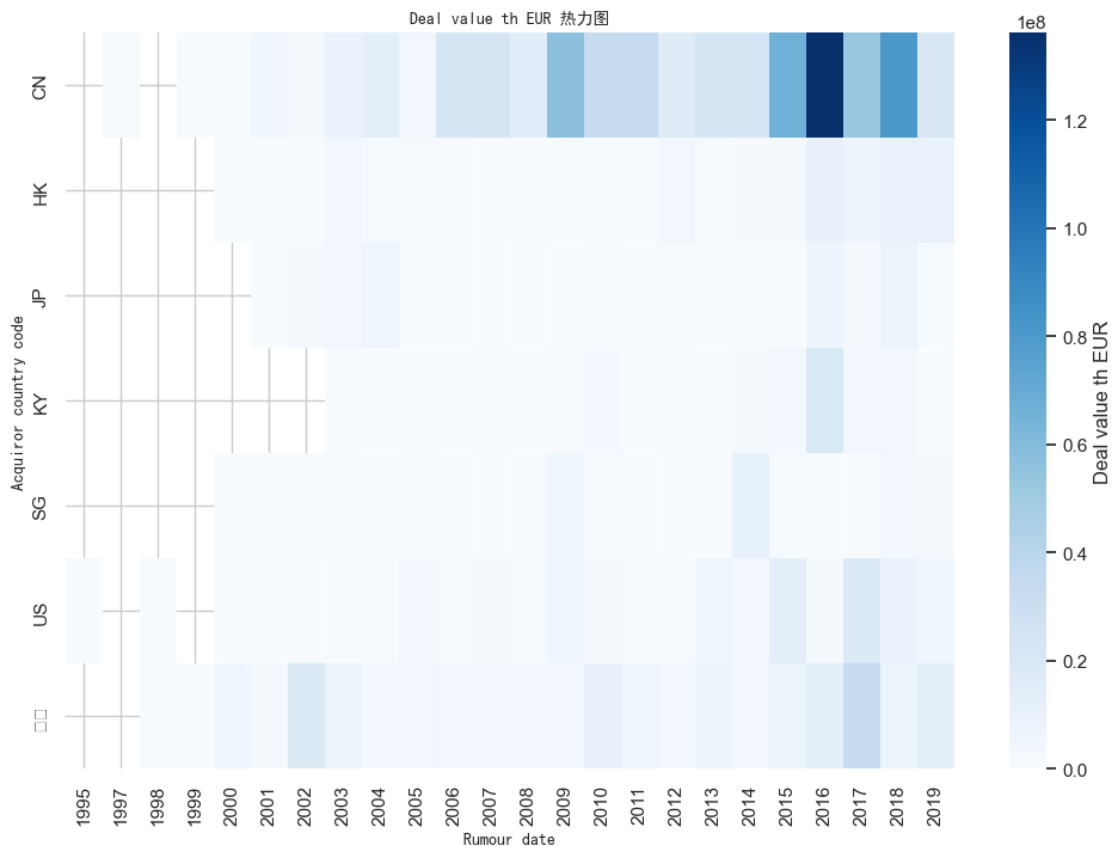
```
[18]: # "Rumour date"          "Deal value th EUR"
grouped_data = data.groupby([data['Rumour date'].dt.year, 'Acquiror country_
↳code'])['Deal value th EUR'].sum().reset_index()

#
heatmap_data = grouped_data.pivot_table(values='Deal value th EUR',
↳index='Acquiror country code', columns='Rumour date', aggfunc='sum')

#
sns.set(style="whitegrid")

#
plt.figure(figsize=(12, 8))
sns.heatmap(heatmap_data, cmap='Blues', fmt='.0f', cbar_kws={'label': 'Deal_
↳value th EUR'}) #annot=True,
plt.title('Deal value th EUR ', fontproperties=font)
plt.xlabel('Rumour date', fontproperties=font)
plt.ylabel('Acquiror country code', fontproperties=font)

plt.show()
```



```
[19]: # "Rumour date"
data['Rumour year'] = data['Rumour date'].dt.year

# "Deal value th EUR"
result = data.groupby(['Acquiror country code', 'Target country code', 'Rumour_
    year'])['Deal value th EUR'].sum().reset_index()

#
print(result)
```

