Contents

In [39]:
```
import pandas as pd
import numpy as np
import warnings
import os
warnings.filterwarnings('ignore')
import seaborn as sns
import matplotlib.pyplot as plt
"""
sns 相关设置
@return:
"""
# 声明使用 Seaborn 样式
sns.set()
# 有五种seaborn的绘图风格，它们分别是：darkgrid, whitegrid, dark, white, ticks。默认的主题是darkgrid。
sns.set_style("whitegrid")
# 有四个预置的环境，按大小从小到大排列分别为：paper, notebook, talk, poster。其中，notebook是默认的。
sns.set_context('talk')
# 中文字体设置-黑体
plt.rcParams['font.sans-serif'] = ['Microsoft YaHei']
# 解决保存图像是负号'-'显示为方块的问题
```

1 HFRX

In [16]:
```
hfrx=pd.read_csv('hfrx_daily_index_data.csv')
```

...

In [18]:
```
hfrx.columns=['date','name','index','change','size']
```

...

In [84]:
```
hfrx_Global=hfrx[hfrx['name']=='HFRX Global Hedge Fund Index']
hfrx_Global['date']=pd.to_datetime(hfrx_Global.date)
hfrx_Global.sort_values(by='date',ascending=True,inplace=True)
hfrx_Global.to_csv('Global_HFI.csv',index=False)
```

Out[84]:

	date	name	index	change	size
131266	2003-03-31	HFRX Global Hedge Fund Index	HFRXGL	0%	1000.00
131251	2003-04-01	HFRX Global Hedge Fund Index	HFRXGL	0.20%	1002.00
131235	2003-04-02	HFRX Global Hedge Fund Index	HFRXGL	0.22%	1004.23
131219	2003-04-03	HFRX Global Hedge Fund Index	HFRXGL	0.07%	1004.97
131203	2003-04-04	HFRX Global Hedge Fund Index	HFRXGL	0%	1004.96
...	...	...	...	...	...

	date	name	index	change	size
148	2021-07-16	HFRX Global Hedge Fund Index	HFRXGL	-0.13%	1426.85
116	2021-07-19	HFRX Global Hedge Fund Index	HFRXGL	-0.57%	1418.72
84	2021-07-20	HFRX Global Hedge Fund Index	HFRXGL	0.14%	1420.70
52	2021-07-21	HFRX Global Hedge Fund Index	HFRXGL	0.33%	1425.37
20	2021-07-22	HFRX Global Hedge Fund Index	HFRXGL	0.14%	1427.37

In [83]:

```
hfrx_Equity_Hedge=hfrx[hfrx['name']=='HFRX Equity Hedge Index']
hfrx_Equity_Hedge['date']=pd.to_datetime(hfrx_Equity_Hedge.date)
hfrx_Equity_Hedge.sort_values(by='date',ascending=True,inplace=True)
hfrx_Equity_Hedge.to_csv('Equity_Hedge.csv',index=False)
```

Out[83]:

	date	name	index	change	size
131262	2003-03-31	HFRX Equity Hedge Index	HFRXEH	0%	1000.00
131246	2003-04-01	HFRX Equity Hedge Index	HFRXEH	0.38%	1003.76
131230	2003-04-02	HFRX Equity Hedge Index	HFRXEH	0.45%	1008.31
131214	2003-04-03	HFRX Equity Hedge Index	HFRXEH	0.14%	1009.74
131198	2003-04-04	HFRX Equity Hedge Index	HFRXEH	-0.15%	1008.19
...	...	...	...	...	...
140	2021-07-16	HFRX Equity Hedge Index	HFRXEH	-0.29%	1433.95
108	2021-07-19	HFRX Equity Hedge Index	HFRXEH	-0.93%	1420.68
76	2021-07-20	HFRX Equity Hedge Index	HFRXEH	0.58%	1428.96
44	2021-07-21	HFRX Equity Hedge Index	HFRXEH	0.59%	1437.32
12	2021-07-22	HFRX Equity Hedge Index	HFRXEH	0.19%	1440.10

4618 rows × 5 columns

In [82]:

```
hfrx_ED=hfrx[hfrx['name']=='HFRX Event Driven Index']
hfrx_ED['date']=pd.to_datetime(hfrx_ED.date)
hfrx_ED.sort_values(by='date',ascending=True,inplace=True)
hfrx_ED.to_csv('Event_Driven.csv',index=False)
```

Out[82]:

	date	name	index	change	size
131264	2003-03-31	HFRX Event Driven Index	HFRXED	0%	1000.00
131248	2003-04-01	HFRX Event Driven Index	HFRXED	0.05%	1000.45
131232	2003-04-02	HFRX Event Driven Index	HFRXED	0.26%	1003.03

	date	name	index	change	size
131216	2003-04-03	HFRX Event Driven Index	HFRXED	0.18%	1004.87
131200	2003-04-04	HFRX Event Driven Index	HFRXED	0.13%	1006.13
...	...	...	...	...	...
142	2021-07-16	HFRX Event Driven Index	HFRXED	0%	1813.60
110	2021-07-19	HFRX Event Driven Index	HFRXED	-0.32%	1807.82
78	2021-07-20	HFRX Event Driven Index	HFRXED	-0.09%	1806.21
46	2021-07-21	HFRX Event Driven Index	HFRXED	0.16%	1809.01
14	2021-07-22	HFRX Event Driven Index	HFRXED	0.04%	1809.72

In [81]:

```
hfrx_Macro=hfrx[hfrx['name']=='HFRX Macro/CTA Index']
hfrx_Macro['date']=pd.to_datetime(hfrx_Macro.date)
hfrx_Macro.sort_values(by='date',ascending=True,inplace=True)
hfrx_Macro.to_csv('Macro_CTA.csv',index=False)
```

Out[81]:

	date	name	index	change	size
131268	2003-03-31	HFRX Macro/CTA Index	HFRXM	0%	1000.00
131253	2003-04-01	HFRX Macro/CTA Index	HFRXM	0.15%	1001.51
131237	2003-04-02	HFRX Macro/CTA Index	HFRXM	0.10%	1002.50
131221	2003-04-03	HFRX Macro/CTA Index	HFRXM	-0.35%	999.00
131205	2003-04-04	HFRX Macro/CTA Index	HFRXM	0.02%	999.18
...	...	...	...	...	...
151	2021-07-16	HFRX Macro/CTA Index	HFRXM	-0.22%	1240.50
119	2021-07-19	HFRX Macro/CTA Index	HFRXM	-0.98%	1228.31
87	2021-07-20	HFRX Macro/CTA Index	HFRXM	-0.06%	1227.52
55	2021-07-21	HFRX Macro/CTA Index	HFRXM	0.53%	1234.08
23	2021-07-22	HFRX Macro/CTA Index	HFRXM	0.33%	1238.21

4618 rows × 5 columns

In [80]:

```
hfrx_RV=hfrx[hfrx['name']=='HFRX Relative Value Arbitrage Index']
hfrx_RV['date']=pd.to_datetime(hfrx_RV.date)
hfrx_RV.sort_values(by='date',ascending=True,inplace=True)
hfrx_RV.to_csv('Relative_Value.csv',index=False)
```

Out[80]:

	date	name	index	change	size

	date	name	index	change	size
131270	2003-03-31	HFRX Relative Value Arbitrage Index	HFRXRVA	0%	1000.00
131255	2003-04-01	HFRX Relative Value Arbitrage Index	HFRXRVA	0.07%	1000.73
131239	2003-04-02	HFRX Relative Value Arbitrage Index	HFRXRVA	0.01%	1000.79
131223	2003-04-03	HFRX Relative Value Arbitrage Index	HFRXRVA	-0.05%	1000.32
131207	2003-04-04	HFRX Relative Value Arbitrage Index	HFRXRVA	0.13%	1001.62
...	...	...	...	...	...
156	2021-07-16	HFRX Relative Value Arbitrage Index	HFRXRVA	-0.02%	1359.53
124	2021-07-19	HFRX Relative Value Arbitrage Index	HFRXRVA	-0.16%	1357.36
92	2021-07-20	HFRX Relative Value Arbitrage Index	HFRXRVA	-0.01%	1357.27
60	2021-07-21	HFRX Relative Value Arbitrage Index	HFRXRVA	0.08%	1358.39
28	2021-07-22	HFRX Relative Value Arbitrage Index	HFRXRVA	0.06%	1359.24

In [79]:
```
hfrx_FI=hfrx[hfrx['name']=='HFRX RV: FI-Convertible Arbitrage Index']
hfrx_FI['date']=pd.to_datetime(hfrx_FI.date)
hfrx_FI.sort_values(by='date',ascending=True,inplace=True)
hfrx_FI.to_csv('RV_FI_Convertible_Arbitrage.csv',index=False)
```

Out[79]:

	date	name	index	change	size
131271	2003-03-31	HFRX RV: FI-Convertible Arbitrage Index	HFRXCA	0%	1000.00
131256	2003-04-01	HFRX RV: FI-Convertible Arbitrage Index	HFRXCA	0.05%	1000.54
131240	2003-04-02	HFRX RV: FI-Convertible Arbitrage Index	HFRXCA	0.07%	1001.21
131224	2003-04-03	HFRX RV: FI-Convertible Arbitrage Index	HFRXCA	0.19%	1003.15
131208	2003-04-04	HFRX RV: FI-Convertible Arbitrage Index	HFRXCA	0.18%	1005.00
...	...	...	...	...	...
157	2021-07-16	HFRX RV: FI-Convertible Arbitrage Index	HFRXCA	-0.12%	992.75
125	2021-07-19	HFRX RV: FI-Convertible Arbitrage Index	HFRXCA	-0.31%	989.62
93	2021-07-20	HFRX RV: FI-Convertible Arbitrage Index	HFRXCA	0.17%	991.34
61	2021-07-21	HFRX RV: FI-Convertible Arbitrage Index	HFRXCA	0.23%	993.64
29	2021-07-22	HFRX RV: FI-Convertible Arbitrage Index	HFRXCA	0.14%	995.03

4618 rows × 5 columns

In [78]:
```
hfrx_FI['date']=pd.to_datetime(hfrx_FI.date)
```

Out[78]:

	date	name	index	change	size

	date	name	index	change	size
131271	2003-03-31	HFRX RV: FI-Convertible Arbitrage Index	HFRXCA	0%	1000.00
131256	2003-04-01	HFRX RV: FI-Convertible Arbitrage Index	HFRXCA	0.05%	1000.54
131240	2003-04-02	HFRX RV: FI-Convertible Arbitrage Index	HFRXCA	0.07%	1001.21
131224	2003-04-03	HFRX RV: FI-Convertible Arbitrage Index	HFRXCA	0.19%	1003.15
131208	2003-04-04	HFRX RV: FI-Convertible Arbitrage Index	HFRXCA	0.18%	1005.00
...	...	...	...	...	...
157	2021-07-16	HFRX RV: FI-Convertible Arbitrage Index	HFRXCA	-0.12%	992.75
125	2021-07-19	HFRX RV: FI-Convertible Arbitrage Index	HFRXCA	-0.31%	989.62
93	2021-07-20	HFRX RV: FI-Convertible Arbitrage Index	HFRXCA	0.17%	991.34
61	2021-07-21	HFRX RV: FI-Convertible Arbitrage Index	HFRXCA	0.23%	993.64
29	2021-07-22	HFRX RV: FI-Convertible Arbitrage Index	HFRXCA	0.14%	995.03

2 获取 SP，HSI，CS

In [7]:

In [6]:

...

In [8]:

```
#设置token
ts.set_token('3f838e4e9b13f9be758c8d55a3eb7e37365be40173017a6e073d51a4')
#初始化pro接口
```

In [11]:

```
#获取标普500
df_SPX = pro.index_global(ts_code='SPX', start_date='20030101', end_date='20210725')
```

...

In [12]:

In [10]:

```
#获取恒生指数
df_HSI = pro.index_global(ts_code='HSI', start_date='20030101', end_date='20210725')
```

...

In [13]:

In [ ]:

In [53]:
```
#获取招商银行（香港上市）
df = pro.hk_basic()
#获取全部退市股票基础信息
#ts_code=03968.HK
df_zhaoshang=pro.hk_basic(ts_code='03968.HK')
```

```
---------------------------------------------------------------------------
Exception                                 Traceback (most recent call last)
<ipython-input-53-2b8adb300b43> in <module>
      1 #获取招商银行（香港上市）
----> 2 df = pro.hk_basic()
      3 #获取全部退市股票基础信息
      4 #ts_code=03968.HK
      5 df_zhaoshang=pro.hk_basic(ts_code='03968.HK')

D:\anaconda\result_product\lib\site-packages\tushare\pro\client.py in query(self, api_name, fields, **kwargs)
     42             result = json.loads(res.text)
     43             if result['code'] != 0:
---> 44                 raise Exception(result['msg'])
     45             data = result['data']
     46             columns = data['fields']

Exception: 抱歉，您没有访问该接口的权限，权限的具体详情访问：https://tushare.pro/document/1?doc_id=108。
```

In [59]:
```
import baostock as bs
def baostock(
        code="sh.600000",
        fields="date,code,open,high,low,close,preclose,volume,amount,adjustflag,turn,tradestatus,pctChg,isST",
        start_date='2003-01-01',
        end_date='2021-7-26',
        frequency="d",
        adjustflag="3",
        download=False,
        location="D:\\history_A_stock_k_data.csv"):
    #### 登陆系统 ####
    lg = bs.login()
    # 显示登陆返回信息
    print('login respond error_code:' + lg.error_code)
    print('login respond  error_msg:' + lg.error_msg)

    #### 获取沪深A股历史K线数据 ####
    # 详细指标参数，参见“历史行情指标参数”章节
    rs = bs.query_history_k_data(code,
                                 fields,
                                 start_date,
                                 end_date,
                                 frequency,
                                 adjustflag)
    print('query_history_k_data respond error_code:' + rs.error_code)
    print('query_history_k_data respond  error_msg:' + rs.error_msg)

    #### 打印结果集 ####
    data_list = []
    while (rs.error_code == '0') & rs.next():
        # 获取一条记录，将记录合并在一起
        data_list.append(rs.get_row_data())
    result = pd.DataFrame(data_list, columns=rs.fields)
    print(result)
    if download:
        #### 结果集输出到csv文件 ####
        result.to_csv(location, index=False)
        print('---Downloaded Successfully!---')
    else:
        print('---Download:Choose not to download.---')

    #### 登出系统 ####
```

In [60]:

```
#茅台
```

```
login success!
login respond error_code:0
login respond  error_msg:success
query_history_k_data respond error_code:0
query_history_k_data respond  error_msg:success
            date       code     open     high      low    close preclose  \
0     2003-01-02  sz.000651   7.9900   8.1000   7.9000   7.9100   8.3400
1     2003-01-03  sz.000651   7.9100   8.0500   7.9000   7.9700   7.9100
2     2003-01-06  sz.000651   7.9200   8.2500   7.9200   8.1600   7.9700
3     2003-01-07  sz.000651   8.1900   8.2200   8.1000   8.1600   8.1600
4     2003-01-08  sz.000651   8.1600   8.2900   8.1200   8.2600   8.1600
...          ...        ...      ...      ...      ...      ...      ...
4503  2021-07-19  sz.000651  50.3000  51.7300  50.0000  51.4800  49.9900
4504  2021-07-20  sz.000651  51.3500  51.9400  50.7000  51.2000  51.4800
4505  2021-07-21  sz.000651  51.2000  51.4900  49.9000  49.9500  51.2000
4506  2021-07-22  sz.000651  49.8000  50.4500  49.5000  49.9500  49.9500
4507  2021-07-23  sz.000651  49.9000  49.9100  48.9800  49.0200  49.9500
```

In [35]:

```python
sp=pd.read_csv('HistoricalPrices.csv')
```

...

