# S&P500 (Long term historical data_both S&P500 and KOSPI)

June 11, 2025

[1]:

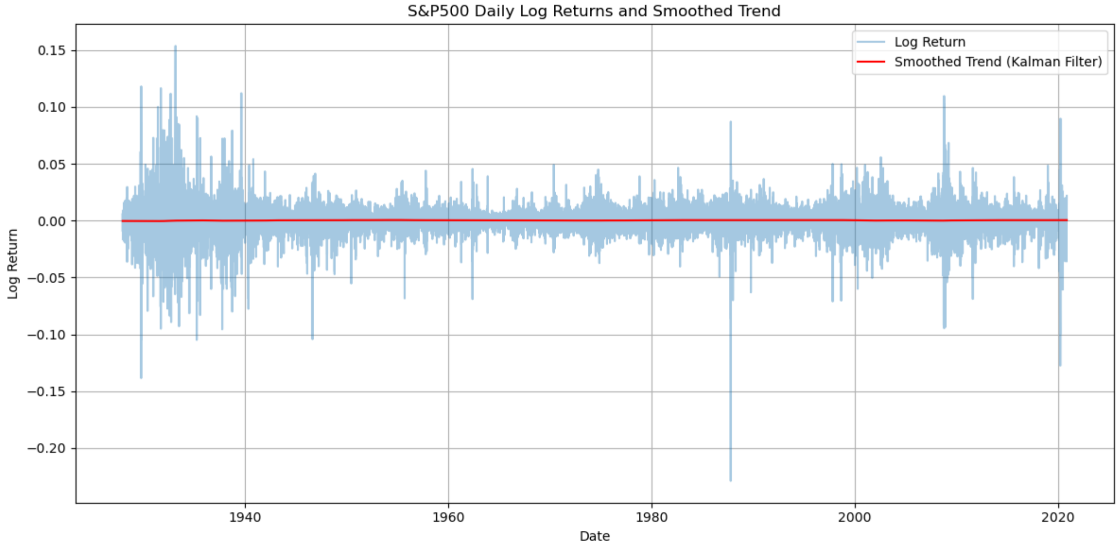
```
import yfinance as yf
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
```

[2]:

```
data = pd.read_csv("/Users/jiminbyun/Downloads/SPX.csv")

data['Date'] = pd.to_datetime(data['Date'])
data['LogReturn'] = np.log(data['Close'] / data['Close'].shift(1))
log_returns = data[['Date', 'LogReturn']].dropna().
  ↪set_index('Date')['LogReturn']

mod = sm.tsa.UnobservedComponents(log_returns, level='local level')
res_fit = mod.fit()

results_df = pd.DataFrame({
    'LogReturn': log_returns,
    'SmoothedTrend': res_fit.smoothed_state[0]
})

plt.figure(figsize=(12, 6))
plt.plot(results_df.index, results_df['LogReturn'], label='Log Return', alpha=0.
  ↪4)
plt.plot(results_df.index, results_df['SmoothedTrend'], label='Smoothed Trend␣
  ↪(Kalman Filter)', color='red')
plt.title("S&P500 Daily Log Returns and Smoothed Trend")
plt.xlabel("Date")
plt.ylabel("Log Return")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

```
/Users/jiminbyun/anaconda3/lib/python3.11/site-
packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: A date index has
been provided, but it has no associated frequency information and so will be
ignored when e.g. forecasting.
```

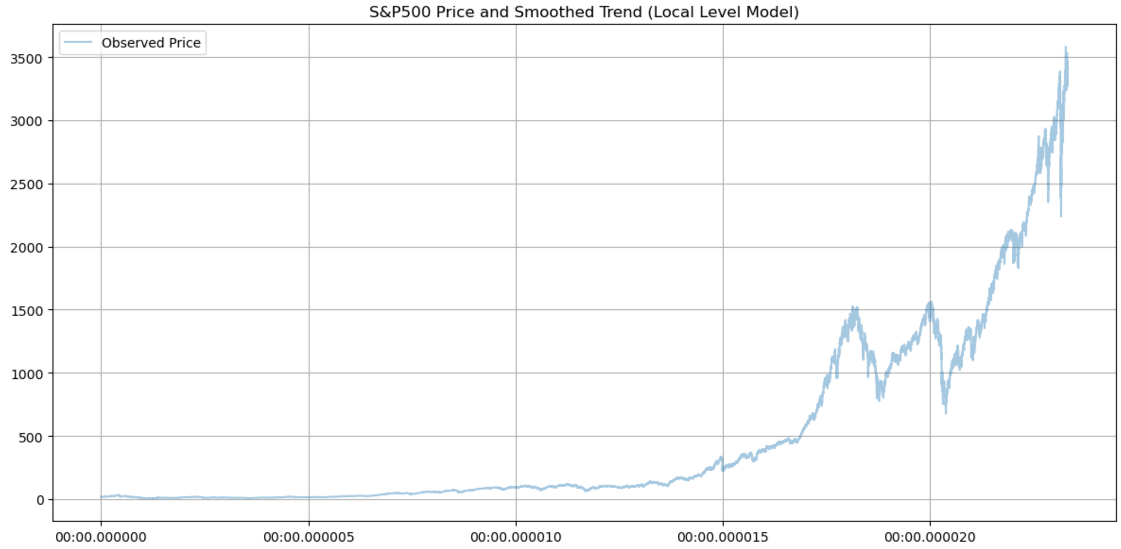
```
  self._init_dates(dates, freq)
 This problem is unconstrained.

RUNNING THE L-BFGS-B CODE

           * * *

Machine precision = 2.220D-16
 N =            2     M =           10

At X0         0 variables are exactly at the bounds

At iterate    0    f= -2.94961D+00    |proj g|=  2.09851D+01

At iterate    5    f= -3.00143D+00    |proj g|=  5.07147D+00


 Bad direction in the line search;
   refresh the lbfgs memory and restart the iteration.

 Line search cannot locate an adequate point after MAXLS
  function and gradient evaluations.
  Previous x, f and g restored.
 Possible causes: 1 error in function or gradient evaluation;
                  2 rounding error dominate computation.
/Users/jiminbyun/anaconda3/lib/python3.11/site-
packages/statsmodels/base/model.py:607: ConvergenceWarning: Maximum Likelihood
optimization failed to converge. Check mle_retvals
  warnings.warn("Maximum Likelihood optimization failed to "


           * * *

Tit   = total number of iterations
Tnf   = total number of function evaluations
Tnint = total number of segments explored during Cauchy searches
Skip  = number of BFGS updates skipped
Nact  = number of active bounds at final generalized Cauchy point
Projg = norm of the final projected gradient
F     = final function value

           * * *

   N    Tit     Tnf  Tnint  Skip  Nact     Projg        F
    2      7     59      2     0     0   4.107D+00  -3.001D+00
  F =  -3.0014276540607971

ABNORMAL_TERMINATION_IN_LNSRCH
```



[3]:
```
data.index = pd.to_datetime(data.index)

data['LogReturn'] = np.log(data['Close'] / data['Close'].shift(1))
log_returns = data['LogReturn'].dropna()
price = data['Close'].dropna()

mod = sm.tsa.UnobservedComponents(log_returns, level='local level')
res_fit = mod.fit()

plt.figure(figsize=(12, 6))
plt.plot(price.index, price, label='Observed Price', alpha=0.4)
plt.title("S&P500 Price and Smoothed Trend (Local Level Model)")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

```
/Users/jiminbyun/anaconda3/lib/python3.11/site-
packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: No frequency
information was provided, so inferred frequency N will be used.
  self._init_dates(dates, freq)
 This problem is unconstrained.

RUNNING THE L-BFGS-B CODE

           * * *

Machine precision = 2.220D-16
 N =            2     M =           10
```

```
At X0         0 variables are exactly at the bounds

At iterate    0    f= -2.94961D+00    |proj g|=  2.09851D+01

At iterate    5    f= -3.00143D+00    |proj g|=  5.07147D+00


 Bad direction in the line search;
   refresh the lbfgs memory and restart the iteration.
/Users/jiminbyun/anaconda3/lib/python3.11/site-
packages/statsmodels/base/model.py:607: ConvergenceWarning: Maximum Likelihood
optimization failed to converge. Check mle_retvals
  warnings.warn("Maximum Likelihood optimization failed to "


           * * *

Tit   = total number of iterations
Tnf   = total number of function evaluations
Tnint = total number of segments explored during Cauchy searches
Skip  = number of BFGS updates skipped
Nact  = number of active bounds at final generalized Cauchy point
Projg = norm of the final projected gradient
F     = final function value

           * * *

   N    Tit     Tnf  Tnint  Skip  Nact     Projg        F
    2      7     59      2     0     0   4.107D+00  -3.001D+00
  F =  -3.0014276540607971

ABNORMAL_TERMINATION_IN_LNSRCH


 Line search cannot locate an adequate point after MAXLS
  function and gradient evaluations.
  Previous x, f and g restored.
 Possible causes: 1 error in function or gradient evaluation;
                  2 rounding error dominate computation.
```