	Acquiror country code	Target country code	Rumour year	Deal value th EUR
0	CN	AE	2005	0.00
1	CN	AE	2006	0.00
2	CN	AE	2007	0.00
3	CN	AE	2011	0.00
4	CN	AE	2015	1891.49
...
1821		ZM	2003	0.00
1822		ZM	2004	0.00
1823		ZM	2016	0.00
1824		ZW	2006	0.00
1825		ZW	2008	0.00

[1826 rows x 4 columns]

```
[20]: # "Target country code"
def map_to_region(country_code):
    #
    if country_code in ['US']:
        return ' '
    elif country_code in ['FR', 'DE', 'GB', 'IT', 'ES']:
        return ' '
    elif country_code in ['AU', 'NZ']:
        return ' '
    elif country_code in ['SG', 'MY', 'TH', 'ID', 'VN', 'PH']:
        return ' '
    elif country_code in ['SA', 'AE', 'IL']:
        return ' '
    elif country_code in ['ZA', 'NG', 'KE']:
        return ' '
    else:
        return ' '

# "dest"
data['dest'] = data['Target country code'].apply(map_to_region)

#
```



```

sankey_data = data.groupby(['Acquiror country code', 'dest']).agg({'Deal value_
    th EUR': 'sum'}).reset_index()

# plotly
fig = go.Figure(data=[go.Sankey(
    node=dict(
        pad=15,
        thickness=20,
        line=dict(color="black", width=0.5),
        label=sankey_data['Acquiror country code'].append(sankey_data['dest']).
        unique(),
    ),
    link=dict(
        source=sankey_data['Acquiror country code'].map(lambda x:
        list(sankey_data['Acquiror country code'].unique()).index(x)),
        target=sankey_data['dest'].map(lambda x: len(sankey_data['Acquiror_
        country code'].unique()) + list(sankey_data['dest'].unique()).index(x)),
        value=sankey_data['Deal value th EUR']
    )
)])

#
fig.update_layout(title_text="Sankey Diagram", font_size=10)
fig.show()

```

C:\Users\Allen\AppData\Local\Temp\ipykernel_2748\2889772019.py:31:

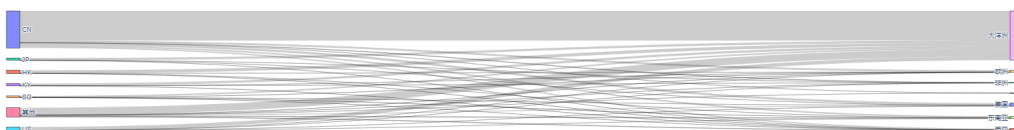
FutureWarning: The series.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.

```

label=sankey_data['Acquiror country
code'].append(sankey_data['dest']).unique(),

```

Sankey Diagram



```

[21]: # "dest" " "
oceanica_data = data[data['dest'] != ' ']

#
sankey_data = oceanica_data.groupby(['Acquiror country code', 'dest']).
    agg({'Deal value th EUR': 'sum'}).reset_index()