In [36]:

...

In [38]:

```python
HS_345=pd.read_csv('HSI.csv')
```

...

In [41]:

```python
HS_345['Date']=pd.to_datetime(HS_345['Date'])
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 770 entries, 0 to 769
Data columns (total 7 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Date       770 non-null    datetime64[ns]
 1   Open       744 non-null    float64
 2   High       744 non-null    float64
 3   Low        744 non-null    float64
 4   Close      744 non-null    float64
 5   Adj Close  744 non-null    float64
 6   Volume     744 non-null    float64
dtypes: datetime64[ns](1), float64(6)
memory usage: 42.2 KB
```

In [45]:

```python
HS_345.sort_values(by='Date',ascending=False,inplace=True)
HS_345
```

...

In [47]:
```
df_HSI['Date']=pd.to_datetime(df_HSI.trade_date)
df_HSI
```

...

3 三大指数

In [61]:
```
df_indices=pd.read_excel('三大指数.xlsx')
df_indices
```

Out[61]:

	时间	恒生指数	美国:标准普尔500指数	沪深300指数	date	close
0	2003-01-02	9365.52	909.03	1074.9840	2003-01-02	7.91
1	2003-01-03	9583.85	908.59	1073.5240	2003-01-03	7.97
2	2003-01-06	9665.96	929.01	1084.8840	2003-01-06	8.16
3	2003-01-07	9652.40	922.93	1081.6220	2003-01-07	8.16
4	2003-01-08	9688.21	909.93	1111.8320	2003-01-08	8.26
...	...	...	...	...	...	...
4814	2021-07-14	27787.46	4374.30	5083.0849	NaT	NaN
4815	2021-07-15	27996.27	4360.03	5151.4626	NaT	NaN
4816	2021-07-16	28004.68	4327.16	5094.7727	NaT	NaN
4817	2021-07-19	27489.78	4258.49	5113.4945	NaT	NaN
4818	2021-07-20	27259.25	4323.06	5108.9941	NaT	NaN

4819 rows × 6 columns

In [62]:
```
df_indices.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4819 entries, 0 to 4818
Data columns (total 6 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   时间            4819 non-null   datetime64[ns]
 1   恒生指数          4576 non-null   float64
 2   美国:标准普尔500指数  4669 non-null   float64
 3   沪深300指数       4505 non-null   float64
 4   date          4508 non-null   datetime64[ns]
 5   close         4508 non-null   float64
dtypes: datetime64[ns](2), float64(4)
memory usage: 226.0 KB
```

In [64]:
```
gree=df_indices.iloc[:,[4,5]]
```

Out[64]:

	date	close
0	2003-01-02	7.91
1	2003-01-03	7.97
2	2003-01-06	8.16
3	2003-01-07	8.16
4	2003-01-08	8.26
...	...	...
4814	NaT	NaN
4815	NaT	NaN
4816	NaT	NaN
4817	NaT	NaN
4818	NaT	NaN

4819 rows × 2 columns

In [66]:
```
indices_2=df_indices.iloc[:,:3]
```

Out[66]:

	时间	恒生指数	美国:标准普尔500指数
0	2003-01-02	9365.52	909.03
1	2003-01-03	9583.85	908.59
2	2003-01-06	9665.96	929.01
3	2003-01-07	9652.40	922.93
4	2003-01-08	9688.21	909.93
...	...	...	...
4814	2021-07-14	27787.46	4374.30
4815	2021-07-15	27996.27	4360.03
4816	2021-07-16	28004.68	4327.16
4817	2021-07-19	27489.78	4258.49
4818	2021-07-20	27259.25	4323.06

4819 rows × 3 columns

In [72]:
```
[illegible]
```

In [76]:
```
SHG=pd.read_csv('SP恒生Gree.csv').drop('date',axis=1)
SHG
```

Out[76]:

	时间	HS	SP	Gree
0	2003/1/2	9365.52	909.03	7.91
1	2003/1/3	9583.85	908.59	7.97
2	2003/1/6	9665.96	929.01	8.16
3	2003/1/7	9652.40	922.93	8.16
4	2003/1/8	9688.21	909.93	8.26
...	...	...	...	...
4500	2021/7/14	27787.46	4374.30	49.60
4501	2021/7/15	27996.27	4360.03	49.50
4502	2021/7/16	28004.68	4327.16	49.99
4503	2021/7/19	27489.78	4258.49	51.48
4504	2021/7/20	27259.25	4323.06	51.20

4505 rows × 4 columns

4 All

In [304]:
```
df_all_price=pd.read_excel('All.xlsx')
df_all_price
```

Out[304]:

	date	Global HFI	Equity Hedge	Event Driven	Macro/CTA	Relative Value	RV:FI Convertible Arbitrage	SP	HS	Gree
0	2003-03-31	1000.00	1000.00	1000.00	1000.00	1000.00	1000.00	848.18	8634.45	8.98
1	2003-04-01	1002.00	1003.76	1000.45	1001.51	1000.73	1000.54	858.48	8596.89	8.98
2	2003-04-02	1004.23	1008.31	1003.03	1002.50	1000.79	1001.21	880.90	8706.19	9.01
3	2003-04-03	1004.97	1009.74	1004.87	999.00	1000.32	1003.15	876.45	8648.16	8.90
4	2003-04-04	1004.96	1008.19	1006.13	999.18	1001.62	1005.00	878.85	8822.45	8.92
...	...	...	...	...	...	...	...	...	...	...
4306	2021-07-14	1431.02	1441.47	1815.23	1246.92	1360.82	996.46	4374.30	27787.46	49.60
4307	2021-07-15	1428.69	1438.09	1813.67	1243.23	1359.78	993.93	4360.03	27996.27	49.50
4308	2021-07-16	1426.85	1433.95	1813.60	1240.50	1359.53	992.75	4327.16	28004.68	49.99

	date	Global HFI	Equity Hedge	Event Driven	Macro/CTA	Relative Value	RV:FI Convertible Arbitrage	SP	HS	Gree
4309	2021-07-19	1418.72	1420.68	1807.82	1228.31	1357.36	989.62	4258.49	27489.78	51.48
4310	2021-07-20	1420.70	1428.96	1806.21	1227.52	1357.27	991.34	4323.06	27259.25	51.20

In [305]:

Out[305]:

```
date                           4311
Global HFI                     4311
Equity Hedge                   4311
Event Driven                   4311
Macro/CTA                      4311
Relative Value                 4311
RV:FI Convertible Arbitrage    4311
SP                             4305
HS                             4211
Gree                           4311
dtype: int64
```

In [307]:

```
df_all_price=df_all_price.dropna()
```

Out[307]:

```
date                           4205
Global HFI                     4205
Equity Hedge                   4205
Event Driven                   4205
Macro/CTA                      4205
Relative Value                 4205
RV:FI Convertible Arbitrage    4205
SP                             4205
HS                             4205
Gree                           4205
dtype: int64
```

In [92]:

```
# df_all=pd.merge(df_all.iloc[:,:7],df_all.iloc[:,7:],left_on='date',right_on='时间')
# df_all.drop('时间',axis=0,inplace=True).to_excel('All.xlsx',index=False)
```

...

In [94]:

Out[94]:

```
RangeIndex(start=0, stop=4311, step=1)
```

In [95]:

```
df_all.index=df_all.date
```

Out[95]:

	date	Global HFI	Equity Hedge	Event Driven	Macro/CTA	Relative Value	RV:FI Convertible Arbitrage	SP	HS	Gree
date										
2003-03-31	2003-03-31	1000.00	1000.00	1000.00	1000.00	1000.00	1000.00	848.18	8634.45	8.98

	date	Global HFI	Equity Hedge	Event Driven	Macro/CTA	Relative Value	RV:FI Convertible Arbitrage	SP	HS	Gree
date										
2003-04-01	2003-04-01	1002.00	1003.76	1000.45	1001.51	1000.73	1000.54	858.48	8596.89	8.98
2003-04-02	2003-04-02	1004.23	1008.31	1003.03	1002.50	1000.79	1001.21	880.90	8706.19	9.01
2003-04-03	2003-04-03	1004.97	1009.74	1004.87	999.00	1000.32	1003.15	876.45	8648.16	8.90
2003-04-04	2003-04-04	1004.96	1008.19	1006.13	999.18	1001.62	1005.00	878.85	8822.45	8.92
...	...	...	...	...	...	...	...	...	...	...
2021-07-14	2021-07-14	1431.02	1441.47	1815.23	1246.92	1360.82	996.46	4374.30	27787.46	49.60
2021-07-15	2021-07-15	1428.69	1438.09	1813.67	1243.23	1359.78	993.93	4360.03	27996.27	49.50
2021-07-16	2021-07-16	1426.85	1433.95	1813.60	1240.50	1359.53	992.75	4327.16	28004.68	49.99
2021-07-19	2021-07-19	1418.72	1420.68	1807.82	1228.31	1357.36	989.62	4258.49	27489.78	51.48

5 all_return

In [162]:

```
df_all=pd.read_excel('All_return.xlsx')
df_all.index=df_all.date
df_all
```

Out[162]:

	date	Global HFI	Equity Hedge	Event Driven	Macro/CTA	Relative Value	RV:FI Convertible Arbitrage	SP	HS	Gree
date										
2003-04-01	2003-04-01	0.002000	0.003760	0.000450	0.001510	0.000730	0.000540	0.012144	-0.004350	0.000000
2003-04-02	2003-04-02	0.002226	0.004533	0.002579	0.000989	0.000060	0.000670	0.026116	0.012714	0.003341
2003-04-03	2003-04-03	0.000737	0.001418	0.001834	-0.003491	-0.000470	0.001938	-0.005052	-0.006665	-0.012209
2003-04-04	2003-04-04	-0.000010	-0.001535	0.001254	0.000180	0.001300	0.001844	0.002738	0.020153	0.002247
2003-04-07	2003-04-07	-0.000896	-0.003928	0.001153	-0.002942	0.004902	0.001264	0.001229	0.015841	0.001121
...	...	...	...	...	...	...	...	...	...	...
2021-07-14	2021-07-14	-0.000629	-0.001718	-0.001320	0.001325	0.000184	-0.000782	0.001165	-0.006292	0.004049
2021-07-15	2021-07-15	-0.001628	-0.002345	-0.000859	-0.002959	-0.000764	-0.002539	-0.003262	0.007515	-0.002016
2021-07-16	2021-07-16	-0.001288	-0.002879	-0.000039	-0.002196	-0.000184	-0.001187	-0.007539	0.000300	0.009899
2021-07-19	2021-07-19	-0.005698	-0.009254	-0.003187	-0.009827	-0.001596	-0.003153	-0.015870	-0.018386	0.029806
2021-07-20	2021-07-20	0.001396	0.005828	-0.000891	-0.000643	-0.000066	0.001738	0.015163	-0.008386	-0.005439

4310 rows × 10 columns

In [163]:

Out[163]:
```
DatetimeIndex(['2003-04-01', '2003-04-02', '2003-04-03', '2003-04-04',
               '2003-04-07', '2003-04-08', '2003-04-09', '2003-04-10',
               '2003-04-11', '2003-04-14',
               ...
               '2021-07-07', '2021-07-08', '2021-07-09', '2021-07-12',
               '2021-07-13', '2021-07-14', '2021-07-15', '2021-07-16',
               '2021-07-19', '2021-07-20'],
              dtype='datetime64[ns]', name='date', length=4310, freq=None)
```

In [220]:

Out[220]:
```
date                           4310
Global HFI                     4310
Equity Hedge                   4310
Event Driven                   4310
Macro/CTA                      4310
Relative Value                 4310
RV:FI Convertible Arbitrage    4310
SP                             4304
HS                             4210
Gree                           4310
enumerate                      4310
dtype: int64
```

In [221]:
```
df_all=df_all.dropna()
```

Out[221]:
```
date                           4204
Global HFI                     4204
Equity Hedge                   4204
Event Driven                   4204
Macro/CTA                      4204
Relative Value                 4204
RV:FI Convertible Arbitrage    4204
SP                             4204
HS                             4204
Gree                           4204
enumerate                      4204
dtype: int64
```

In [252]:

In [222]:
```
df_all['enumerate']=np.arange(len(df_all.index))
```

Out[222]:

	date	Global HFI	Equity Hedge	Event Driven	Macro/CTA	Relative Value	RV:FI Convertible Arbitrage	SP	HS	Gree	enumerate

date										
2003-04-01	2003-04-01	0.002000	0.003760	0.000450	0.001510	0.000730	0.000540	0.012144	-0.004350	0.000000	0
2003-04-02	2003-04-02	0.002226	0.004533	0.002579	0.000989	0.000060	0.000670	0.026116	0.012714	0.003341	1
2003-04-03	2003-04-03	0.000737	0.001418	0.001834	-0.003491	-0.000470	0.001938	-0.005052	-0.006665	-0.012209	2
2003-04-04	2003-04-04	-0.000010	-0.001535	0.001254	0.000180	0.001300	0.001844	0.002738	0.020153	0.002247	3
2003-04-07	2003-04-07	-0.000896	-0.003928	0.001153	-0.002942	0.004902	0.001264	0.001229	0.015841	0.001121	4
...	...	...	...	...	...	...	...	...	...	...	...
2021-07-14	2021-07-14	-0.000629	-0.001718	-0.001320	0.001325	0.000184	-0.000782	0.001165	-0.006292	0.004049	4199
2021-07-15	2021-07-15	-0.001628	-0.002345	-0.000859	-0.002959	-0.000764	-0.002539	-0.003262	0.007515	-0.002016	4200
2021-07-16	2021-07-16	-0.001288	-0.002879	-0.000039	-0.002196	-0.000184	-0.001187	-0.007539	0.000300	0.009899	4201
2021-07-19	2021-07-19	-0.005698	-0.009254	-0.003187	-0.009827	-0.001596	-0.003153	-0.015870	-0.018386	0.029806	4202
2021-07-20	2021-07-20	0.001396	0.005828	-0.000891	-0.000643	-0.000066	0.001738	0.015163	-0.008386	-0.005439	4203

4204 rows × 11 columns

5.1 two year window

In [223]:

```
## 获取具体某天的数据，用datafrme直接选取某天时会报错，而series的数据就没有问题
# print(df['2013-11-06'])
#可以考虑用区间来获取某天的数据 print(df['2013-11-06':'2013-11-06'])
df_all['2005-03-31':'2005-03-31']
```

Out[223]:

	date	Global HFI	Equity Hedge	Event Driven	Macro/CTA	Relative Value	RV:FI Convertible Arbitrage	SP	HS	Gree	enumerate
date											
2005-03-31	2005-03-31	0.001424	0.00281	0.003279	0.001489	0.001685	-0.008255	-0.000694	0.006788	0.0	452

In [224]:

```
def window(end=480):
    windows=[]
    for i in df_all.index[466:end]:
        right = str(i.date())

        left = right.split('-')
        left[0] = str(int(left[0]) - 2)
        if right[-5:]=='02-29':
            left[-1]='28'
        left = '-'.join(left)

        windows.append((left,right))
    return windows
window(end=1160)[-20:]
```

Out[224]:

```
[('2006-04-03', '2008-04-03'),
 ('2006-04-07', '2008-04-07'),
 ('2006-04-08', '2008-04-08'),
 ('2006-04-09', '2008-04-09'),
 ('2006-04-10', '2008-04-10'),
 ('2006-04-11', '2008-04-11'),
 ('2006-04-14', '2008-04-14'),
 ('2006-04-15', '2008-04-15'),
 ('2006-04-16', '2008-04-16'),
 ('2006-04-17', '2008-04-17'),
 ('2006-04-18', '2008-04-18'),
 ('2006-04-21', '2008-04-21'),
 ('2006-04-22', '2008-04-22'),
 ('2006-04-23', '2008-04-23'),
 ('2006-04-24', '2008-04-24'),
 ('2006-04-25', '2008-04-25'),
 ('2006-04-28', '2008-04-28'),
 ('2006-04-29', '2008-04-29'),
 ('2006-04-30', '2008-04-30'),
```

```
In  [225]: windows=window(end=None)
```

```
Out[225]: [('2019-07-07', '2021-07-07'),
           ('2019-07-08', '2021-07-08'),
           ('2019-07-09', '2021-07-09'),
           ('2019-07-12', '2021-07-12'),
           ('2019-07-13', '2021-07-13'),
           ('2019-07-14', '2021-07-14'),
           ('2019-07-15', '2021-07-15'),
           ('2019-07-16', '2021-07-16'),
           ('2019-07-19', '2021-07-19'),
           ('2019-07-20', '2021-07-20')]
```

```
In  [226]: for pair in windows[:3]:
```

```
date
2003-04-21    0.001370
2003-04-23    0.002058
2003-04-24    0.000178
2003-04-25   -0.000582
2003-04-28    0.001887
                 ...
2005-04-15   -0.005450
2005-04-18   -0.003302
2005-04-19    0.003358
2005-04-20   -0.001791
2005-04-21    0.002443
Name: Global HFI, Length: 454, dtype: float64
date
2003-04-23    0.002058
2003-04-24    0.000178
2003-04-25   -0.000582
2003-04-28    0.001887
2003-04-29   -0.000148
                 ...
2005-04-18   -0.003302
2005-04-19    0.003358
2005-04-20   -0.001791
2005-04-21    0.002443
```

In [227]:

```
def calculate(col_index):
    df_col=df_all.iloc[:,col_index]
    means=[]
    stds=[]
    skews=[]
    kurts=[]
#     print(df_col)
    for pair in windows:
#         print(pair)
        means.append(df_col[pair[0]:pair[1]].mean())
        stds.append(df_col[pair[0]:pair[1]].std())
        skews.append(df_col[pair[0]:pair[1]].skew())
        kurts.append(df_col[pair[0]:pair[1]].kurt())
    return  means,stds,skews,kurts
def endow(col_index):
    df_col_1=pd.DataFrame()
    df_col_1['Date']=df_all.index[466:None]
    df_col_1['mean'],df_col_1['std'],df_col_1['skew'],df_col_1['kurt'] =calculate(col_index)
```

5.1.1 df_1

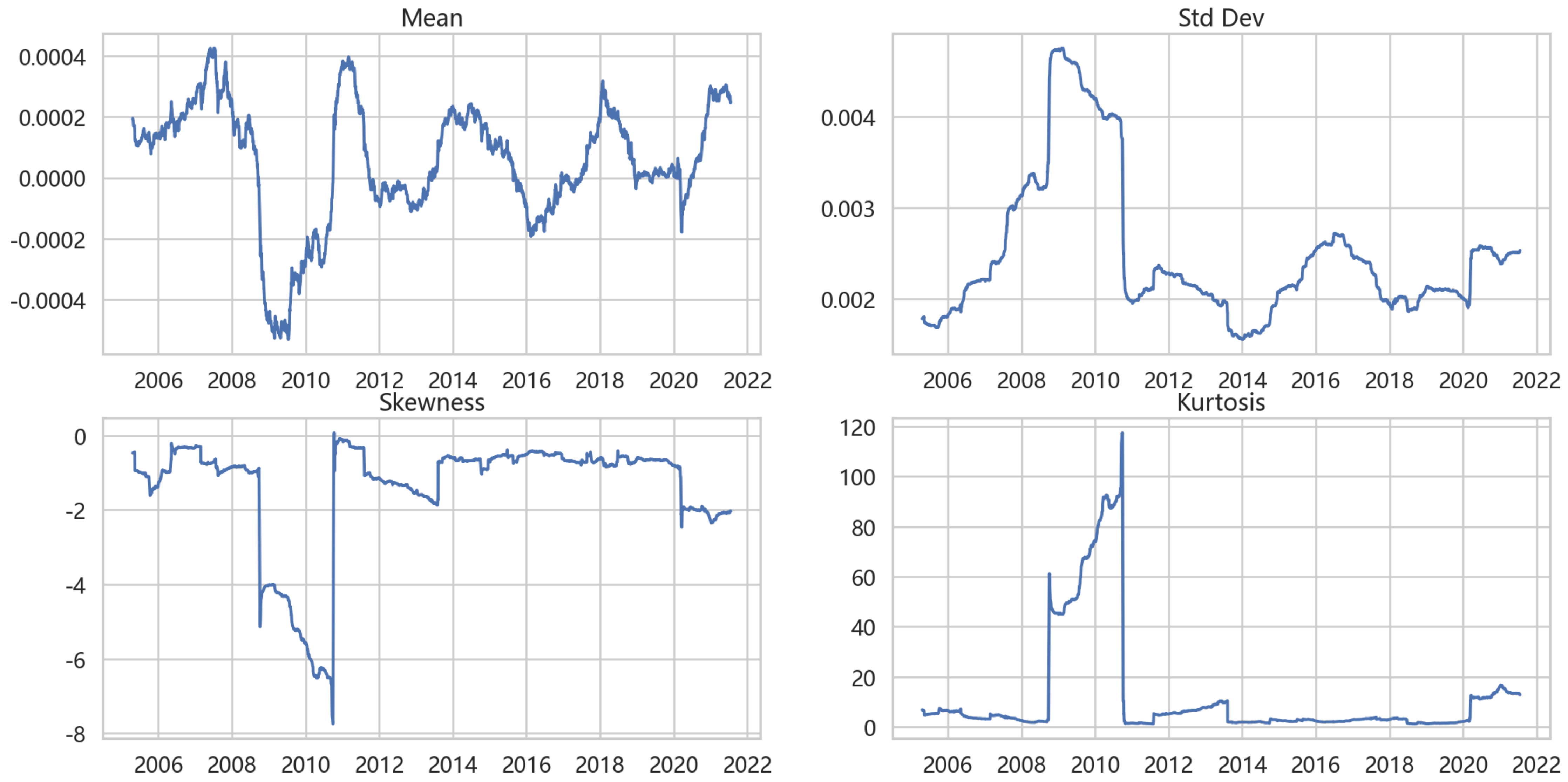
In [228]:
```
df_1= endow(1)
```

Out[228]:

	Date	mean	std	skew	kurt
0	2005-04-21	0.000196	0.001781	-0.473273	6.806719
1	2005-04-22	0.000191	0.001781	-0.467195	6.812442
2	2005-04-25	0.000190	0.001781	-0.467207	6.810387
3	2005-04-26	0.000185	0.001788	-0.466218	6.695391
4	2005-04-27	0.000179	0.001790	-0.460529	6.634321
...	...	...	...	...	...
3733	2021-07-14	0.000267	0.002511	-2.058816	13.272516
3734	2021-07-15	0.000262	0.002510	-2.053260	13.254505
3735	2021-07-16	0.000258	0.002511	-2.045969	13.217848
3736	2021-07-19	0.000246	0.002531	-2.020512	12.803246
3737	2021-07-20	0.000249	0.002532	-2.023181	12.802115

3738 rows × 5 columns

In [229]:
```
def graph(df_1):
    x=df_1.Date
    plt.figure(figsize=(20,10),dpi=120)
    plt.subplot(221)
    plt.plot(x,df_1.iloc[:,1],label='mean')
    plt.title('Mean')
    plt.subplot(222)
    plt.plot(x,df_1.iloc[:,2],label='std')
    plt.title('Std Dev')
    plt.subplot(223)
    plt.plot(x,df_1.iloc[:,3],label='skewness')
    plt.title('Skewness')
    plt.subplot(224)
    plt.plot(x,df_1.iloc[:,4],label='kurtosis')
    plt.title('Kurtosis')
#     plt.legend()
    plt.show()
```

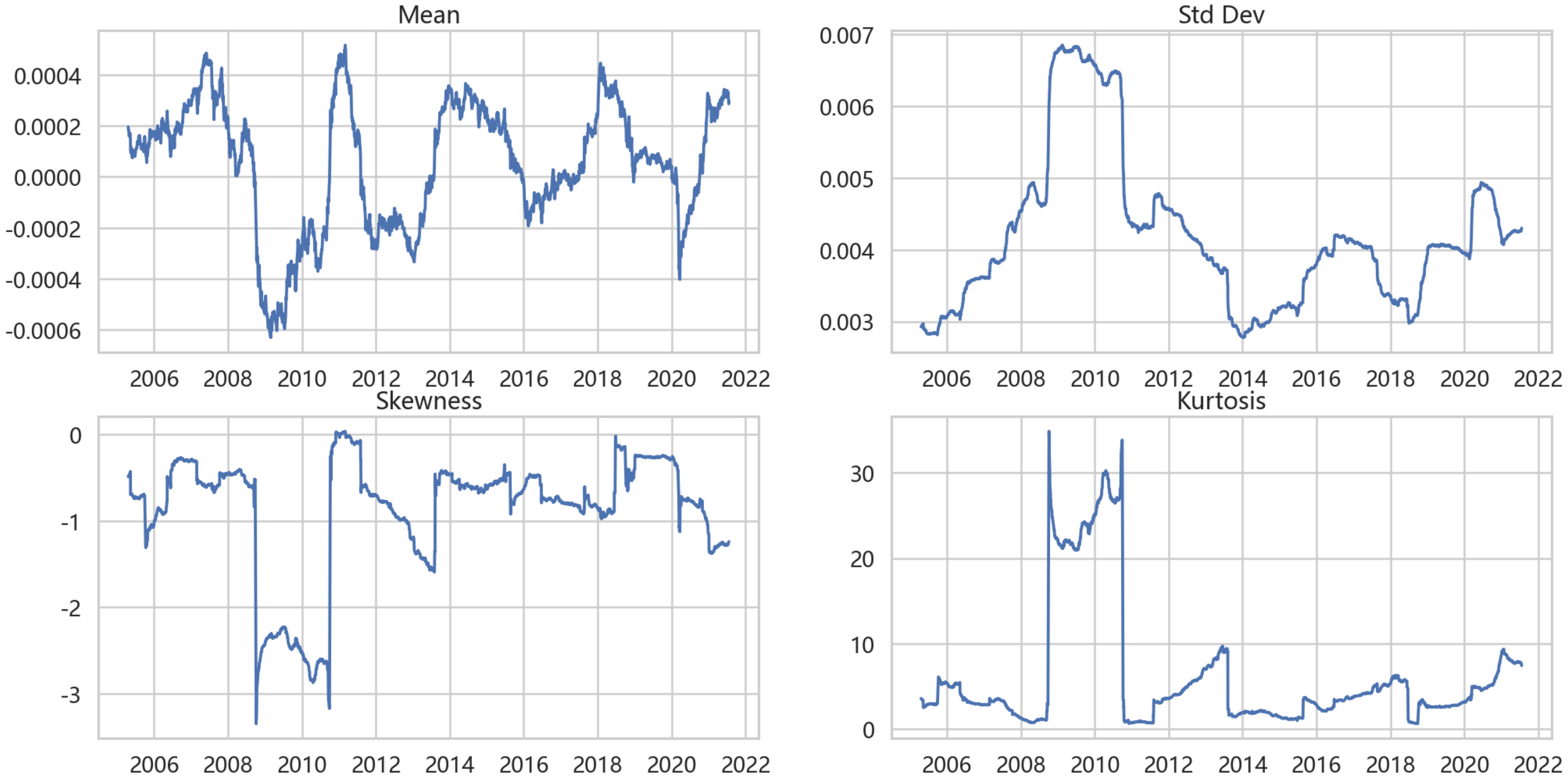


5.1.2 df_2-9

In [230]:

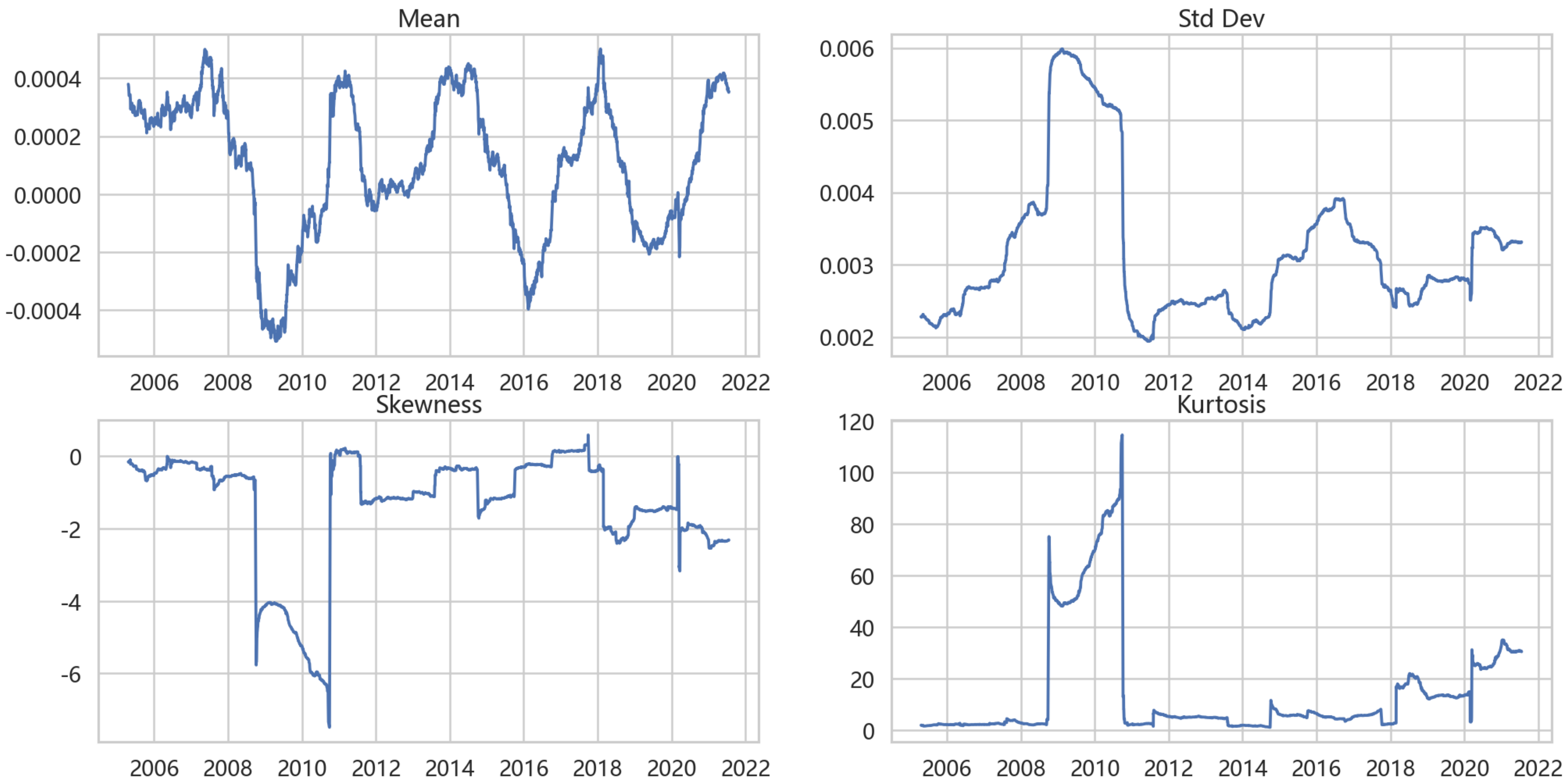
```
df_2= endow(2)
df_3= endow(3)
df_4= endow(4)
df_5= endow(5)
df_6= endow(6)
df_7= endow(7)
df_8= endow(8)
df_9= endow(9)
```

In [231]:

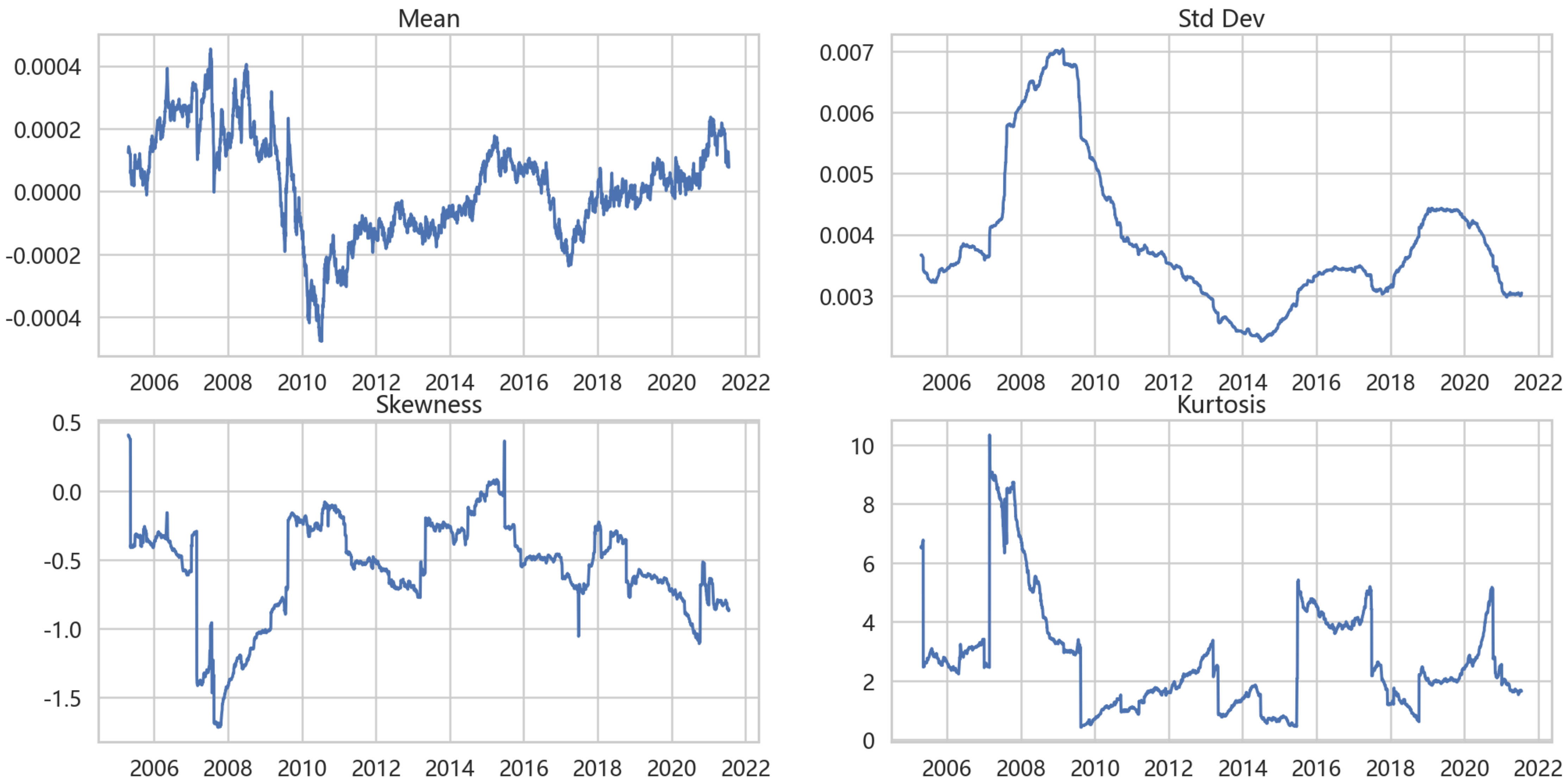


Contents

In [232]:

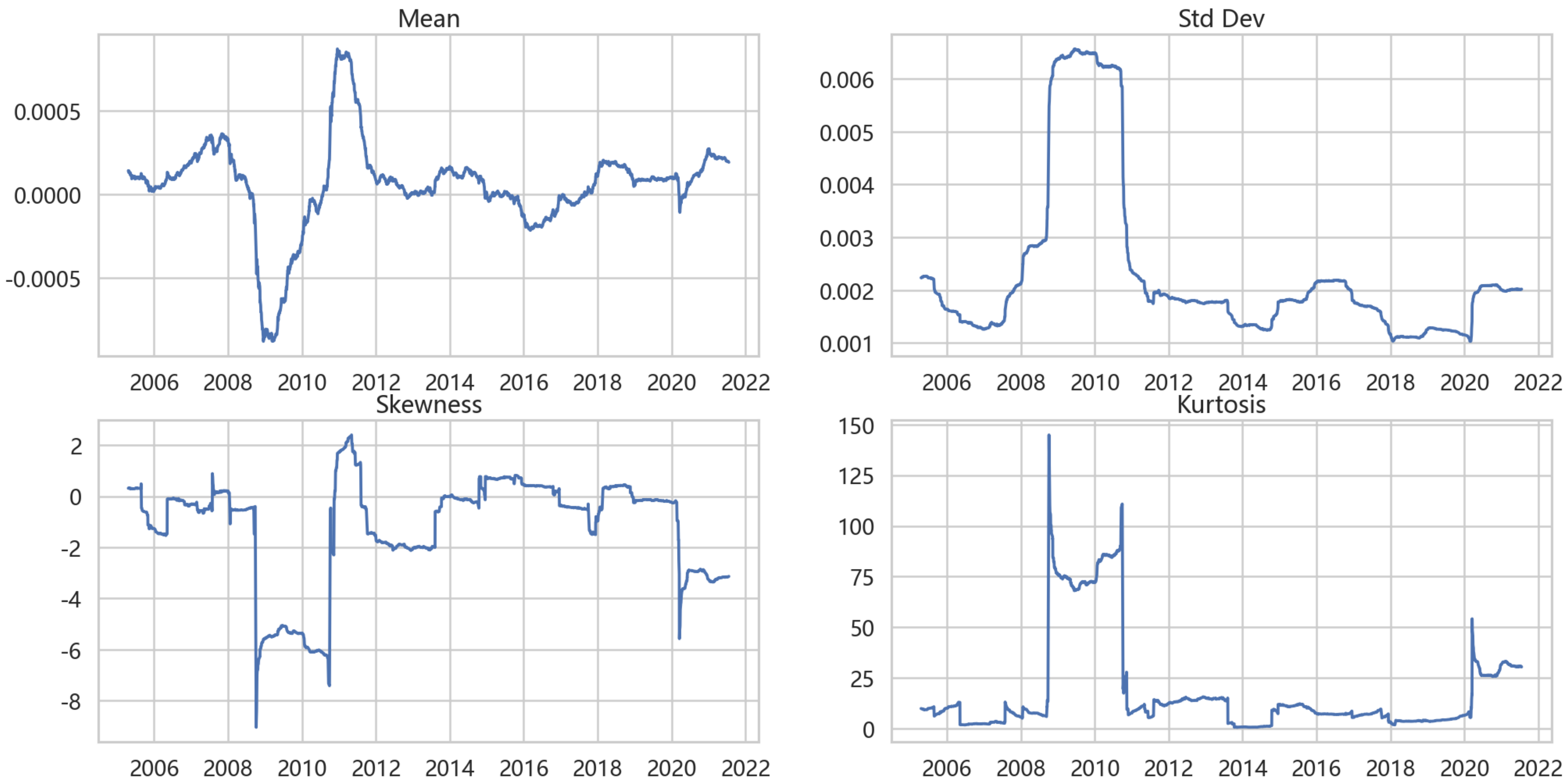


In [233]:

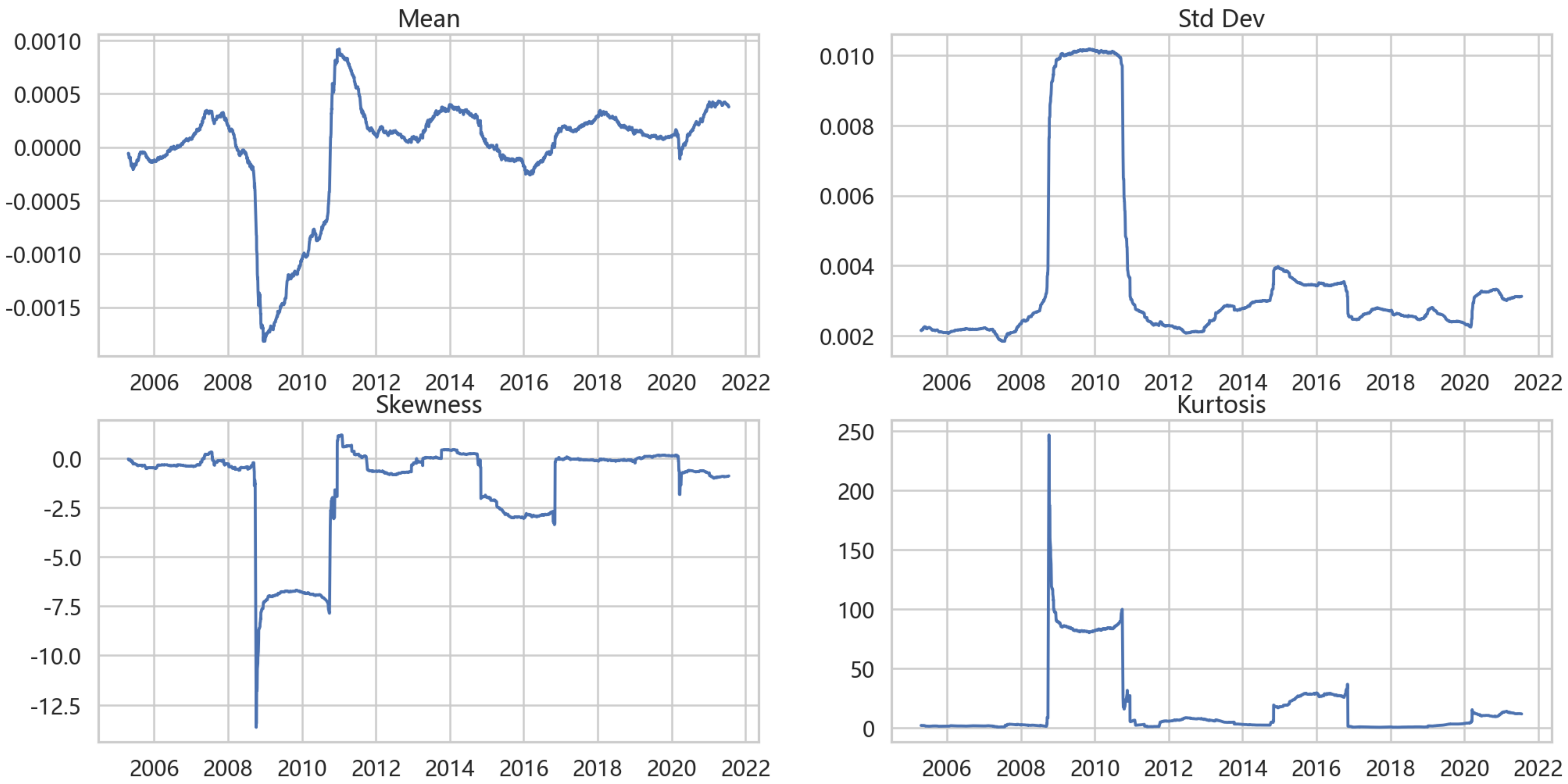

Mean
0.0004
0.0002
0.0000
-0.0002
-0.0004
2006
2008
2010
2012
2014
2016
2018
2020
2022
Std Dev
0.007
0.006
0.005
0.004
0.003
2006
2008
2010
2012
2014
2016
2018
2020
2022
Skewness
0.5
0.0
-0.5
-1.0
-1.5
2006
2008
2010
2012
2014
2016
2018
2020
2022
Kurtosis
10
8
6
4
2
0
2006
2008
2010
2012
2014
2016
2018
2020
2022

In [234]:

```
[illegible]
```

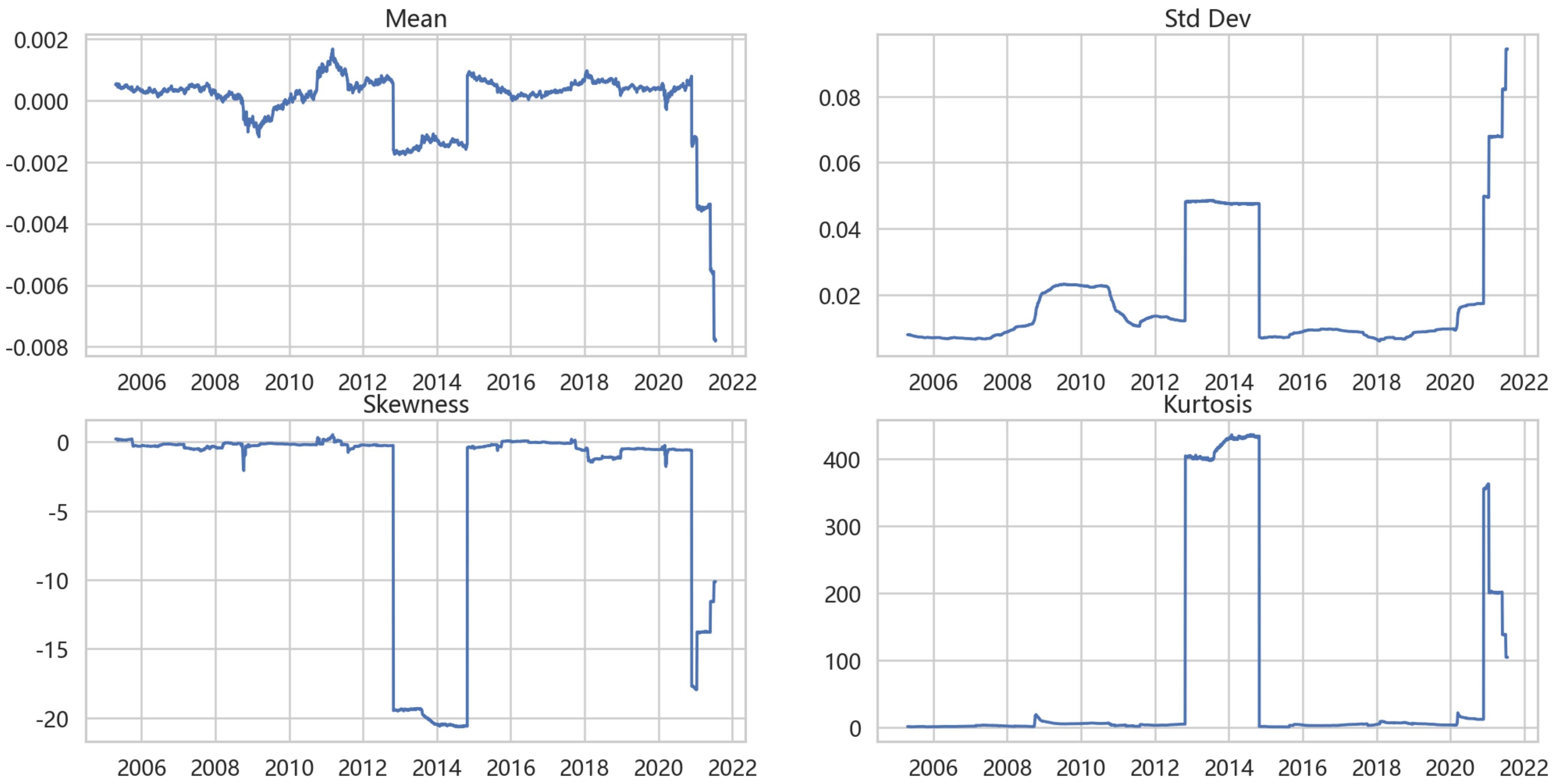

Mean
0.0005
0.0000
-0.0005
2006
2008
2010
2012
2014
2016
2018
2020
2022
Std Dev
0.006
0.005
0.004
0.003
0.002
0.001
2006
2008
2010
2012
2014
2016
2018
2020
2022
Skewness
2
0
-2
-4
-6
-8
2006
2008
2010
2012
2014
2016
2018
2020
2022
Kurtosis
150
125
100
75
50
25
0
2006
2008
2010
2012
2014
2016
2018
2020
2022

In [235]:

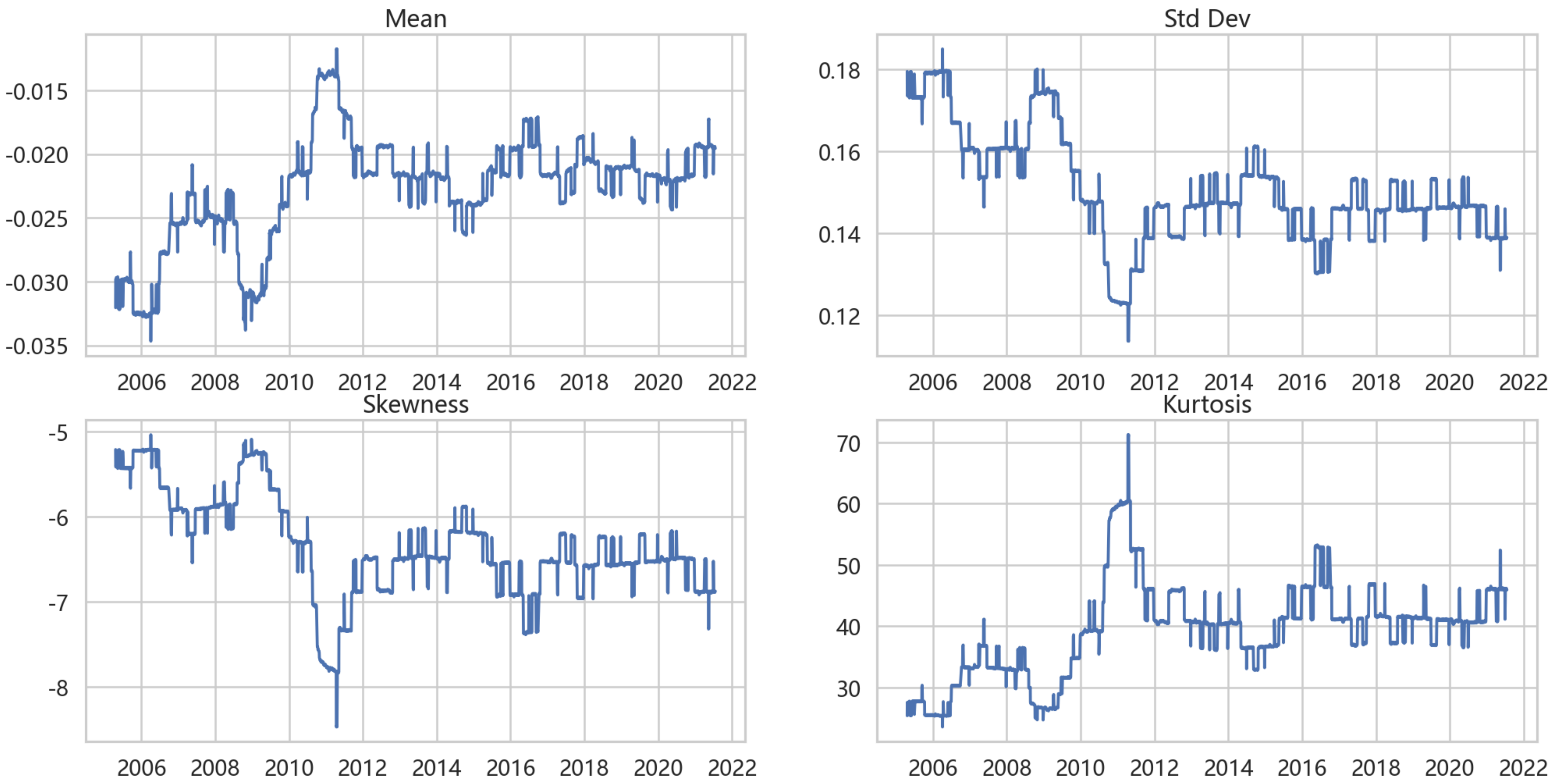

Mean
0.0010
0.0005
0.0000
-0.0005
-0.0010
-0.0015
2006
2008
2010
2012
2014
2016
2018
2020
2022
Std Dev
0.010
0.008
0.006
0.004
0.002
2006
2008
2010
2012
2014
2016
2018
2020
2022
Skewness
0.0
-2.5
-5.0
-7.5
-10.0
-12.5
2006
2008
2010
2012
2014
2016
2018
2020
2022
Kurtosis
250
200
150
100
50
0
2006
2008
2010
2012
2014
2016
2018
2020
2022

Contents

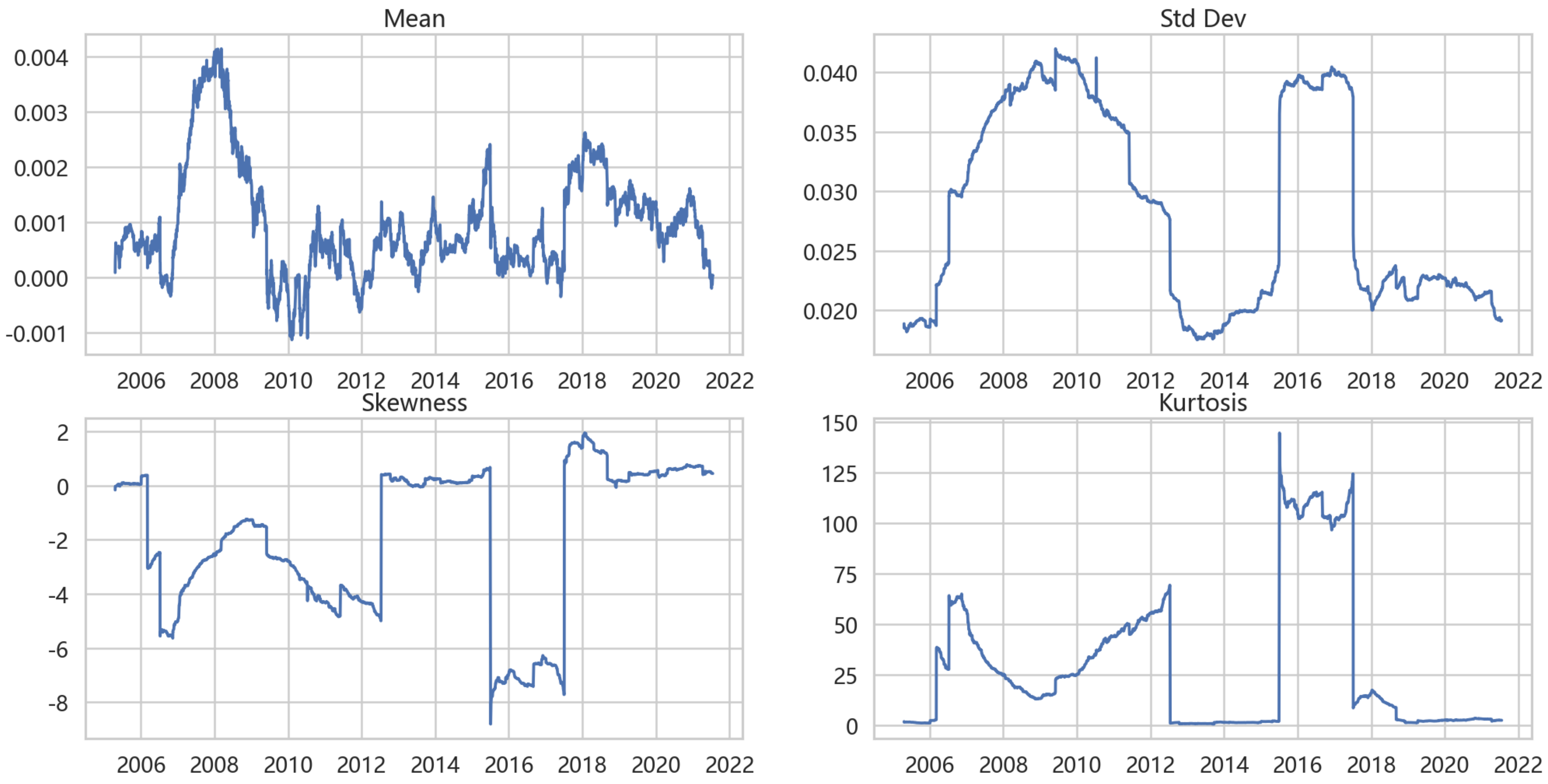
In [236]:



In [237]:


Mean
-0.015
-0.020
-0.025
-0.030
-0.035
2006 2008 2010 2012 2014 2016 2018 2020 2022
Std Dev
0.18
0.16
0.14
0.12
2006 2008 2010 2012 2014 2016 2018 2020 2022
Skewness
-5
-6
-7
-8
2006 2008 2010 2012 2014 2016 2018 2020 2022
Kurtosis
70
60
50
40
30
2006 2008 2010 2012 2014 2016 2018 2020 2022

In [238]:



5.1.3 mean

In [244]:

```
lst_name=np.array(df_all.columns[1:])
lst_df=[df_1,df_2,df_3,df_4,df_5,df_6,df_7,df_8,df_9]
```

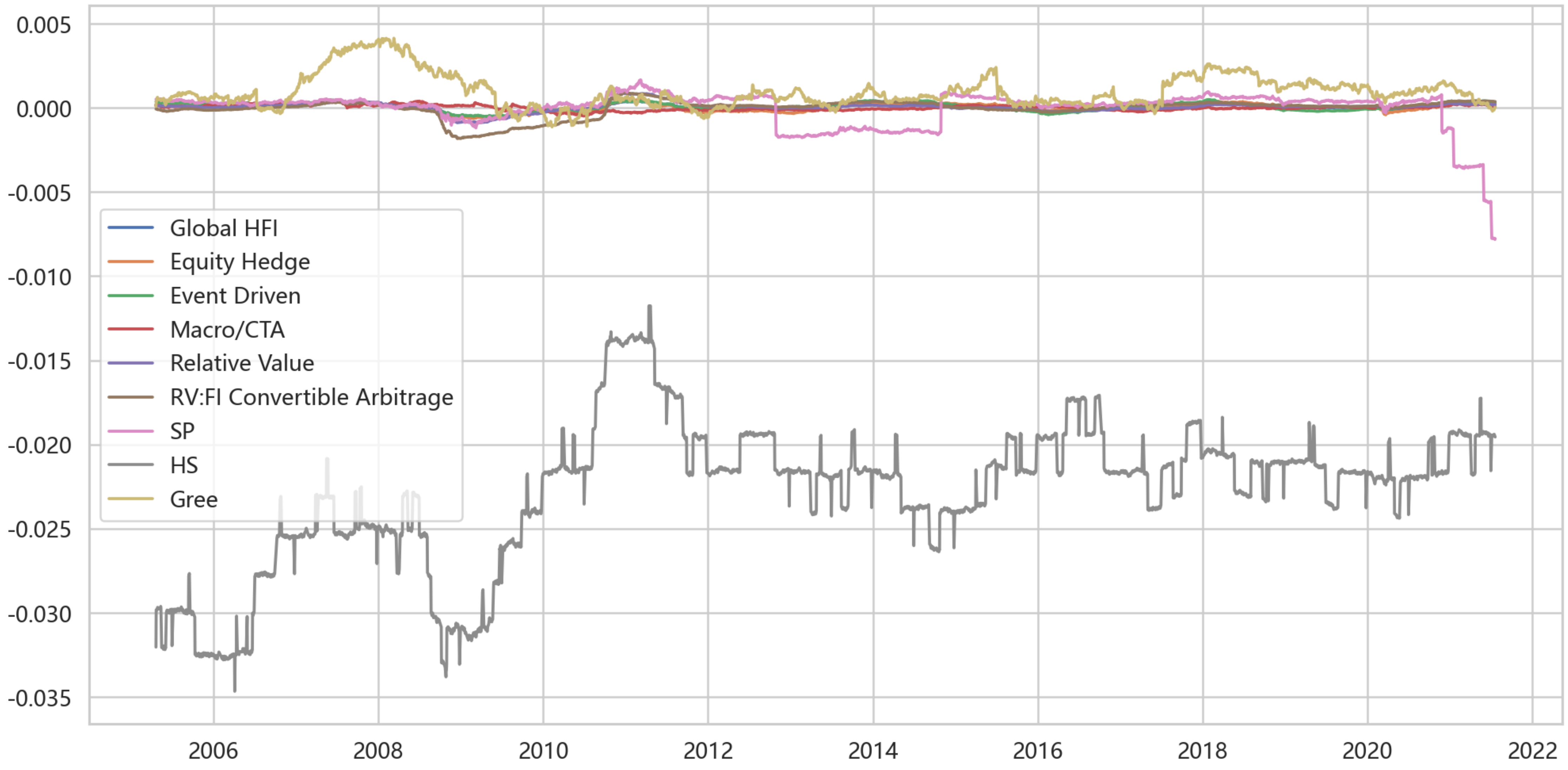
Out[244]:

```
(3738,)
```

In [247]:

```
def chart(col='mean'):
    plt.figure(figsize=(20,10),dpi=120)
    x=df_1.Date
    for i in range(9):
#         print(x.shape,lst_df[i][col].shape)
        plt.plot(x,lst_df[i][col],label=lst_name[i])
#     plt.plot(x,df_1[col],label=)
#     plt.plot(x,df_2[col])
#     plt.plot(x,df_3[col])
#     plt.plot(x,df_4[col])
#     plt.plot(x, df_5[col])
#     plt.plot(x, df_6[col])
#     plt.plot(x, df_7[col])
#     plt.plot(x, df_8[col])
#     plt.plot(x, df_9[col])
    plt.legend()
```

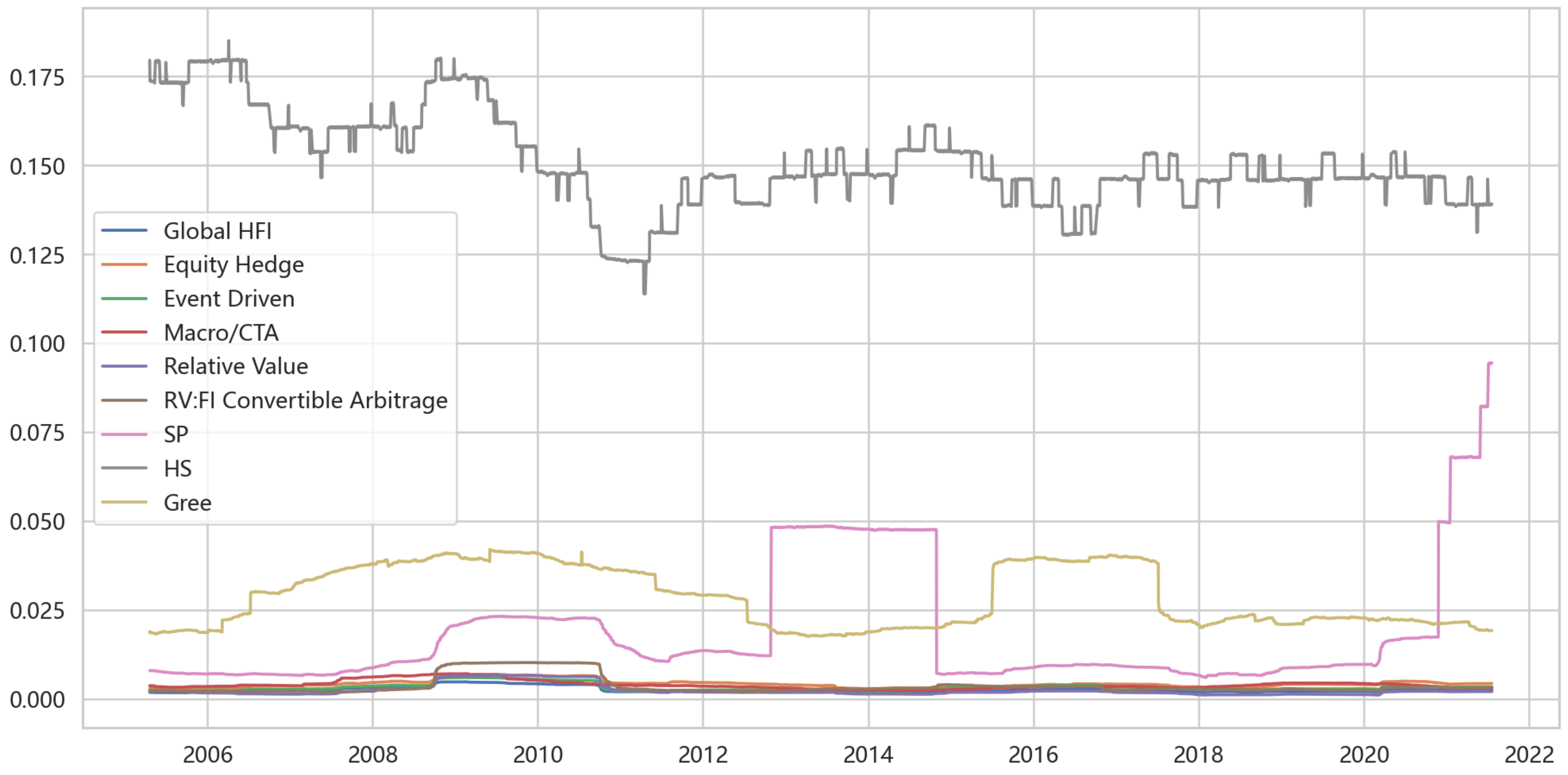
In [253]:



5.1.4 std

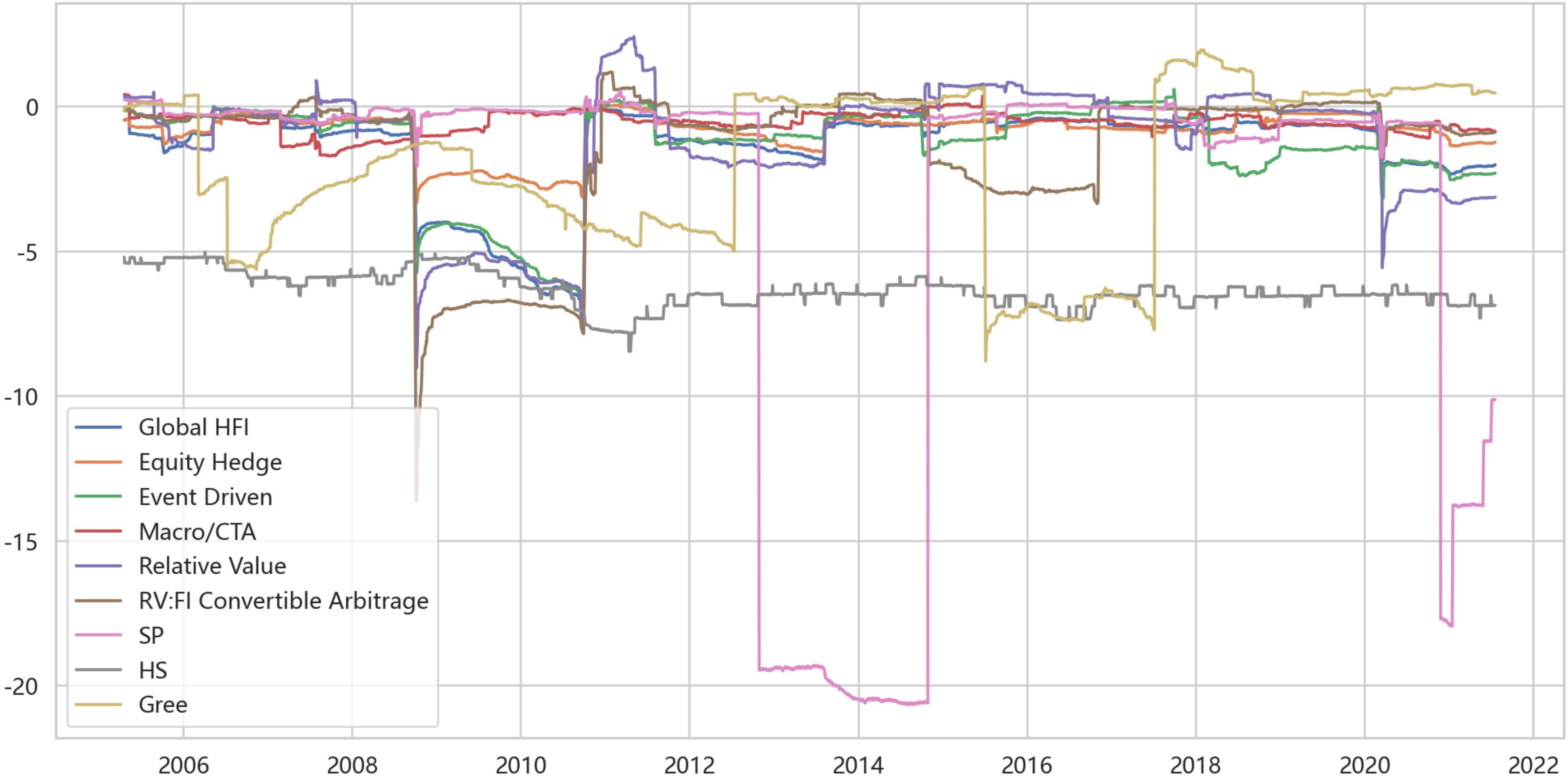
In [249]:

```
[illegible]
```



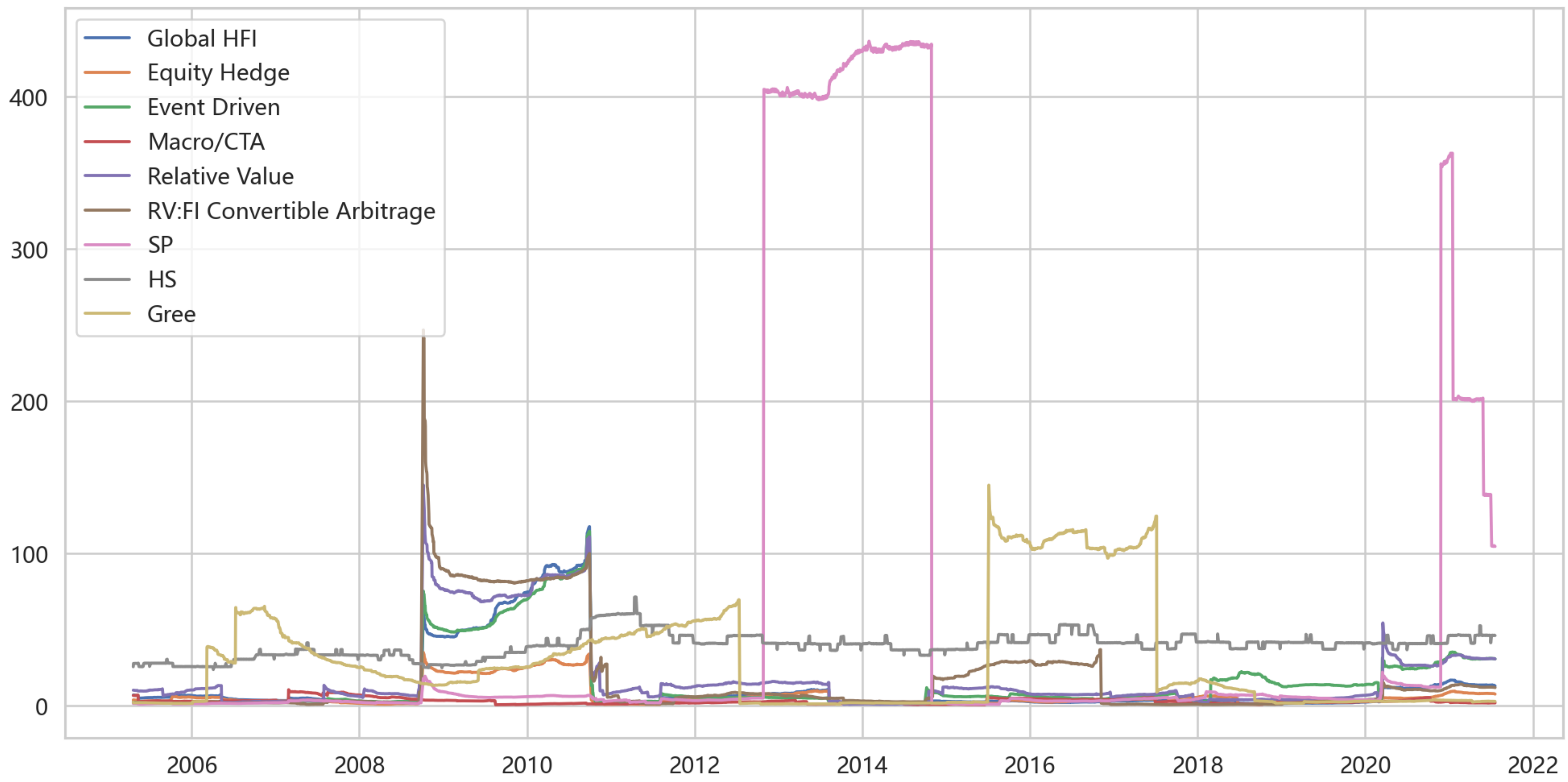
5.1.5 skew

In [250]:



5.1.6 kurt

In [251]:



6 Alphas and Betas

In [316]:

```
import statsmodels.api as sm
def regression(col=9,price=False):
    #statsmodels中的线性模型大致分为两种：
    #基于数组的（array-based），
    #和基于公式的（formula-based）
    if price ==True:
        df_all=df_all_price
    y_SP = df_all.iloc[:, 7]
    y_HS=df_all.iloc[:, 8]
    x = df_all.iloc[:, col]
    x = sm.add_constant(x) # 若模型中有截距，必须有这一步
    model_1 = sm.OLS(y_SP, x).fit() # 构建最小二乘模型并拟合
    model_2 = sm.OLS(y_HS, x).fit() # 构建最小二乘模型并拟合
    return model_1, model_2
```

In [270]:
```
m9_1,m9_2=regression()
# print(m9_1.summary())
# print(m9_2.summary())
```

Out[270]:
```
const   -0.000751
Gree     0.047244
dtype: float64
```

In [317]:
```
def alpha_beta(price=False):
    df_temp=pd.DataFrame()
    alphas_1=[]
    alphas_2=[]
    betas_1=[]
    betas_2=[]
    for i in range(1,8):
        if i==7:
            i=9
        m_1,m_2=regression(i,price)
        alphas_1.append(m_1.params[0])
        betas_1.append(m_1.params[1])
        alphas_2.append(m_2.params[0])
        betas_2.append(m_2.params[1])
    df_temp['name']=np.array(lst_name)[[0,1,2,3,4,5,8]]
    print(lst_name)
    print(alphas_1)
    df_temp['alpha_SP']=alphas_1
    df_temp['beta_SP']=betas_1
    df_temp['alpha_HS']=alphas_2
    df_temp['beta_HS']=betas_2
```

6.1 return

In [288]:
```
df_alpha_beta=alpha_beta()
df_alpha_beta.to_csv('alpha_beta.csv',index=False)
```

```
['Global HFI' 'Equity Hedge' 'Event Driven' 'Macro/CTA' 'Relative Value'
 'RV:FI Convertible Arbitrage' 'SP' 'HS' 'Gree' 'enumerate']
[-0.0009522275652900272, -0.0009072669159627 95, -0.0010191619805141592, -0.0007170031167052903, -0.0008096769720295743, -0.0007124812
253831038, -0.0007508437554679121]
```

Out[288]:

	name	alpha_SP	beta_SP	alpha_HS	beta_HS
0	Global HFI	-0.000952	2.989255	-0.023375	2.385513
1	Equity Hedge	-0.000907	2.109623	-0.023311	1.390624
2	Event Driven	-0.001019	2.246352	-0.023429	1.796787

	name	alpha_SP	beta_SP	alpha_HS	beta_HS
3	Macro/CTA	-0.000717	0.228308	-0.023182	0.036259
4	Relative Value	-0.000810	1.350105	-0.023339	2.126690

6.2 price

In [318]:

```
df_alpha_beta_price=alpha_beta(price=True)
df_alpha_beta_price.to_csv('alpha_beta_Price.csv',index=False)
```

```
['Global HFI' 'Equity Hedge' 'Event Driven' 'Macro/CTA' 'Relative Value'
 'RV:FI Convertible Arbitrage' 'SP' 'HS' 'Gree' 'enumerate']
[-6160.634892524356, -3393.9556638003937, -3875.663262529208, 3158.8623757466876, -4710.647537253917, 1474.7348805855395, 674.1617761
491389]
```

Out[318]:

	name	alpha_SP	beta_SP	alpha_HS	beta_HS
0	Global HFI	-6160.634893	6.632036	-36235.773139	48.024614
1	Equity Hedge	-3393.955664	4.422275	-15283.467380	31.243179
2	Event Driven	-3875.663263	3.967166	-17778.378229	27.394202
3	Macro/CTA	3158.862376	-1.136742	7663.311715	11.662946
4	Relative Value	-4710.647537	5.671904	-22913.796150	38.617616
5	RV:FI Convertible Arbitrage	1474.734881	0.411174	26073.818543	-5.601350
6	Gree	674.161776	39.684205	13932.710591	263.785059

7 Correlation

7.1 HFRX

In [310]:

```
df_HFRX_return=df_all.iloc[:,1:7]
```

Out[310]:

	Global HFI	Equity Hedge	Event Driven	Macro/CTA	Relative Value	RV:FI Convertible Arbitrage
date						
2003-04-01	0.002000	0.003760	0.000450	0.001510	0.000730	0.000540
2003-04-02	0.002226	0.004533	0.002579	0.000989	0.000060	0.000670
2003-04-03	0.000737	0.001418	0.001834	-0.003491	-0.000470	0.001938

	Global HFI	Equity Hedge	Event Driven	Macro/CTA	Relative Value	RV:FI Convertible Arbitrage
date						
2003-04-04	-0.000010	-0.001535	0.001254	0.000180	0.001300	0.001844
2003-04-07	-0.000896	-0.003928	0.001153	-0.002942	0.004902	0.001264
...	...	...	...	...	...	...
2021-07-14	-0.000629	-0.001718	-0.001320	0.001325	0.000184	-0.000782
2021-07-15	-0.001628	-0.002345	-0.000859	-0.002959	-0.000764	-0.002539
2021-07-16	-0.001288	-0.002879	-0.000039	-0.002196	-0.000184	-0.001187
2021-07-19	-0.005698	-0.009254	-0.003187	-0.009827	-0.001596	-0.003153

In [311]:

```
df_HFRX_price=df_all_price.iloc[:,1:7]
```

Out[311]:

	Global HFI	Equity Hedge	Event Driven	Macro/CTA	Relative Value	RV:FI Convertible Arbitrage
0	1000.00	1000.00	1000.00	1000.00	1000.00	1000.00
1	1002.00	1003.76	1000.45	1001.51	1000.73	1000.54
2	1004.23	1008.31	1003.03	1002.50	1000.79	1001.21
3	1004.97	1009.74	1004.87	999.00	1000.32	1003.15
4	1004.96	1008.19	1006.13	999.18	1001.62	1005.00
...	...	...	...	...	...	...
4306	1431.02	1441.47	1815.23	1246.92	1360.82	996.46
4307	1428.69	1438.09	1813.67	1243.23	1359.78	993.93
4308	1426.85	1433.95	1813.60	1240.50	1359.53	992.75
4309	1418.72	1420.68	1807.82	1228.31	1357.36	989.62
4310	1420.70	1428.96	1806.21	1227.52	1357.27	991.34

4205 rows × 6 columns

7.1.1 相关性矩阵

In [343]:

```
corr1=df_HFRX_return.corr()
corr1.to_csv('corr1.csv')
```

Out[343]:

	Global HFI	Equity Hedge	Event Driven	Macro/CTA	Relative Value	RV:FI Convertible Arbitrage
Global HFI	1.000000	0.905598	0.848660	0.466090	0.674846	0.363431
Equity Hedge	0.905598	1.000000	0.748281	0.243756	0.492142	0.228905

	Global HFI	Equity Hedge	Event Driven	Macro/CTA	Relative Value	RV:FI Convertible Arbitrage
Event Driven	0.848660	0.748281	1.000000	0.177378	0.541754	0.297735
Macro/CTA	0.466090	0.243756	0.177378	1.000000	0.094235	0.006554
Relative Value	0.674846	0.492142	0.541754	0.094235	1.000000	0.556836

In [344]:

```
corr2=df_HFRX_price.corr()
corr2.to_csv('corr2.csv')
```

Out[344]:

	Global HFI	Equity Hedge	Event Driven	Macro/CTA	Relative Value	RV:FI Convertible Arbitrage
Global HFI	1.000000	0.905444	0.861406	0.286802	0.867129	0.369747
Equity Hedge	0.905444	1.000000	0.633211	0.289259	0.674235	0.567127
Event Driven	0.861406	0.633211	1.000000	0.085207	0.810946	0.001556
Macro/CTA	0.286802	0.289259	0.085207	1.000000	-0.085157	-0.175962
Relative Value	0.867129	0.674235	0.810946	-0.085157	1.000000	0.341457
RV:FI Convertible Arbitrage	0.369747	0.567127	0.001556	-0.175962	0.341457	1.000000

7.1.2 correlation histogram

In [332]:

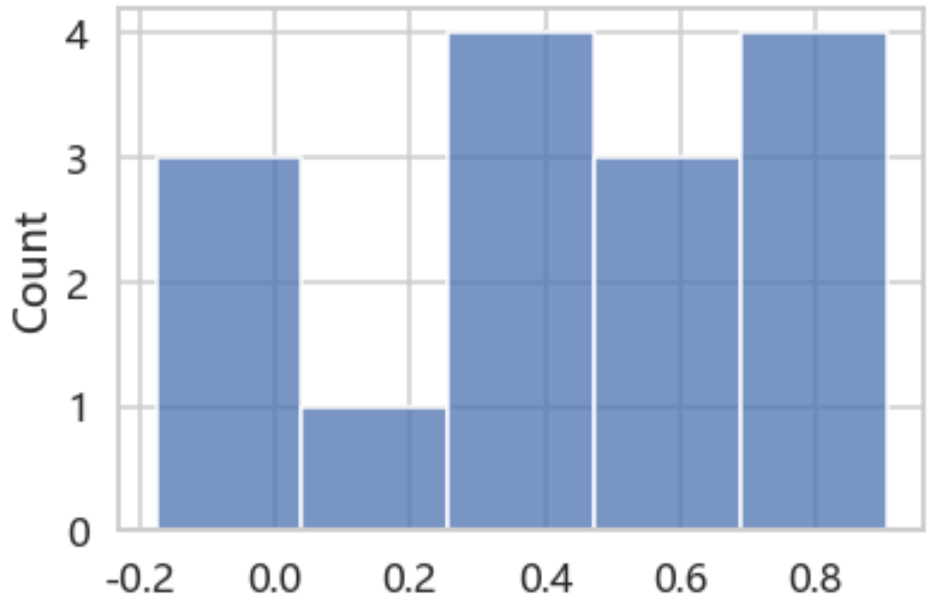
```
array_corr2=np.delete(np.unique(np.array(corr2)),[-1])#按照索引，删除指定元素
array_corr1=np.delete(np.unique(np.array(corr1)),[-1])#按照索引，删除指定元素
```

Out[332]:

```
(array([0.00655375, 0.09423493, 0.17737808, 0.22890456, 0.24375577,
        0.29773505, 0.36343083, 0.46609041, 0.49214164, 0.54175355,
        0.55683598, 0.67484551, 0.74828092, 0.84866041, 0.90559795]),
 array([-0.17596204, -0.08515664,  0.00155606,  0.08520736,  0.28680158,
         0.28925867,  0.34145672,  0.36974709,  0.56712736,  0.63321113,
         0.67423466,  0.81094627,  0.86140562,  0.86712885,  0.90544423]))
```