[4]:

```python
mod_price = sm.tsa.UnobservedComponents(price, level='local level')
res_price = mod_price.fit()

fitted_trend = pd.Series(res_price.smoothed_state[0], index=price.index)
residuals = price - fitted_trend
residuals.name = 'residual'

log_price_trend = np.cumsum(fitted_trend)

initial_price = data['Close'].iloc[1]
reconstructed_price = np.exp(log_price_trend) * initial_price
reconstructed_price = pd.Series(reconstructed_price, index=data.index[1:])

residuals_price = data['Close'].iloc[1:] - reconstructed_price
residuals_pct = residuals_price / reconstructed_price
```

```
/Users/jiminbyun/anaconda3/lib/python3.11/site-
packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: No frequency
information was provided, so inferred frequency N will be used.
  self._init_dates(dates, freq)
 This problem is unconstrained.

RUNNING THE L-BFGS-B CODE

           * * *

Machine precision = 2.220D-16
 N =            2     M =           10
```

```
At X0         0 variables are exactly at the bounds

At iterate    0    f=  7.50667D+00    |proj g|=  1.37615D-03

At iterate    5    f=  3.73290D+00    |proj g|=  6.42940D-03

           * * *

Tit   = total number of iterations
Tnf   = total number of function evaluations
Tnint = total number of segments explored during Cauchy searches
Skip  = number of BFGS updates skipped
Nact  = number of active bounds at final generalized Cauchy point
Projg = norm of the final projected gradient
F     = final function value

           * * *

   N    Tit     Tnf  Tnint  Skip  Nact     Projg        F
    2      9     28      1     0     0   3.370D-06   3.730D+00
  F =   3.7304359633794708

CONVERGENCE: NORM_OF_PROJECTED_GRADIENT_<=_PGTOL
```

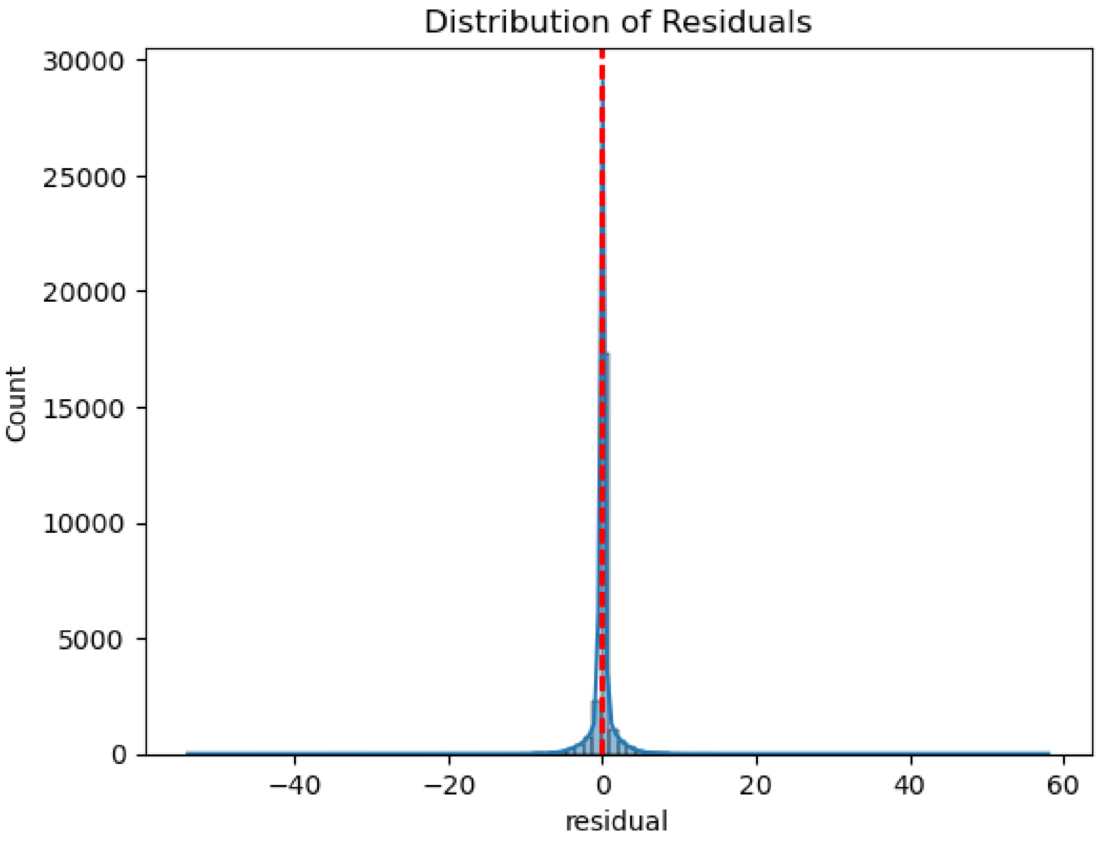
```
/Users/jiminbyun/anaconda3/lib/python3.11/site-
packages/pandas/core/arraylike.py:402: RuntimeWarning: overflow encountered in
exp
  result = getattr(ufunc, method)(*inputs, **kwargs)
```

[5]:

```python
threshold = 0.03
outliers = residuals_pct[np.abs(residuals_pct) > threshold]
```

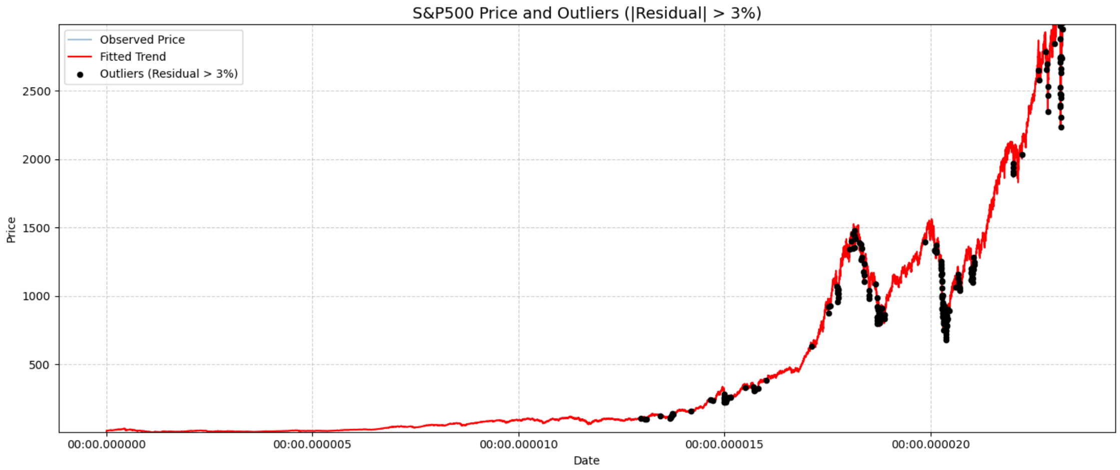
[6]:

```python
import seaborn as sns
sns.histplot(residuals, bins=100, kde=True)
plt.axvline(threshold, color='r', linestyle='--')
plt.axvline(-threshold, color='r', linestyle='--')
plt.title("Distribution of Residuals")
plt.show()
```