```

```

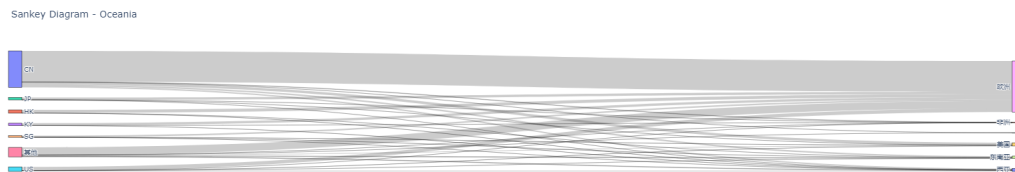
# plotly
fig = go.Figure(data=[go.Sankey(
    node=dict(
        pad=15,
        thickness=20,
        line=dict(color="black", width=0.5),
        label=sankey_data['Acquiror country code'].append(sankey_data['dest']).
        ↪unique(),
    ),
    link=dict(
        source=sankey_data['Acquiror country code'].map(lambda x: ↪
        ↪list(sankey_data['Acquiror country code'].unique()).index(x)),
        target=sankey_data['dest'].map(lambda x: len(sankey_data['Acquiror ↪
        ↪country code'].unique()) + list(sankey_data['dest'].unique()).index(x)),
        value=sankey_data['Deal value th EUR']
    )
)])

#
fig.update_layout(title_text="Sankey Diagram - Oceania", font_size=10)
fig.show()

```

C:\Users\Allen\AppData\Local\Temp\ipykernel_2748\36876018.py:13: FutureWarning:

The series.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.



```

[26]: # 'Deal value th EUR'
data['Log_Deal_Value'] = np.log(data['Deal value th EUR'])

#
sns.set(style="whitegrid")

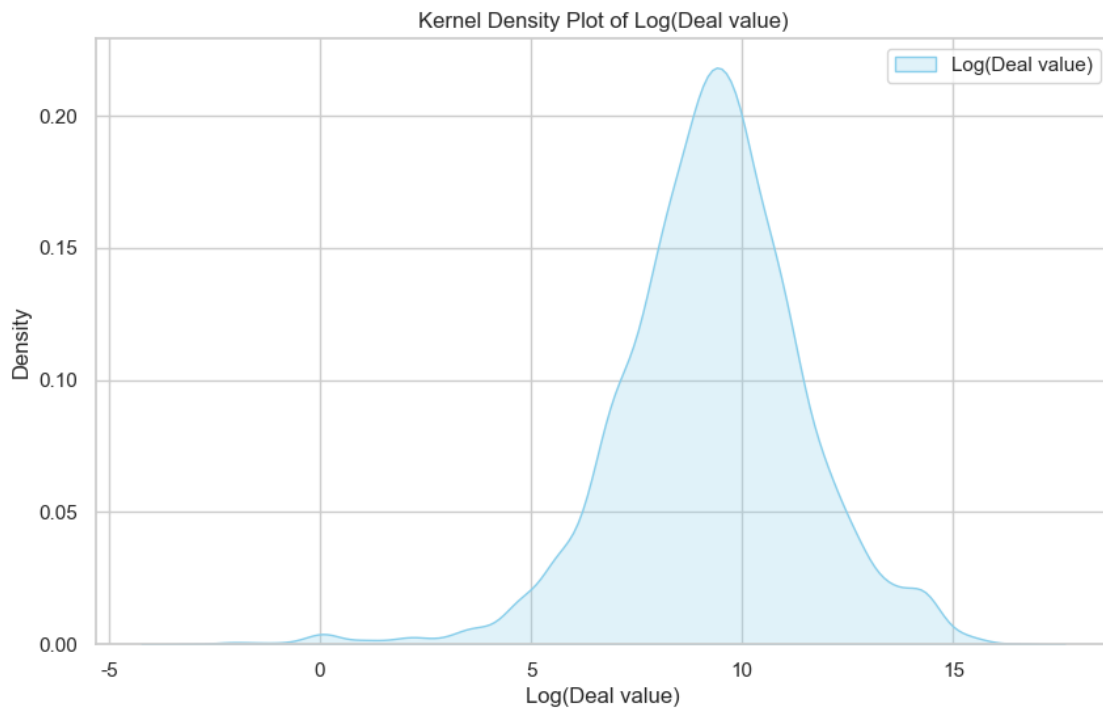
#
plt.figure(figsize=(10, 6))

```

```

sns.kdeplot(data['Log_Deal_Value'], fill=True, color='skyblue', label='Log(Deal_
↵value)')
plt.title('Kernel Density Plot of Log(Deal value)')
plt.xlabel('Log(Deal value)')
plt.ylabel('Density')
plt.legend()
plt.show()