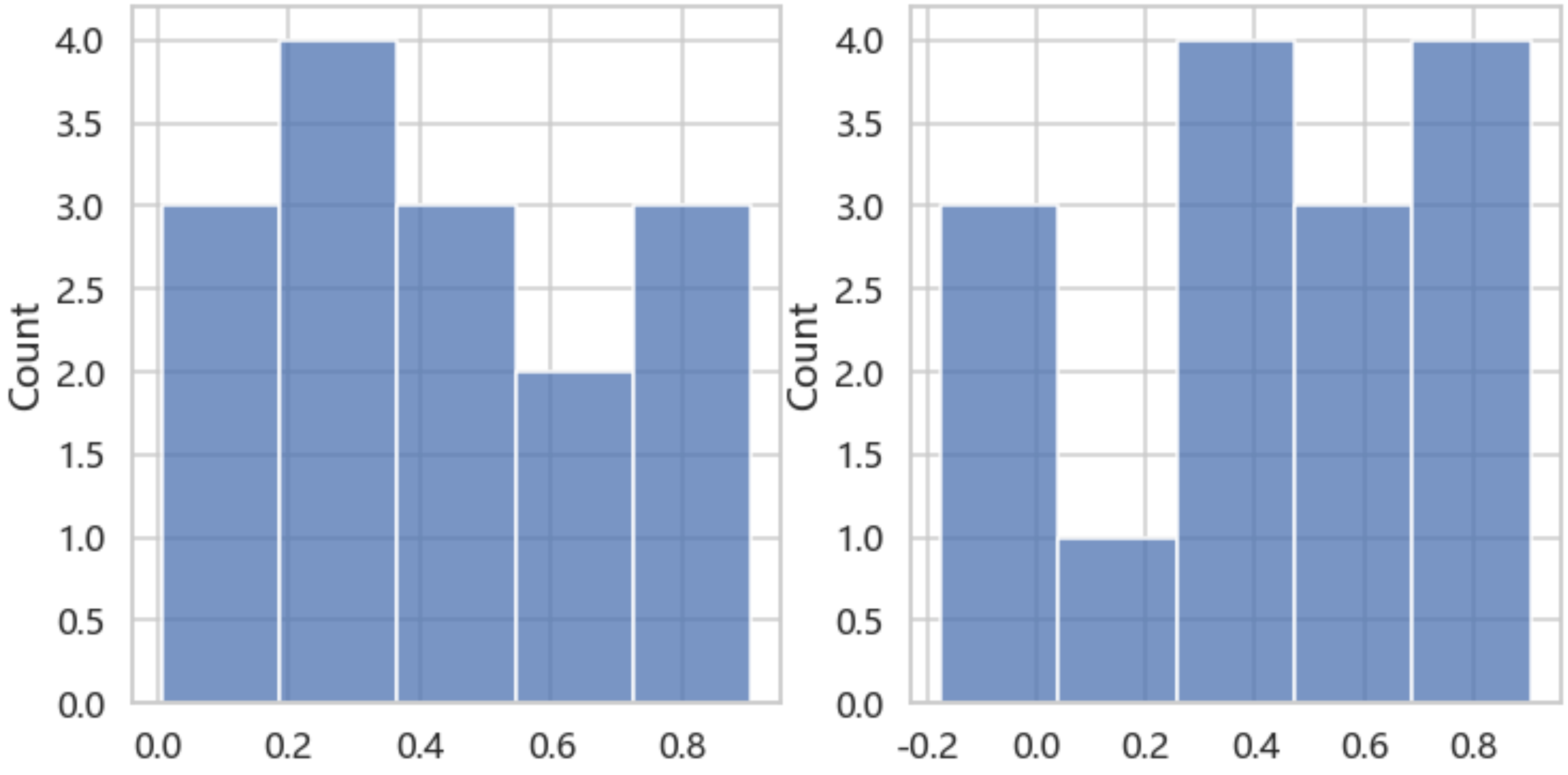
In [346]:

```python
sns.histplot(array_corr2)
```



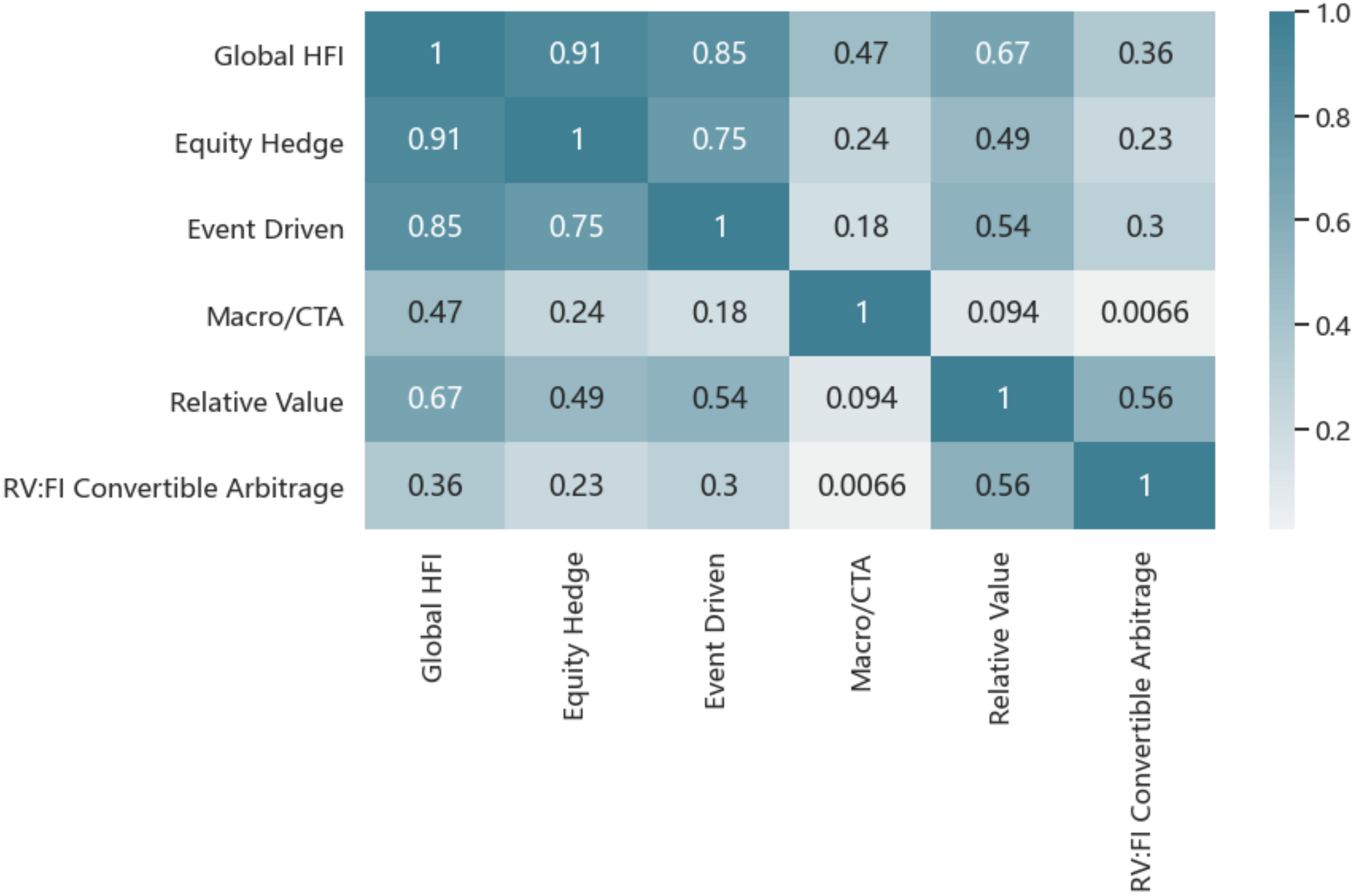
In [342]:

```python
plt.figure(figsize=(12,6))
plt.subplot(121)
sns.histplot(array_corr1)
plt.subplot(122)
sns.histplot(array_corr2)
```

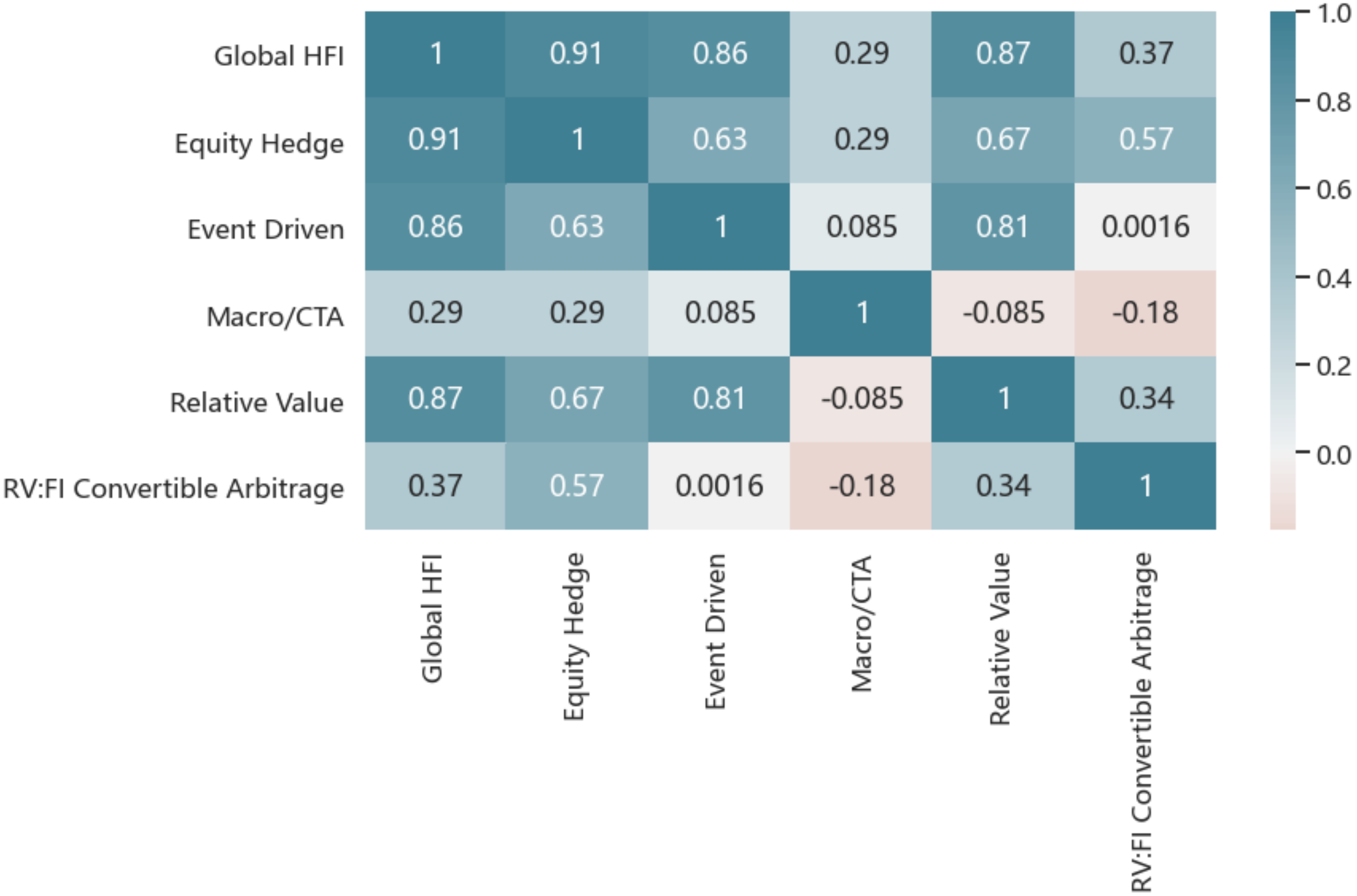


7.1.3 heatmap

In [297]:
```
plt.figure(figsize=(12,6))
sns.heatmap(df_HFRX.corr(),cmap=sns.diverging_palette(20, 220, n=200),
            annot=True, # 注入数据
            center = 0) # 绘制有色数据时将色彩映射居中的值)
```



In [314]:
```
plt.figure(figsize=(12,6))
sns.heatmap(df_HFRX_price.corr(),cmap=sns.diverging_palette(20, 220, n=200),
            annot=True,  # 注入数据
            center = 0)  # 绘制有色数据时将色彩映射居中的值）
```



7.2 Equity Indices

In [301]:
```
df_equity_return=df_all.iloc[:,7:-1]
```

Out[301]:
```
            SP        HS       Gree
date
```

	SP	HS	Gree
date			
2003-04-01	0.012144	-0.004350	0.000000
2003-04-02	0.026116	0.012714	0.003341
2003-04-03	-0.005052	-0.006665	-0.012209
2003-04-04	0.002738	0.020153	0.002247
2003-04-07	0.001229	0.015841	0.001121
...	...	...	...
2021-07-14	0.001165	-0.006292	0.004049
2021-07-15	-0.003262	0.007515	-0.002016
2021-07-16	-0.007539	0.000300	0.009899
2021-07-19	-0.015870	-0.018386	0.029806

In [303]:

Out[303]:

	SP	HS	Gree
SP	1.000000	0.004176	0.037078
HS	0.004176	1.000000	0.048601
Gree	0.037078	0.048601	1.000000

In [308]:

```
df_equity_price=df_all_price.iloc[:,7:]
```

Out[308]:

	SP	HS	Gree
0	848.18	8634.45	8.98
1	858.48	8596.89	8.98
2	880.90	8706.19	9.01
3	876.45	8648.16	8.90
4	878.85	8822.45	8.92
...	...	...	...
4306	4374.30	27787.46	49.60
4307	4360.03	27996.27	49.50
4308	4327.16	28004.68	49.99
4309	4258.49	27489.78	51.48
4310	4323.06	27259.25	51.20

4205 rows × 3 columns

In [309]:

```
[illegible]
```

Out[309]:

	SP	HS	Gree
SP	1.000000	0.748537	0.810837
HS	0.748537	1.000000	0.818348
Gree	0.810837	0.818348	1.000000

In [ ]:

Contents