[7]:

```
plt.figure(figsize=(14, 6))
plt.plot(price.index, price, label='Observed Price', color='steelblue', alpha=0.
 ↪5)
plt.plot(price.index, res_price.fittedvalues, label='Fitted Trend',␣
 ↪color='red', linewidth=1.5)

residuals = price - res_price.fittedvalues
residuals_pct = residuals / price
valid_range = price[price > 100]
residuals_pct = (residuals / price).loc[valid_range.index]

threshold = 0.03
outliers = residuals_pct[np.abs(residuals_pct) > threshold]

plt.scatter(outliers.index, price.loc[outliers.index],
            color='black', label='Outliers (Residual > 3%)', s=20, zorder=5)

plt.title("S&P500 Price and Outliers (|Residual| > 3%)", fontsize=14)
plt.xlabel("Date")
plt.ylabel("Price")
plt.legend()
plt.grid(True, linestyle='--', alpha=0.6)
plt.tight_layout()
plt.ylim(price.quantile(0.01), price.quantile(0.99))
plt.show()
```



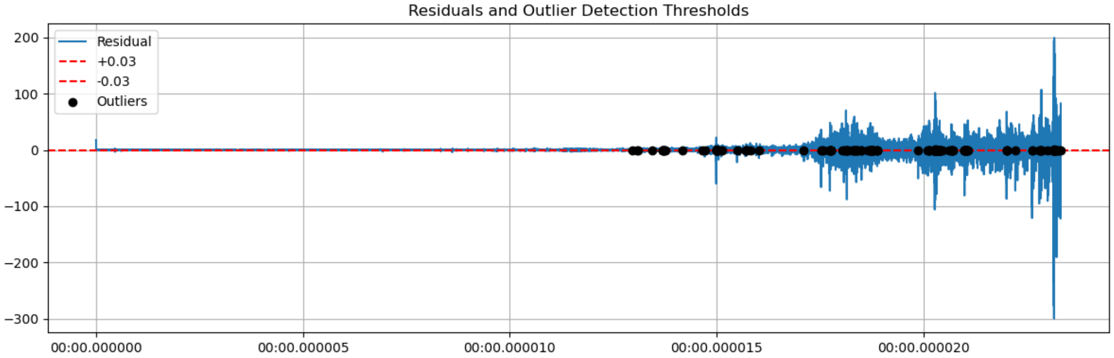
[8]:

```
price = price.squeeze()  # (1-column DataFrame → Series)

residuals = price - res_price.fittedvalues
residuals.name = 'Residual'

threshold = 0.03
outliers = residuals_pct[np.abs(residuals_pct) > threshold]

outlier_table = pd.DataFrame({
    'Price': price.loc[outliers.index],
    'Residual': outliers
})

print(outlier_table.head())
```

```
                                    Price  Residual
1970-01-01 00:00:00.000012964  106.629997 -0.031991
1970-01-01 00:00:00.000013074  102.260002 -0.031466
1970-01-01 00:00:00.000013099  103.430000  0.034055
1970-01-01 00:00:00.000013437  125.500000 -0.031207
1970-01-01 00:00:00.000013685  109.040001  0.045902
```

[9]:

```
plt.figure(figsize=(12, 4))
plt.plot(residuals.index, residuals, label='Residual')
plt.axhline(threshold, color='red', linestyle='--', label='+0.03')
plt.axhline(-threshold, color='red', linestyle='--', label='-0.03')
plt.scatter(outliers.index, outliers, color='black', zorder=5, label='Outliers')
plt.title("Residuals and Outlier Detection Thresholds")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```



[10]:

```
outlier_info = pd.DataFrame({
    'Date': outliers.index,
    'Price': price.loc[outliers.index],
    'Residual': residuals.loc[outliers.index]
}).reset_index(drop=True)

pd.set_option('display.max_rows', None)
print(outlier_info)
```

```
                             Date       Price   Residual
0   1970-01-01 00:00:00.000012964  106.629997  -3.411235
1   1970-01-01 00:00:00.000013074  102.260002  -3.217708
2   1970-01-01 00:00:00.000013099  103.430000   3.522277
3   1970-01-01 00:00:00.000013437  125.500000  -3.916517
4   1970-01-01 00:00:00.000013685  109.040001   5.005138
5   1970-01-01 00:00:00.000013688  113.019997   3.943441
6   1970-01-01 00:00:00.000013689  116.110001   3.604246
7   1970-01-01 00:00:00.000013720  125.970001   4.046491
8   1970-01-01 00:00:00.000013733  133.320007  -5.536932
9   1970-01-01 00:00:00.000013740  142.869995   5.672996
10  1970-01-01 00:00:00.000013758  138.529999   4.259953
11  1970-01-01 00:00:00.000014183  162.350006   4.928842
12  1970-01-01 00:00:00.000014667  244.050003  -7.844382
13  1970-01-01 00:00:00.000014714  235.179993 -11.973536
14  1970-01-01 00:00:00.000014992  282.700012 -16.459597
15  1970-01-01 00:00:00.000014993  224.839996 -60.006418
```

```
16  1970-01-01 00:00:00.000014995   258.380005   22.093125
17  1970-01-01 00:00:00.000014998   227.669998  -20.677186
18  1970-01-01 00:00:00.000015001   244.770004   11.549757
19  1970-01-01 00:00:00.000015002   251.789993    8.526127
20  1970-01-01 00:00:00.000015008   243.169998   -7.682241
21  1970-01-01 00:00:00.000015022   230.300003  -10.561383
22  1970-01-01 00:00:00.000015025   225.210007   -8.045416
23  1970-01-01 00:00:00.000015045   255.940002    8.809824
24  1970-01-01 00:00:00.000015049   243.399994  -17.372798
25  1970-01-01 00:00:00.000015116   259.750000  -11.779402
26  1970-01-01 00:00:00.000015148   262.160004    8.598260
27  1970-01-01 00:00:00.000015496   333.649994  -21.986177
28  1970-01-01 00:00:00.000015700   334.429993  -11.363130
29  1970-01-01 00:00:00.000015713   307.059998  -10.295618
30  1970-01-01 00:00:00.000015715   321.440002   10.335213
31  1970-01-01 00:00:00.000015739   314.940002    9.476622
32  1970-01-01 00:00:00.000015814   327.970001   12.133485
33  1970-01-01 00:00:00.000016025   382.619995  -14.544430
34  1970-01-01 00:00:00.000017114   633.500000  -19.986272
35  1970-01-01 00:00:00.000017528   876.989990  -66.136090
36  1970-01-01 00:00:00.000017529   921.849976   36.235556
37  1970-01-01 00:00:00.000017579   927.690002  -29.463066
38  1970-01-01 00:00:00.000017721  1072.119995  -41.734790
39  1970-01-01 00:00:00.000017738  1042.589966  -42.637127
40  1970-01-01 00:00:00.000017740   957.280029  -72.599782
41  1970-01-01 00:00:00.000017745  1023.460022   48.338314
42  1970-01-01 00:00:00.000017756  1066.089966   37.268459
43  1970-01-01 00:00:00.000017761  1017.010010  -31.900817
44  1970-01-01 00:00:00.000017762   986.390015  -34.779999
45  1970-01-01 00:00:00.000017772  1047.489990   43.345976
46  1970-01-01 00:00:00.000018033  1342.439941   47.448008
47  1970-01-01 00:00:00.000018079  1399.420044  -57.544271
48  1970-01-01 00:00:00.000018111  1346.089966  -42.311089
49  1970-01-01 00:00:00.000018129  1458.469971   70.189918
50  1970-01-01 00:00:00.000018150  1356.560059  -88.006240
51  1970-01-01 00:00:00.000018152  1441.609985   44.526003
52  1970-01-01 00:00:00.000018156  1477.439941   47.071405
53  1970-01-01 00:00:00.000018180  1422.449951   43.723114
54  1970-01-01 00:00:00.000018280  1388.760010   45.200860
55  1970-01-01 00:00:00.000018312  1376.540039   52.786386
56  1970-01-01 00:00:00.000018323  1264.739990  -42.984823
57  1970-01-01 00:00:00.000018330  1283.270020  -38.703397
58  1970-01-01 00:00:00.000018331  1347.560059   59.242950
59  1970-01-01 00:00:00.000018377  1180.160034  -57.274041
60  1970-01-01 00:00:00.000018393  1106.459961  -41.104155
61  1970-01-01 00:00:00.000018395  1151.439941   47.072361
62  1970-01-01 00:00:00.000018403  1238.160034   47.906347
63  1970-01-01 00:00:00.000018504  1038.770020  -53.295689
```