```



```

[28]: # CN US
cn_data = data[data['Acquiror country code'] == 'CN']
us_data = data[data['Acquiror country code'] == 'US']

# 'Deal value th EUR'
data['Log_Deal_Value'] = np.log(data['Deal value th EUR'])

#
sns.set(style="whitegrid")

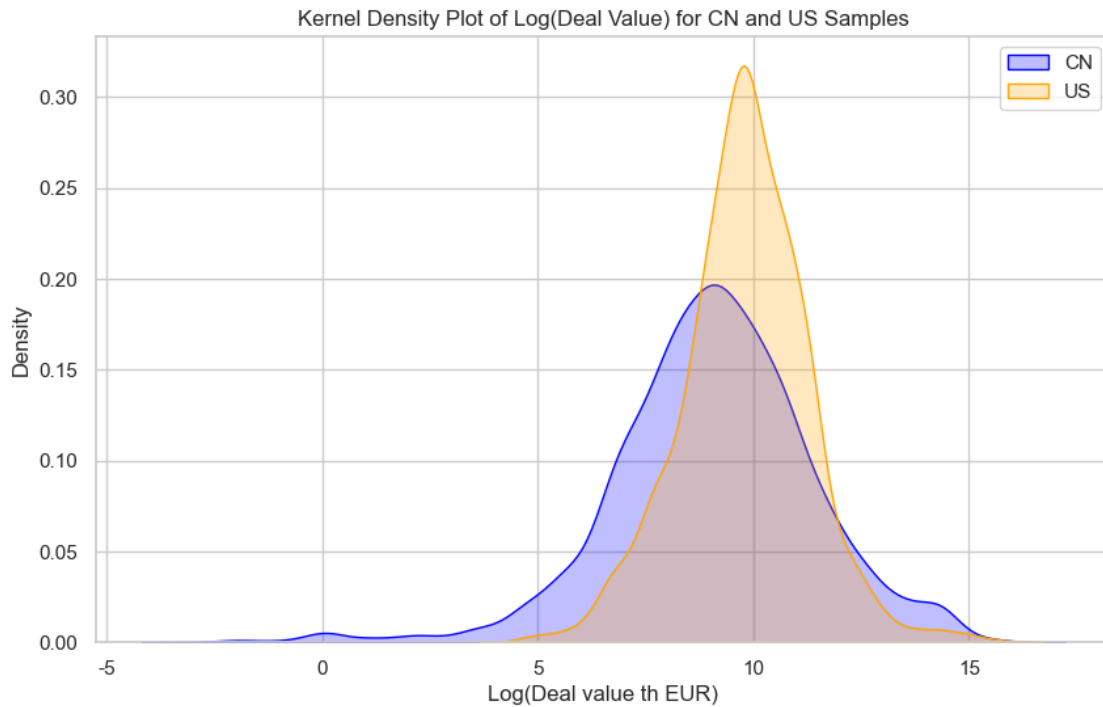
#
plt.figure(figsize=(10, 6))

# CN
sns.kdeplot(cn_data['Log_Deal_Value'], fill=True, label='CN', color='blue')

```

```
# US
sns.kdeplot(us_data['Log_Deal_Value'], fill=True, label='US', color='orange')

plt.title('Kernel Density Plot of Log(Deal Value) for CN and US Samples')
plt.xlabel('Log(Deal value th EUR)')
plt.ylabel('Density')
plt.legend()
plt.show()
```



```
[29]: # CN US
cn_data = data[data['Acquiror country code'] == 'CN']
us_data = data[data['Acquiror country code'] == 'US']

#
sns.set(style="whitegrid")

#
fig, axes = plt.subplots(1, 2, figsize=(12, 6))

# CN
sns.kdeplot(cn_data['Log_Deal_Value'], fill=True, label='CN', color='blue',
            ax=axes[0])
axes[0].set_title('Kernel Density Plot for CN Samples')
axes[0].set_xlabel('Log(Deal value th EUR)')
```

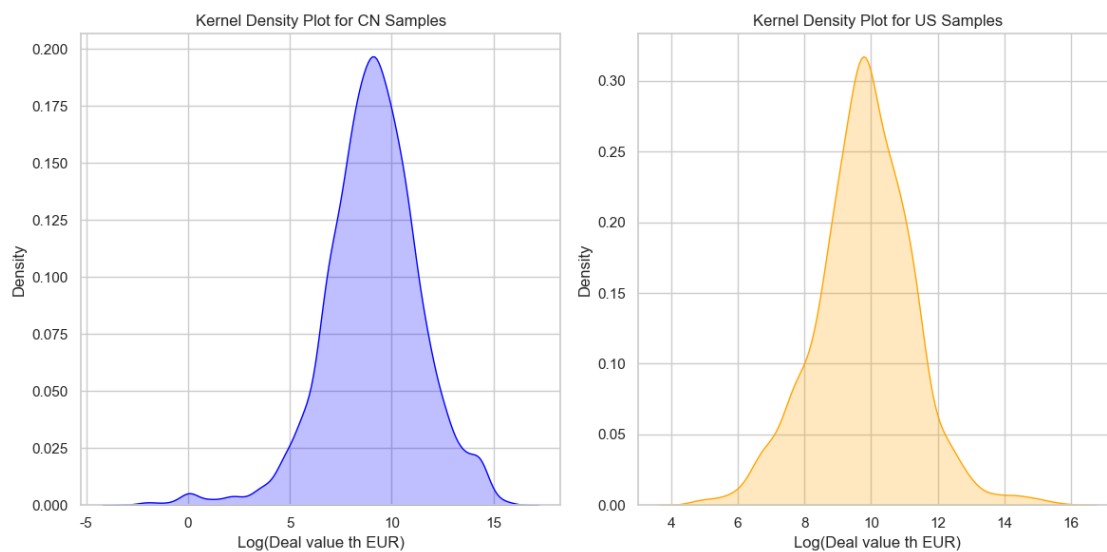
```

axes[0].set_ylabel('Density')

# US
sns.kdeplot(us_data['Log_Deal_Value'], fill=True, label='US', color='orange',
            ax=axes[1])
axes[1].set_title('Kernel Density Plot for US Samples')
axes[1].set_xlabel('Log(Deal value th EUR)')
axes[1].set_ylabel('Density')

#
plt.tight_layout()
plt.show()