```
64  1970-01-01 00:00:00.000018507   984.539978  -33.950656
65  1970-01-01 00:00:00.000018509  1003.450012   34.628908
66  1970-01-01 00:00:00.000018665  1088.849976   38.567440
67  1970-01-01 00:00:00.000018705   989.030029   35.413230
68  1970-01-01 00:00:00.000018708   920.469971  -35.635688
69  1970-01-01 00:00:00.000018715   847.750000  -36.954457
70  1970-01-01 00:00:00.000018716   819.849976  -32.719044
71  1970-01-01 00:00:00.000018717   797.700012  -26.416668
72  1970-01-01 00:00:00.000018718   843.429993   42.285134
73  1970-01-01 00:00:00.000018721   898.960022   47.979519
74  1970-01-01 00:00:00.000018726   834.599976  -32.738813
75  1970-01-01 00:00:00.000018729   905.460022   31.284980
76  1970-01-01 00:00:00.000018733   919.619995   32.789150
77  1970-01-01 00:00:00.000018746   878.020020  -38.317990
78  1970-01-01 00:00:00.000018758   843.320007  -26.965555
79  1970-01-01 00:00:00.000018764   827.369995  -25.275123
80  1970-01-01 00:00:00.000018766   847.909973   30.623550
81  1970-01-01 00:00:00.000018773   803.919983   24.504468
82  1970-01-01 00:00:00.000018774   835.320007   34.595512
83  1970-01-01 00:00:00.000018776   881.270020   41.216398
84  1970-01-01 00:00:00.000018830   909.030029   29.301340
85  1970-01-01 00:00:00.000018878   831.900024   28.000661
86  1970-01-01 00:00:00.000018880   862.789978   30.174770
87  1970-01-01 00:00:00.000018885   864.229980  -28.889223
88  1970-01-01 00:00:00.000019874  1399.040039  -50.658756
89  1970-01-01 00:00:00.000020099  1333.250000  -41.532183
90  1970-01-01 00:00:00.000020111  1336.640015  -45.748000
91  1970-01-01 00:00:00.000020135  1320.650024   44.422103
92  1970-01-01 00:00:00.000020140  1330.739990   52.183339
93  1970-01-01 00:00:00.000020149  1370.180054   48.239010
94  1970-01-01 00:00:00.000020258  1236.829956  -38.612043
95  1970-01-01 00:00:00.000020261  1224.510010  -39.949764
96  1970-01-01 00:00:00.000020265  1192.699951  -58.360036
97  1970-01-01 00:00:00.000020267  1156.390015  -55.476937
98  1970-01-01 00:00:00.000020268  1206.510010   42.885564
99  1970-01-01 00:00:00.000020269  1255.079956   54.162408
100 1970-01-01 00:00:00.000020270  1207.089966  -40.926979
101 1970-01-01 00:00:00.000020275  1106.420044 -105.932444
102 1970-01-01 00:00:00.000020276  1166.359985   46.125895
103 1970-01-01 00:00:00.000020278  1114.280029  -46.686779
104 1970-01-01 00:00:00.000020280  1056.890015  -45.096478
105 1970-01-01 00:00:00.000020281   996.229980  -66.540810
106 1970-01-01 00:00:00.000020283   909.919983  -77.623826
107 1970-01-01 00:00:00.000020285  1003.349976  101.414662
108 1970-01-01 00:00:00.000020287   907.840027  -89.141753
109 1970-01-01 00:00:00.000020290   985.400024   44.541814
110 1970-01-01 00:00:00.000020292   896.780029  -61.470288
111 1970-01-01 00:00:00.000020294   876.770020  -30.907810
```

```
112 1970-01-01 00:00:00.000020295   848.919983  -31.880548
113 1970-01-01 00:00:00.000020296   940.510010   87.432666
114 1970-01-01 00:00:00.000020301  1005.750000   39.433322
115 1970-01-01 00:00:00.000020302   952.770020  -47.837705
116 1970-01-01 00:00:00.000020303   904.880005  -54.128257
117 1970-01-01 00:00:00.000020307   852.299988  -49.450090
118 1970-01-01 00:00:00.000020308   911.289978   52.541486
119 1970-01-01 00:00:00.000020309   873.289978  -31.148364
120 1970-01-01 00:00:00.000020310   850.750000  -26.601859
121 1970-01-01 00:00:00.000020312   806.580017  -51.900867
122 1970-01-01 00:00:00.000020313   752.440002  -60.908111
123 1970-01-01 00:00:00.000020314   800.030029   39.647348
124 1970-01-01 00:00:00.000020315   851.809998   56.950154
125 1970-01-01 00:00:00.000020317   887.679993   31.986090
126 1970-01-01 00:00:00.000020319   816.210022  -78.369775
127 1970-01-01 00:00:00.000020323   876.070007   27.944671
128 1970-01-01 00:00:00.000020324   909.700012   37.274110
129 1970-01-01 00:00:00.000020330   913.179993   43.204656
130 1970-01-01 00:00:00.000020341   931.799988   30.549222
131 1970-01-01 00:00:00.000020349   842.619995  -29.355212
132 1970-01-01 00:00:00.000020352   805.219971  -44.114095
133 1970-01-01 00:00:00.000020353   840.239990   29.267351
134 1970-01-01 00:00:00.000020358   874.090027   29.657754
135 1970-01-01 00:00:00.000020367   827.159973  -42.146082
136 1970-01-01 00:00:00.000020371   789.169983  -38.731855
137 1970-01-01 00:00:00.000020375   743.330017  -28.053340
138 1970-01-01 00:00:00.000020376   773.140015   26.151722
139 1970-01-01 00:00:00.000020380   700.820007  -36.799355
140 1970-01-01 00:00:00.000020383   682.549988  -28.321081
141 1970-01-01 00:00:00.000020386   719.599976   42.127992
142 1970-01-01 00:00:00.000020388   750.739990   30.325916
143 1970-01-01 00:00:00.000020391   778.119995   24.049158
144 1970-01-01 00:00:00.000020395   822.919983   52.226359
145 1970-01-01 00:00:00.000020400   787.530029  -30.279332
146 1970-01-01 00:00:00.000020403   834.380005   25.131319
147 1970-01-01 00:00:00.000020408   856.559998   32.302022
148 1970-01-01 00:00:00.000020414   832.390015  -36.405399
149 1970-01-01 00:00:00.000020424   907.239990   30.360110
150 1970-01-01 00:00:00.000020458   893.039978  -27.693818
151 1970-01-01 00:00:00.000020615  1063.109985  -34.686966
152 1970-01-01 00:00:00.000020678  1128.150024  -39.186047
153 1970-01-01 00:00:00.000020680  1159.729980   45.931521
154 1970-01-01 00:00:00.000020688  1071.589966  -44.488327
155 1970-01-01 00:00:00.000020693  1103.060059   34.295546
156 1970-01-01 00:00:00.000020698  1064.880005  -36.943244
157 1970-01-01 00:00:00.000020715  1041.239990  -33.605675
158 1970-01-01 00:00:00.000020720  1060.270020   32.831408
159 1970-01-01 00:00:00.000020993  1200.069946  -60.023654
```

```
160 1970-01-01 00:00:00.000020995  1119.459961  -81.030734
161 1970-01-01 00:00:00.000020996  1172.530029   42.503312
162 1970-01-01 00:00:00.000020997  1120.760010  -46.227404
163 1970-01-01 00:00:00.000020998  1172.640015   45.851753
164 1970-01-01 00:00:00.000021003  1140.650024  -53.231645
165 1970-01-01 00:00:00.000021006  1162.349976   38.158662
166 1970-01-01 00:00:00.000021026  1166.760010  -35.771608
167 1970-01-01 00:00:00.000021027  1129.560059  -41.864722
168 1970-01-01 00:00:00.000021034  1099.229980  -35.859679
169 1970-01-01 00:00:00.000021039  1194.890015   38.596322
170 1970-01-01 00:00:00.000021052  1284.589966   43.893288
171 1970-01-01 00:00:00.000021055  1218.280029  -39.059723
172 1970-01-01 00:00:00.000021061  1229.099976  -44.766198
173 1970-01-01 00:00:00.000021075  1246.959961   52.675602
174 1970-01-01 00:00:00.000022011  1970.890015  -70.865275
175 1970-01-01 00:00:00.000022012  1893.209961  -86.921191
176 1970-01-01 00:00:00.000022014  1940.510010   68.083559
177 1970-01-01 00:00:00.000022018  1913.849976  -60.361696
178 1970-01-01 00:00:00.000022223  2037.410034  -72.318537
179 1970-01-01 00:00:00.000022629  2648.939941 -121.032269
180 1970-01-01 00:00:00.000022632  2581.000000 -101.900513
181 1970-01-01 00:00:00.000022801  2785.679932  -95.255279
182 1970-01-01 00:00:00.000022811  2656.100098  -86.786695
183 1970-01-01 00:00:00.000022839  2700.060059  -85.986362
184 1970-01-01 00:00:00.000022852  2351.100098  -72.906071
185 1970-01-01 00:00:00.000022853  2467.699951  107.092588
186 1970-01-01 00:00:00.000022859  2531.939941   76.048026
187 1970-01-01 00:00:00.000023005  2844.739990  -90.646208
188 1970-01-01 00:00:00.000023144  3225.889893 -116.674018
189 1970-01-01 00:00:00.000023145  3128.209961 -112.894725
190 1970-01-01 00:00:00.000023147  2978.760010 -141.091077
191 1970-01-01 00:00:00.000023149  3090.229980  130.410589
192 1970-01-01 00:00:00.000023151  3130.120117  117.640769
193 1970-01-01 00:00:00.000023152  3023.939941  -90.839314
194 1970-01-01 00:00:00.000023154  2746.560059 -234.079733
195 1970-01-01 00:00:00.000023155  2882.229980  105.144918
196 1970-01-01 00:00:00.000023156  2741.379883 -127.138748
197 1970-01-01 00:00:00.000023157  2480.639893 -277.319429
198 1970-01-01 00:00:00.000023158  2711.020020  194.216481
199 1970-01-01 00:00:00.000023159  2386.129883 -299.563473
200 1970-01-01 00:00:00.000023160  2529.189941  103.995693
201 1970-01-01 00:00:00.000023161  2398.100098 -117.528358
202 1970-01-01 00:00:00.000023163  2304.919922 -104.996336
203 1970-01-01 00:00:00.000023164  2237.399902  -81.211994
204 1970-01-01 00:00:00.000023165  2447.330078  199.339783
205 1970-01-01 00:00:00.000023167  2630.070068  161.581149
206 1970-01-01 00:00:00.000023171  2470.500000 -118.276110
207 1970-01-01 00:00:00.000023174  2663.679932  170.738877
```

```
208 1970-01-01 00:00:00.000023176  2749.979980   92.916702
209 1970-01-01 00:00:00.000023184  2736.560059  -92.006238
210 1970-01-01 00:00:00.000023203  2953.909912   92.094912
211 1970-01-01 00:00:00.000023220  3002.100098 -190.581886
212 1970-01-01 00:00:00.000023279  3455.060059 -118.275367
213 1970-01-01 00:00:00.000023281  3331.840088 -100.795552
214 1970-01-01 00:00:00.000023317  3271.030029 -122.055915
```

[11]:

```python
residuals = price - res_price.fittedvalues
residuals = residuals.dropna()

log_sq_resid = np.log(residuals ** 2)
```

[12]:

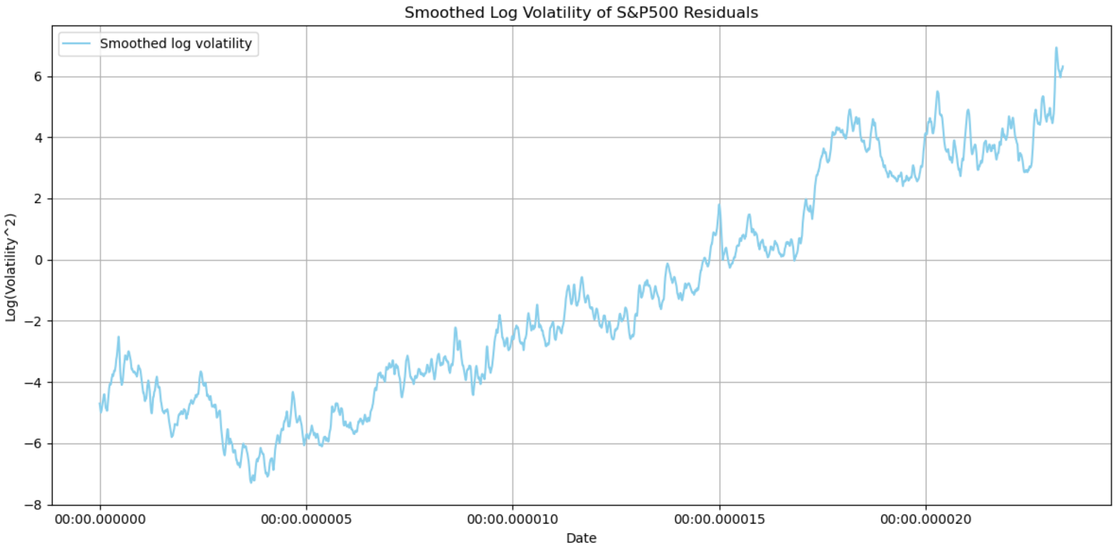
```python
vol_model = sm.tsa.UnobservedComponents(log_sq_resid, level='local level')
vol_fit = vol_model.fit()

vol_smoothed = vol_fit.smoothed_state[0]
```

```
/Users/jiminbyun/anaconda3/lib/python3.11/site-
packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: No frequency
information was provided, so inferred frequency N will be used.
  self._init_dates(dates, freq)
 This problem is unconstrained.
```

```
RUNNING THE L-BFGS-B CODE

           * * *

Machine precision = 2.220D-16
 N =            2     M =           10

At X0         0 variables are exactly at the bounds

At iterate    0    f=  2.66190D+00    |proj g|=  1.17247D-01

At iterate    5    f=  2.29048D+00    |proj g|=  1.86803D-01

At iterate   10    f=  2.28063D+00    |proj g|=  3.64017D-03

           * * *

Tit   = total number of iterations
Tnf   = total number of function evaluations
Tnint = total number of segments explored during Cauchy searches
Skip  = number of BFGS updates skipped
Nact  = number of active bounds at final generalized Cauchy point
Projg = norm of the final projected gradient
F     = final function value
```

```
           * * *

   N    Tit     Tnf  Tnint  Skip  Nact     Projg        F
    2     13     20      1     0     0   1.911D-06   2.281D+00
  F =    2.2806209950968666

CONVERGENCE: NORM_OF_PROJECTED_GRADIENT_<=_PGTOL
```

[13]:
```
plt.figure(figsize=(12, 6))
plt.plot(log_sq_resid.index, vol_smoothed, label='Smoothed log volatility',␣
  ↪color='skyblue')
plt.title('Smoothed Log Volatility of S&P500 Residuals')
plt.xlabel('Date')
plt.ylabel('Log(Volatility^2)')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```



[ ]:
```

```

[ ]:
```
#   (S&P500)          (KOSPI   KOSDAQ  )        (connectedness analysis)
```

[14]:
```
import pandas as pd
import numpy as np
import yfinance as yf
import matplotlib.pyplot as plt
```

```python
sp500 = pd.read_csv("/Users/jiminbyun/Downloads/SPX.csv", parse_dates=['Date'],␣
 ↪index_col='Date')
sp500 = sp500[['Close']].dropna()
sp500['log_ret'] = np.log(sp500['Close'] / sp500['Close'].shift(1))

df = pd.read_csv("/Users/jiminbyun/Downloads/kospi_english.csv")
df_long = df.melt(id_vars='Sector', var_name='Date', value_name='Close')
df_long['Date'] = pd.to_datetime(df_long['Date'], format='%Y.%m',␣
 ↪errors='coerce')
df_long['Close'] = pd.to_numeric(df_long['Close'].replace(',', '', regex=True),␣
 ↪errors='coerce')

df_long = df_long.sort_values(['Sector', 'Date'])
df_long['log_ret'] = df_long.groupby('Sector')['Close'].transform(lambda x: np.
 ↪log(x / x.shift(1)))

kospi = df_long[df_long['Sector'] == 'KOSPI 200'].dropna(subset=['log_ret'])
kospi = kospi.set_index('Date')
```

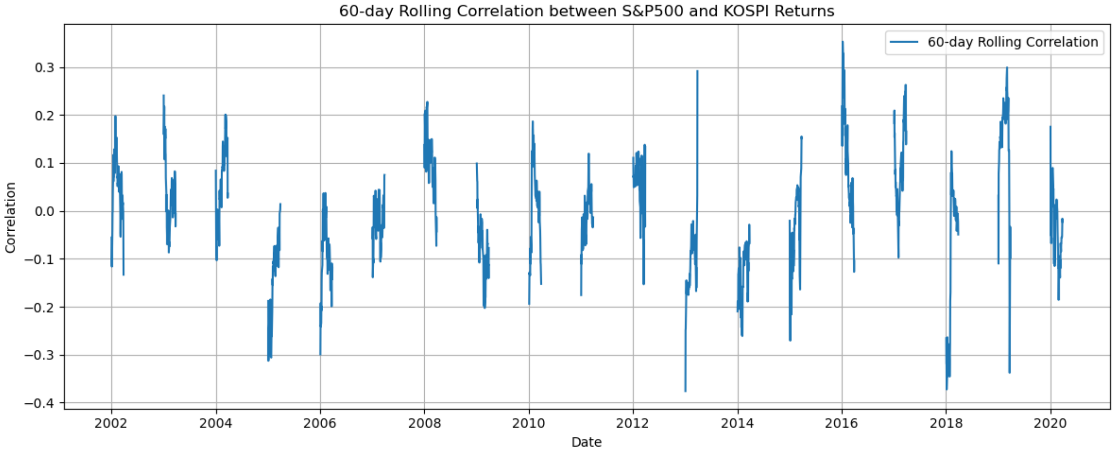
```
/Users/jiminbyun/anaconda3/lib/python3.11/site-
packages/pandas/core/arraylike.py:402: RuntimeWarning: divide by zero
encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```

[15]:

```python
kospi = df_long.groupby('Date')['log_ret'].mean().to_frame(name='log_ret')

kospi = kospi.sort_index()
kospi = kospi.resample('D').ffill()
```

[16]:

```python
merged = pd.merge(
    sp500[['log_ret']],
    kospi[['log_ret']],
    left_index=True, right_index=True,
    how='inner'
)

merged.columns = ['log_ret_sp', 'log_ret_ko']
merged.dropna(inplace=True)

rolling_corr = merged['log_ret_sp'].rolling(60).corr(merged['log_ret_ko'])

plt.figure(figsize=(12, 5))
plt.plot(rolling_corr.index, rolling_corr, label='60-day Rolling Correlation')
plt.title('60-day Rolling Correlation between S&P500 and KOSPI Returns')
plt.xlabel('Date')
plt.ylabel('Correlation')
plt.grid(True)
plt.legend()
plt.tight_layout()
plt.show()
```



[ ]:

```
```

[68]:

```
#Getting expected reasons for residuals(Relative news)
```

[17]:

```
#From GNews

import requests
import pandas as pd
from datetime import timedelta

API_KEY = 'f3106a94916a7f52c360f7a9a6e0bf6d'
BASE_URL = 'https://gnews.io/api/v4/search'

def get_news_on_date(date_str, query="S P 500"):
    date = pd.to_datetime(date_str)
    from_date = date.strftime('%Y-%m-%d')
    to_date = (date + timedelta(days=1)).strftime('%Y-%m-%d')

    params = {
        'q': query,
        'from': from_date,
        'to': to_date,
        'lang': 'en',
        'sortBy': 'relevance',
        'max': 10,
        'token': API_KEY
    }
    response = requests.get(BASE_URL, params=params)
    if response.status_code == 200:
        articles = response.json().get('articles', [])
        print(f"\n {date_str} news:")
        for i, article in enumerate(articles, 1):
            print(f"\n[news {i}]")
            print("Title:", article['title'])
            print("Ref:", article['source']['name'])
            print("URL:", article['url'])
    else:
        print(f"Error {response.status_code} - {response.text}")

dates_to_check = [
    "1987-10-19",
    "2008-09-29",
    "2020-03-16"
]

for date_str in dates_to_check:
    get_news_on_date(date_str)
```

```
 1987-10-19 news:

[news 1]
Title: US stocks drift closer to their record as Wall Street waits to see what
US-China talks will bear
Ref: GuelphToday
URL: https://www.guelphtoday.com/national-business/us-stocks-drift-closer-to-
their-record-as-wall-street-waits-to-see-what-us-china-talks-will-bear-10786172

[news 2]
Title: US stocks drift closer to their record as Wall Street waits to see what
US-China talks will bear
Ref: Vancouver Is Awesome
URL: https://www.vancouverisawesome.com/national-business/us-stocks-drift-
closer-to-their-record-as-wall-street-waits-to-see-what-us-china-talks-will-
bear-10786183

[news 3]
Title: US stocks drift closer to their record as Wall Street waits to see what
US-China talks will bear
Ref: BradfordToday
URL: https://www.bradfordtoday.ca/national-business/us-stocks-drift-closer-to-
their-record-as-wall-street-waits-to-see-what-us-china-talks-will-bear-10786172
```

[news 4]
Title: US stocks drift closer to their record as Wall Street waits to see what US-China talks will bear
Ref: ElliotLakeToday.com
URL: https://www.elliotlaketoday.com/national-business/us-stocks-drift-closer-to-their-record-as-wall-street-waits-to-see-what-us-china-talks-will-bear-10786172

[news 5]
Title: US stocks drift closer to their record as Wall Street waits to see what US-China talks will bear
Ref: BayToday
URL: https://www.baytoday.ca/business/us-stocks-drift-closer-to-their-record-as-wall-street-waits-to-see-what-us-china-talks-will-bear-10786172

[news 6]
Title: US stocks drift closer to their record as Wall Street waits to see what US-China talks will bear
Ref: SooToday
URL: https://www.sootoday.com/national-business/us-stocks-drift-closer-to-their-record-as-wall-street-waits-to-see-what-us-china-talks-will-bear-10786172

[news 7]
Title: ASX poised to top 8600, S&P 500 edges higher
Ref: The Australian Financial Review
URL: https://www.afr.com/markets/equity-markets/asx-poised-to-top-8600-s-and-p-500-edges-higher-20250611-p5m6g3

[news 8]
Title: US stocks remain in limbo as talks continue with China on defusing trade tensions
Ref: BayToday
URL: https://www.baytoday.ca/business/us-stocks-remain-in-limbo-as-talks-continue-with-china-on-defusing-trade-tensions-10786172

[news 9]
Title: US stocks remain in limbo as talks continue with China on defusing trade tensions
Ref: BradfordToday
URL: https://www.bradfordtoday.ca/national-business/us-stocks-remain-in-limbo-as-talks-continue-with-china-on-defusing-trade-tensions-10786172

[news 10]
Title: US stocks remain in limbo as talks continue with China on defusing trade tensions
Ref: SooToday
URL: https://www.sootoday.com/national-business/us-stocks-remain-in-limbo-as-talks-continue-with-china-on-defusing-trade-tensions-10786172

2008-09-29 news:

[news 1]
Title: US stocks drift closer to their record as Wall Street waits to see what US-China talks will bear
Ref: GuelphToday
URL: https://www.guelphtoday.com/national-business/us-stocks-drift-closer-to-their-record-as-wall-street-waits-to-see-what-us-china-talks-will-bear-10786172

[news 2]
Title: US stocks drift closer to their record as Wall Street waits to see what US-China talks will bear
Ref: Vancouver Is Awesome
URL: https://www.vancouverisawesome.com/national-business/us-stocks-drift-closer-to-their-record-as-wall-street-waits-to-see-what-us-china-talks-will-bear-10786183

[news 3]
Title: US stocks drift closer to their record as Wall Street waits to see what US-China talks will bear
Ref: BradfordToday
URL: https://www.bradfordtoday.ca/national-business/us-stocks-drift-closer-to-their-record-as-wall-street-waits-to-see-what-us-china-talks-will-bear-10786172

[news 4]
Title: US stocks drift closer to their record as Wall Street waits to see what US-China talks will bear
Ref: ElliotLakeToday.com
URL: https://www.elliotlaketoday.com/national-business/us-stocks-drift-closer-to-their-record-as-wall-street-waits-to-see-what-us-china-talks-will-bear-10786172

[news 5]
Title: US stocks drift closer to their record as Wall Street waits to see what US-China talks will bear
Ref: BayToday
URL: https://www.baytoday.ca/business/us-stocks-drift-closer-to-their-record-as-wall-street-waits-to-see-what-us-china-talks-will-bear-10786172

[news 6]
Title: US stocks drift closer to their record as Wall Street waits to see what US-China talks will bear
Ref: SooToday
URL: https://www.sootoday.com/national-business/us-stocks-drift-closer-to-their-record-as-wall-street-waits-to-see-what-us-china-talks-will-bear-10786172

[news 7]

Title: ASX poised to top 8600, S&P 500 edges higher
Ref: The Australian Financial Review
URL: https://www.afr.com/markets/equity-markets/asx-poised-to-top-8600-s-and-p-500-edges-higher-20250611-p5m6g3

[news 8]
Title: US stocks remain in limbo as talks continue with China on defusing trade tensions
Ref: BayToday
URL: https://www.baytoday.ca/business/us-stocks-remain-in-limbo-as-talks-continue-with-china-on-defusing-trade-tensions-10786172

[news 9]
Title: US stocks remain in limbo as talks continue with China on defusing trade tensions
Ref: BradfordToday
URL: https://www.bradfordtoday.ca/national-business/us-stocks-remain-in-limbo-as-talks-continue-with-china-on-defusing-trade-tensions-10786172

[news 10]
Title: US stocks remain in limbo as talks continue with China on defusing trade tensions
Ref: SooToday
URL: https://www.sootoday.com/national-business/us-stocks-remain-in-limbo-as-talks-continue-with-china-on-defusing-trade-tensions-10786172

2020-03-16 news:

[news 1]
Title: US stocks drift closer to their record as Wall Street waits to see what US-China talks will bear
Ref: GuelphToday
URL: https://www.guelphtoday.com/national-business/us-stocks-drift-closer-to-their-record-as-wall-street-waits-to-see-what-us-china-talks-will-bear-10786172

[news 2]
Title: US stocks drift closer to their record as Wall Street waits to see what US-China talks will bear
Ref: Vancouver Is Awesome
URL: https://www.vancouverisawesome.com/national-business/us-stocks-drift-closer-to-their-record-as-wall-street-waits-to-see-what-us-china-talks-will-bear-10786183

[news 3]
Title: US stocks drift closer to their record as Wall Street waits to see what US-China talks will bear
Ref: BradfordToday
URL: https://www.bradfordtoday.ca/national-business/us-stocks-drift-closer-to-

their-record-as-wall-street-waits-to-see-what-us-china-talks-will-bear-10786172

[news 4]
Title: US stocks drift closer to their record as Wall Street waits to see what US-China talks will bear
Ref: ElliotLakeToday.com
URL: https://www.elliotlaketoday.com/national-business/us-stocks-drift-closer-to-their-record-as-wall-street-waits-to-see-what-us-china-talks-will-bear-10786172

[news 5]
Title: US stocks drift closer to their record as Wall Street waits to see what US-China talks will bear
Ref: BayToday
URL: https://www.baytoday.ca/business/us-stocks-drift-closer-to-their-record-as-wall-street-waits-to-see-what-us-china-talks-will-bear-10786172

[news 6]
Title: US stocks drift closer to their record as Wall Street waits to see what US-China talks will bear
Ref: SooToday
URL: https://www.sootoday.com/national-business/us-stocks-drift-closer-to-their-record-as-wall-street-waits-to-see-what-us-china-talks-will-bear-10786172

[news 7]
Title: ASX poised to top 8600, S&P 500 edges higher
Ref: The Australian Financial Review
URL: https://www.afr.com/markets/equity-markets/asx-poised-to-top-8600-s-and-p-500-edges-higher-20250611-p5m6g3

[news 8]
Title: US stocks remain in limbo as talks continue with China on defusing trade tensions
Ref: BayToday
URL: https://www.baytoday.ca/business/us-stocks-remain-in-limbo-as-talks-continue-with-china-on-defusing-trade-tensions-10786172

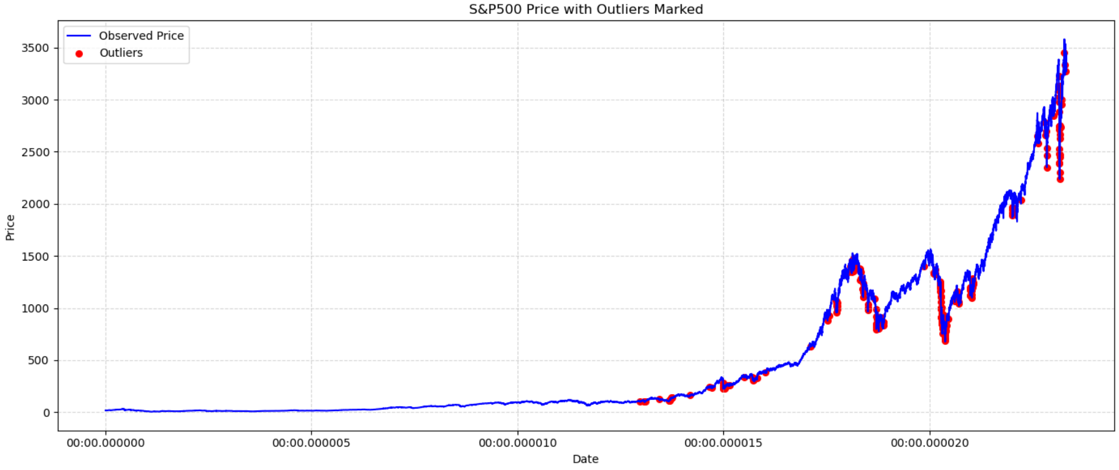
[news 9]
Title: US stocks remain in limbo as talks continue with China on defusing trade tensions
Ref: BradfordToday
URL: https://www.bradfordtoday.ca/national-business/us-stocks-remain-in-limbo-as-talks-continue-with-china-on-defusing-trade-tensions-10786172

[news 10]
Title: US stocks remain in limbo as talks continue with China on defusing trade tensions
Ref: SooToday

URL: https://www.sootoday.com/national-business/us-stocks-remain-in-limbo-as-talks-continue-with-china-on-defusing-trade-tensions-10786172

[18]:

```
plt.figure(figsize=(14, 6))
plt.plot(price.index, price, label='Observed Price', color='blue')
plt.scatter(outliers.index, price.loc[outliers.index], color='red', s=30,␣
 ↪label='Outliers')
plt.title("S&P500 Price with Outliers Marked")
plt.xlabel("Date")
plt.ylabel("Price")
plt.legend()
plt.grid(True, linestyle='--', alpha=0.5)
plt.tight_layout()
plt.show()
```



[ ]:

```

```

[ ]:

```
#IRF Graph
```

[19]:

```
import pandas as pd
import numpy as np

data = pd.read_csv("/Users/jiminbyun/Downloads/SPX.csv", parse_dates=['Date'])
data = data.set_index('Date')
data = data.sort_index()

data['log_ret_sp'] = np.log(data['Close'] / data['Close'].shift(1))

data['log_vol_sp'] = np.log(data['log_ret_sp']**2 + 1e-8)

df = pd.read_csv("/Users/jiminbyun/Downloads/kospi_english.csv")
```

```
df_long = df.melt(id_vars='Sector', var_name='Date', value_name='Close')
df_long['Date'] = pd.to_datetime(df_long['Date'], format='%Y.%m',␣
 ↪errors='coerce')
df_long['Close'] = pd.to_numeric(df_long['Close'].replace(',', '', regex=True),␣
 ↪errors='coerce')

df_long = df_long.sort_values(['Sector', 'Date'])
df_long['log_ret'] = df_long.groupby('Sector')['Close'].transform(lambda x: np.
 ↪log(x / x.shift(1)))

kospi = df_long[df_long['Sector'] == 'KOSPI 200'].dropna(subset=['log_ret'])
kospi = kospi.set_index('Date')
kospi['log_ret_ko'] = np.log(kospi['Close'] / kospi['Close'].shift(1))
kospi['log_vol_ko'] = np.log(kospi['log_ret_ko']**2 + 1e-8)

merged = pd.merge(data[['log_vol_sp']], kospi[['log_vol_ko']], left_index=True,␣
 ↪right_index=True)
vol_df = merged.dropna()
```

```
/Users/jiminbyun/anaconda3/lib/python3.11/site-
packages/pandas/core/arraylike.py:402: RuntimeWarning: divide by zero
encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```

[20]:

```
df = pd.read_csv("/Users/jiminbyun/Downloads/kospi_english.csv")

df_long = df.melt(id_vars='Sector', var_name='Date', value_name='Close')

df_long['Date'] = pd.to_datetime(df_long['Date'], errors='coerce')

df_long['Close'] = pd.to_numeric(df_long['Close'].replace(',', '', regex=True),␣
 ↪errors='coerce')

df_long = df_long.sort_values(['Sector', 'Date'])
df_long['log_ret'] = df_long.groupby('Sector')['Close'].transform(lambda x: np.
 ↪log(x / x.shift(1)))

kospi = df_long[df_long['Sector'] == 'Finance'].copy()
kospi = kospi.set_index('Date')
kospi['log_ret_ko'] = np.log(kospi['Close'] / kospi['Close'].shift(1))
kospi['log_vol_ko'] = np.log(kospi['log_ret_ko']**2 + 1e-8)
```

```
/Users/jiminbyun/anaconda3/lib/python3.11/site-
packages/pandas/core/arraylike.py:402: RuntimeWarning: divide by zero
encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```

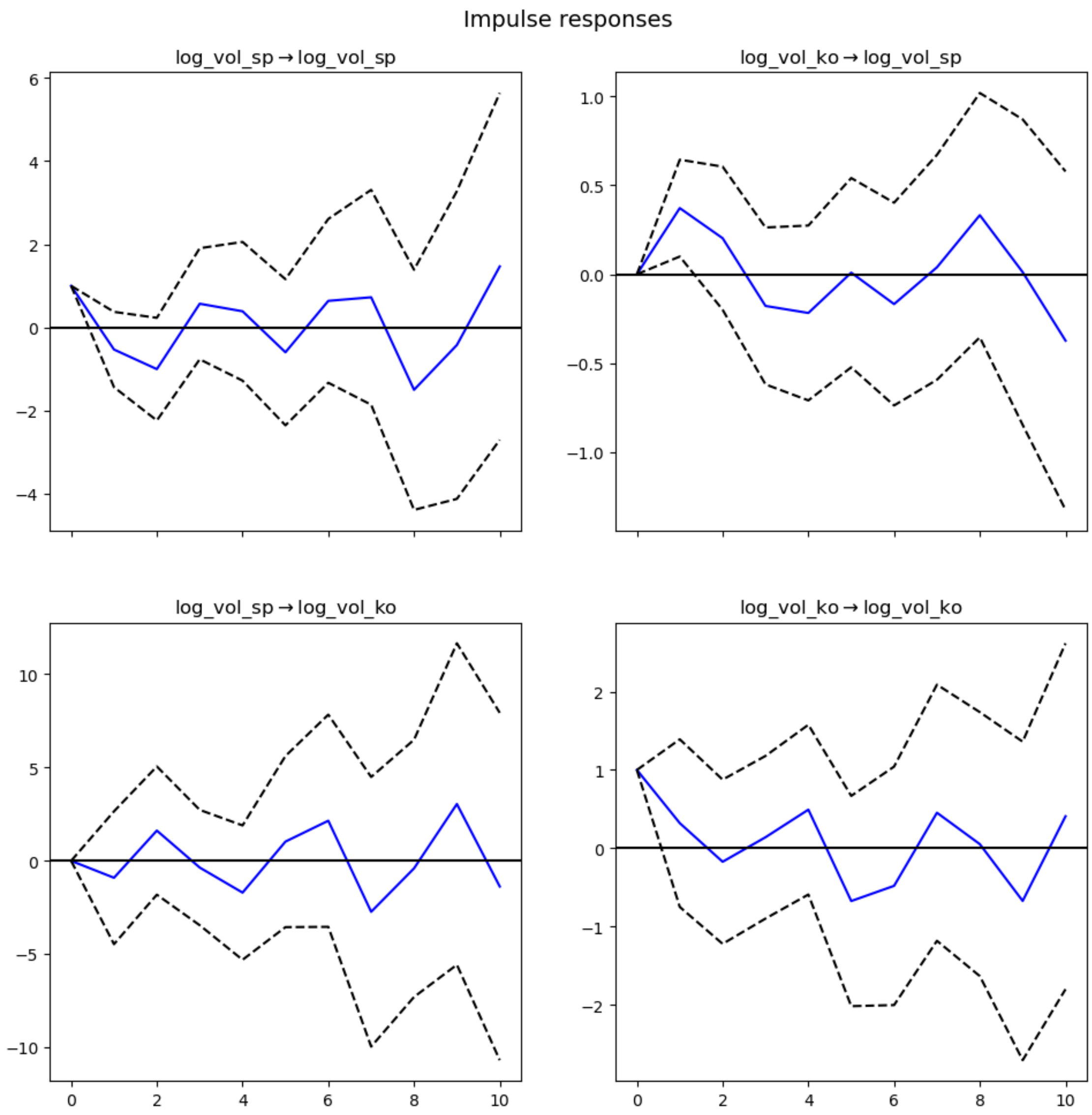
[21]:

```
data['Month'] = data.index.to_period('M')
monthly_sp = data.groupby('Month')['log_vol_sp'].mean().dropna()
monthly_sp.index = monthly_sp.index.to_timestamp()

merged = pd.merge(monthly_sp, kospi['log_vol_ko'], left_index=True,␣
  ↪right_index=True)
vol_df = merged.dropna()
```

[22]:

```
from statsmodels.tsa.api import VAR
import matplotlib.pyplot as plt

model = VAR(vol_df)
results = model.fit(maxlags=5, ic='aic')
irf = results.irf(10)
irf.plot(orth=False)
plt.show()
```

```
/Users/jiminbyun/anaconda3/lib/python3.11/site-
packages/statsmodels/tsa/base/tsa_model.py:473: ValueWarning: No frequency
information was provided, so inferred frequency AS-JAN will be used.
  self._init_dates(dates, freq)
```



[23]:
```
rows = []

variable_names = [col[0] if isinstance(col, tuple) else col for col in vol_df.
 ↪columns.tolist()]

for i in range(irf.irfs.shape[1]):
    for j in range(irf.irfs.shape[2]):
        ir_series = irf.irfs[:, i, j]
        max_effect = np.max(np.abs(ir_series))
        max_idx = np.argmax(np.abs(ir_series))

        rows.append({
            "Impulse": variable_names[i],
            "Response": variable_names[j],
            "Max Effect": round(ir_series[max_idx], 4),
            "Days to Max": max_idx,
            "Significant": "- (Not Tested)"
        })

irf_summary = pd.DataFrame(rows)
print(irf_summary)
```

```
      Impulse    Response  Max Effect  Days to Max     Significant
0  log_vol_sp  log_vol_sp     -1.5016            8  - (Not Tested)
1  log_vol_sp  log_vol_ko     -0.3735           10  - (Not Tested)
2  log_vol_ko  log_vol_sp      3.0434            9  - (Not Tested)
3  log_vol_ko  log_vol_ko      1.0000            0  - (Not Tested)
```

[ ]:

```

```