```



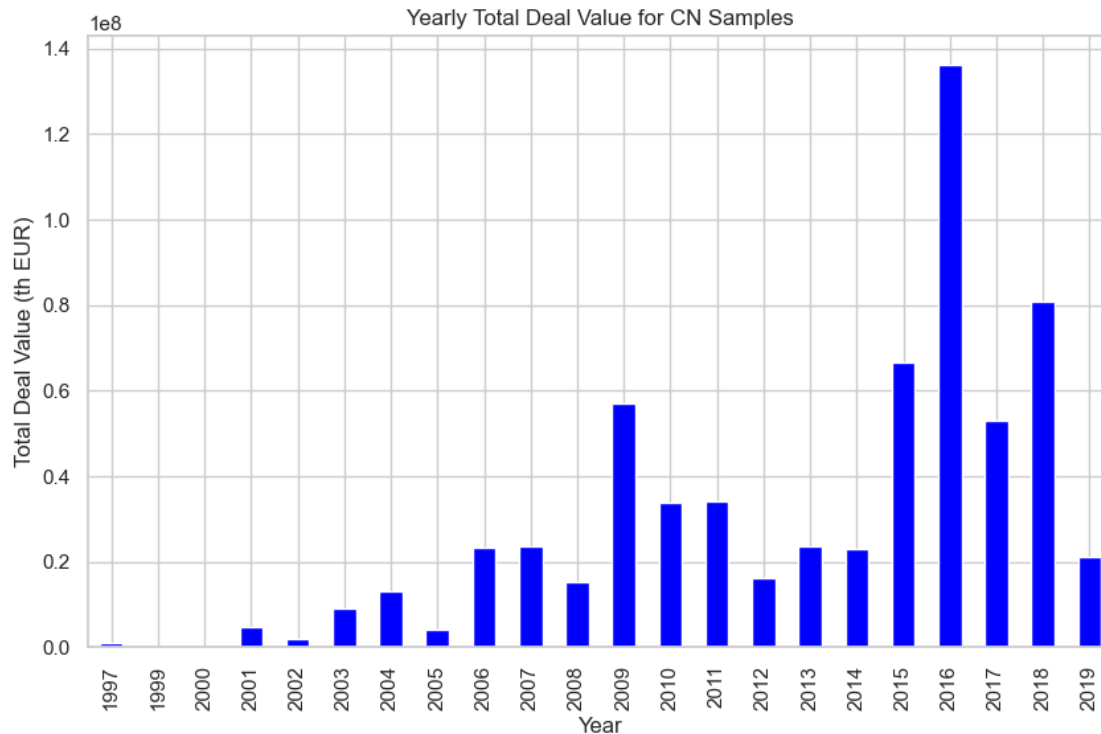
```

[30]: # CN
cn_data = data[data['Acquiror country code'] == 'CN']

# 'Deal value th EUR'
cn_yearly_total = cn_data.groupby(cn_data['Rumour date'].dt.year)['Deal value_
    th EUR'].sum()

# matplotlib
plt.figure(figsize=(10, 6))
cn_yearly_total.plot(kind='bar', color='blue')
plt.title('Yearly Total Deal Value for CN Samples')
plt.xlabel('Year')
plt.ylabel('Total Deal Value (th EUR)')
plt.show()

```

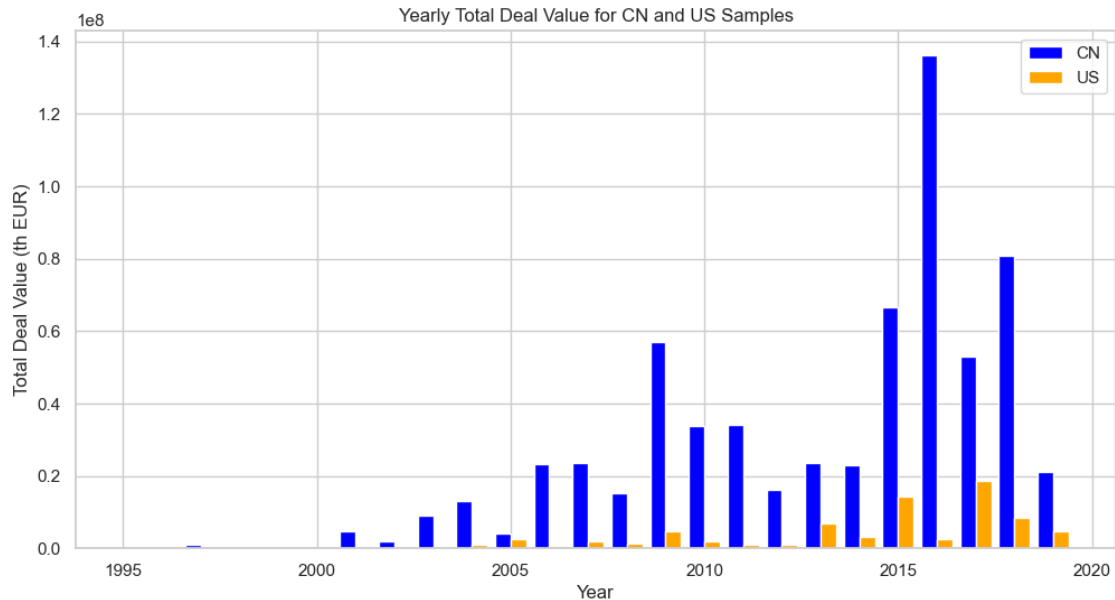


```
[31]: # CN US
cn_data = data[data['Acquiror country code'] == 'CN']
us_data = data[data['Acquiror country code'] == 'US']

# 'Deal value th EUR'
cn_yearly_total = cn_data.groupby(cn_data['Rumour date'].dt.year)['Deal value_
↳th EUR'].sum()
us_yearly_total = us_data.groupby(us_data['Rumour date'].dt.year)['Deal value_
↳th EUR'].sum()

# matplotlib
plt.figure(figsize=(12, 6))
plt.bar(cn_yearly_total.index - 0.2, cn_yearly_total.values, width=0.4,
↳color='blue', label='CN')
plt.bar(us_yearly_total.index + 0.2, us_yearly_total.values, width=0.4,
↳color='orange', label='US')

plt.title('Yearly Total Deal Value for CN and US Samples')
plt.xlabel('Year')
plt.ylabel('Total Deal Value (th EUR)')
plt.legend()
plt.show()
```



```
[32]: # CN US
cn_data = data[data['Acquiror country code'] == 'CN']
us_data = data[data['Acquiror country code'] == 'US']

# 'Deal value th EUR'
cn_yearly_total = cn_data.groupby(cn_data['Rumour date'].dt.year)['Deal value_
    ↳th EUR'].sum()
us_yearly_total = us_data.groupby(us_data['Rumour date'].dt.year)['Deal value_
    ↳th EUR'].sum()

#
fig, axes = plt.subplots(1, 2, figsize=(14, 6))

# CN
axes[0].bar(cn_yearly_total.index, cn_yearly_total.values, color='blue',
    ↳label='CN')
axes[0].set_title('Yearly Total Deal Value for CN Samples')
axes[0].set_xlabel('Year')
axes[0].set_ylabel('Total Deal Value (th EUR)')
axes[0].legend()

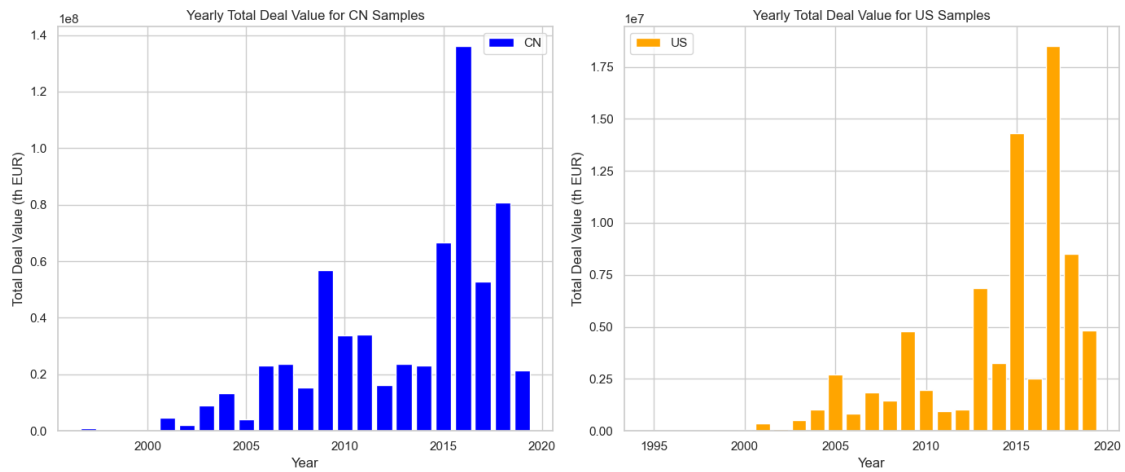
# US
axes[1].bar(us_yearly_total.index, us_yearly_total.values, color='orange',
    ↳label='US')
axes[1].set_title('Yearly Total Deal Value for US Samples')
axes[1].set_xlabel('Year')
```

```

axes[1].set_ylabel('Total Deal Value (th EUR)')
axes[1].legend()

#
plt.tight_layout()
plt.show()

```



[]:

[]:

[]:

[]: