Stationary Test_경기방어주

February 23, 2021

[35]:

```
from dateutil.parser import parse
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import pandas as pd
```

[36]:

```
# FinanceDataReader로 데이터를 불러옵니다
# 예측할 종목은 한양증권(001750) 입니다

import FinanceDataReader as fdr

LG_Household_Health_Care_Ltd = '051900'
Amorepacific_Corp = '090430'
Bukwang_Pharmaceutical_Co_Ltd = '003000'
Cj_Logistics_Corporation = '000120'
CS_Wind_Corporation = '112610'
E_MART = '139480'
Green_Cross_Holdings_Corporation = '005250'
Hanmi_Pharm_Co_Ltd = '128940'
Hanmi_Semiconductor_Co_Ltd = '042700'
Hansol_Chemical_Co_Ltd = '014680'
HMM_Co_Ltd = '011200'
Hyundai_Heavy_Industries_Holdings_Co_Ltd = '267250'
IS_DongSeo_Co_Ltd = '010780'
Kakao_Corp = '035720'
Kiwoom_Securities_Co_Ltd = '039490'
Korea_Petrochemical_Ind_Co_Ltd = '006650'
Lotte_Chilsung_Beverage_Co_Ltd = '005300'
Shinsegae_Inc = '004170'
SKC_Co_Ltd = '011790'
Solus_Advanced_Materials_Co_Ltd = '336370'
Youngone_Corporation = '111770'
Yuhan_Corporation = '000100'
```

[37]:

```
df_LG = fdr.DataReader(LG_Household_Health_Care_Ltd , '2020-02-22', '2021-02-22')
df_Amore = fdr.DataReader(Amorepacific_Corp , '2020-02-22', '2021-02-22')
```

```
df_Bukwang = fdr.DataReader(Bukwang_Pharmaceutical_Co_Ltd , '2020-02-22',␣
 ↪'2021-02-22')
df_Cj = fdr.DataReader(Cj_Logistics_Corporation , '2020-02-22', '2021-02-22')
df_CS = fdr.DataReader(CS_Wind_Corporation , '2020-02-22', '2021-02-22')
df_E_MART = fdr.DataReader(E_MART , '2020-02-22', '2021-02-22')
df_Green_Cross = fdr.DataReader(Green_Cross_Holdings_Corporation , '2020-02-22',␣
 ↪'2021-02-22')
df_Hanmi_Pharm = fdr.DataReader(Hanmi_Pharm_Co_Ltd , '2020-02-22', '2021-02-22')
df_Hanmi_Semiconductor = fdr.DataReader(Hanmi_Semiconductor_Co_Ltd ,␣
 ↪'2020-02-22', '2021-02-22')
df_Hansol_Chemical = fdr.DataReader(Hansol_Chemical_Co_Ltd , '2020-02-22',␣
 ↪'2021-02-22')
df_HMM = fdr.DataReader(HMM_Co_Ltd , '2020-02-22', '2021-02-22')
df_Hyundai_Heavy_Industries = fdr.
 ↪DataReader(Hyundai_Heavy_Industries_Holdings_Co_Ltd , '2020-02-22',␣
 ↪'2021-02-22')
df_IS_DongSeo = fdr.DataReader(IS_DongSeo_Co_Ltd , '2020-02-22', '2021-02-22')
df_Kakao = fdr.DataReader(Kakao_Corp , '2020-02-22', '2021-02-22')
df_Kiwoom_Securities = fdr.DataReader(Kiwoom_Securities_Co_Ltd , '2020-02-22',␣
 ↪'2021-02-22')
df_Korea_Petrochemical = fdr.DataReader(Korea_Petrochemical_Ind_Co_Ltd ,␣
 ↪'2020-02-22', '2021-02-22')
df_Lotte_Chilsung_Beverage = fdr.DataReader(Lotte_Chilsung_Beverage_Co_Ltd ,␣
 ↪'2020-02-22', '2021-02-22')
df_Shinsegae = fdr.DataReader(Shinsegae_Inc , '2020-02-22', '2021-02-22')
df_SKC = fdr.DataReader(SKC_Co_Ltd  , '2020-02-22', '2021-02-22')
df_Solus_Advanced_Materials = fdr.DataReader(Solus_Advanced_Materials_Co_Ltd ,␣
 ↪'2020-02-22', '2021-02-22')
df_Youngone = fdr.DataReader(Youngone_Corporation  , '2020-02-22', '2021-02-22')
df_Yuhan = fdr.DataReader(Yuhan_Corporation  , '2020-02-22', '2021-02-22')
```

[38]:

```
df = pd.merge(df_LG['Close'], df_Amore['Close'], left_index=True,␣
 ↪right_index=True, how='left').rename(columns ={'Close_x':'LG', 'Close_y':
 ↪'Amore'})
df = pd.merge(df, df_Bukwang['Close'], left_index = True, right_index=True,␣
 ↪how='left').rename(columns = {'Close':'Bukwang'})
df = pd.merge(df, df_Cj['Close'], left_index = True, right_index=True,␣
 ↪how='left').rename(columns = {'Close':'Cj'})
df = pd.merge(df, df_CS['Close'], left_index = True, right_index=True,␣
 ↪how='left').rename(columns = {'Close':'CS'})
df = pd.merge(df, df_E_MART['Close'], left_index = True, right_index=True,␣
 ↪how='left').rename(columns = {'Close':'E_MART'})
df = pd.merge(df, df_Green_Cross['Close'], left_index = True, right_index=True,␣
 ↪how='left').rename(columns = {'Close':'Green_Cross'})
df = pd.merge(df, df_Hanmi_Pharm['Close'], left_index = True, right_index=True,␣
 ↪how='left').rename(columns = {'Close':'Hanmi_Pharm'})
```

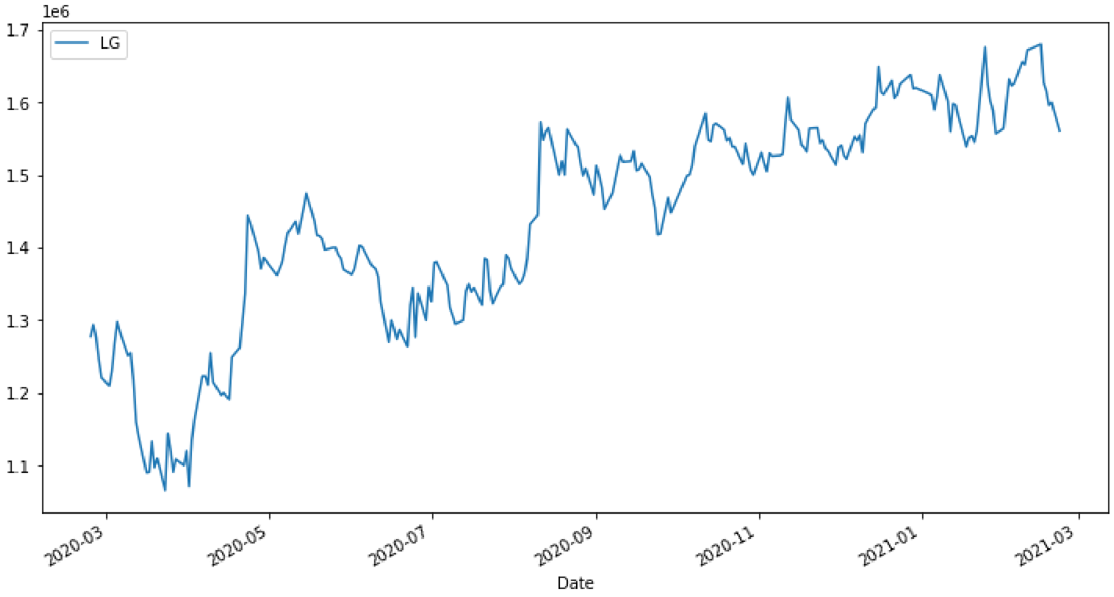
```
df = pd.merge(df, df_Hanmi_Semiconductor['Close'], left_index = True,␣
 ↪right_index=True, how='left').rename(columns = {'Close':'Hanmi_Semiconductor'})
df = pd.merge(df, df_Hansol_Chemical['Close'], left_index = True,␣
 ↪right_index=True, how='left').rename(columns = {'Close':'Hansol_Chemical'})
df = pd.merge(df, df_HMM['Close'], left_index = True, right_index=True,␣
 ↪how='left').rename(columns = {'Close':'HMM'})
df = pd.merge(df, df_Hyundai_Heavy_Industries['Close'], left_index = True,␣
 ↪right_index=True, how='left').rename(columns = {'Close':
 ↪'Hyundai_Heavy_Industries'})
df = pd.merge(df, df_IS_DongSeo['Close'], left_index = True, right_index=True,␣
 ↪how='left').rename(columns = {'Close':'IS_DongSeo'})
df = pd.merge(df, df_Kakao['Close'], left_index = True, right_index=True,␣
 ↪how='left').rename(columns = {'Close':'Kakao'})
df = pd.merge(df, df_Kiwoom_Securities['Close'], left_index = True,␣
 ↪right_index=True, how='left').rename(columns = {'Close':'Kiwoom_Securities'})
df = pd.merge(df, df_Korea_Petrochemical['Close'], left_index = True,␣
 ↪right_index=True, how='left').rename(columns = {'Close':'Korea_Petrochemical'})
df = pd.merge(df, df_Lotte_Chilsung_Beverage['Close'], left_index = True,␣
 ↪right_index=True, how='left').rename(columns = {'Close':
 ↪'Lotte_Chilsung_Beverage'})
df = pd.merge(df, df_Shinsegae['Close'], left_index = True, right_index=True,␣
 ↪how='left').rename(columns = {'Close':'Shinsegae'})
df = pd.merge(df, df_SKC['Close'], left_index = True, right_index=True,␣
 ↪how='left').rename(columns = {'Close':'SKC'})
df = pd.merge(df, df_Solus_Advanced_Materials['Close'], left_index = True,␣
 ↪right_index=True, how='left').rename(columns = {'Close':
 ↪'Solus_Advanced_Materials'})
df = pd.merge(df, df_Youngone['Close'], left_index = True, right_index=True,␣
 ↪how='left').rename(columns = {'Close':'Youngone'})
df = pd.merge(df, df_Yuhan['Close'], left_index = True, right_index=True,␣
 ↪how='left').rename(columns = {'Close':'Yuhan'})


df.dropna(inplace = True)

df.head()
```

[38]:

```
                 LG   Amore  Bukwang      Cj     CS  E_MART  Green_Cross  \
Date
2020-02-24  1278000  173000    11697  134500  17343   99300        17700
2020-02-25  1294000  175000    11971  136500  17627  103500        17950
2020-02-26  1277000  171500    12108  137000  17651  106500        17950
2020-02-27  1246000  168500    11971  137500  17129  107500        18100
2020-02-28  1221000  161000    11468  133500  16868  105000        17550

            Hanmi_Pharm  Hanmi_Semiconductor  Hansol_Chemical  ...  \
Date                                                           ...
2020-02-24       256255                 9190            99100  ...
2020-02-25       261164                 9670            96200  ...
2020-02-26       260673                 9680            96900  ...
2020-02-27       265091                 9110            96300  ...
2020-02-28       256255                 8660            92800  ...

            IS_DongSeo   Kakao  Kiwoom_Securities  Korea_Petrochemical  \
Date
2020-02-24       26700  183500              69600                90900
2020-02-25       27900  185500              70300                91700
2020-02-26       27600  184000              69400                92000
2020-02-27       28100  178500              68300                90700
2020-02-28       26800  172000              66100                85600

            Lotte_Chilsung_Beverage  Shinsegae    SKC  \
Date
2020-02-24                   116000     244500  54400
2020-02-25                   116500     254500  56400
2020-02-26                   116500     254000  55300
2020-02-27                   114500     249000  53700
2020-02-28                   112500     236000  51400

            Solus_Advanced_Materials  Youngone  Yuhan
Date
2020-02-24                     26050     28500  39616
2020-02-25                     27250     29600  40573
2020-02-26                     27200     30750  40478
2020-02-27                     26000     31600  41435
2020-02-28                     24750     31500  40956

[5 rows x 22 columns]
```

[39]:
```
columns = ['LG']
df[columns].plot(figsize=(12.2,6.4))
```

[39]:
```
<AxesSubplot:xlabel='Date'>
```



# ADF Test > 귀무(영) 가설(null hypothesis)은 시계열 데이터가 단위 근을 포함하고 Non-stationary 다. 따라서 ADF Test에서 P값이 유의값 0.05 보다 작으면 귀무가설은 기각된다 -> Stationary time-series

[40]:

```
n_obs = 20
df_train, df_test = df[0:-n_obs], df[-n_obs:]

from statsmodels.tsa.stattools import adfuller

# ADF Test는 시계열이 안정적(Stationary)인지 여부를 확인하는데 이용됩니다.
# p-value>0.05이면 귀무 가설을 기각할 수 없습니다. -> 안정x
# 모든 주식이 귀무 가설을 기각할 수 없으므로 안정적이지 않습니다.

def adf_test(df):
    result = adfuller(df.values)
    print('ADF Statistics: %f' % result[0])
    print('p-value: %f' % result[1])
    print('Critical values:')
    for key, value in result[4].items():
        print('\t%s: %.3f' % (key, value))

print('ADF Test: LG_Household_Health_Care_Ltd Time series')
adf_test(df_train['LG'])

print('\n\nADF Test: Amorepacific_Corp')
adf_test(df_train['Amore'])
```

```python
print('\n\nADF Test: Bukwang_Pharmaceutical_Co_Ltd Time series')
adf_test(df_train['Bukwang'])

print('\n\nADF Test: Cj_Logistics_Corporation Time series')
adf_test(df_train['Cj'])

print('\n\nADF Test: CS_Wind_Corporation Time series')
adf_test(df_train['CS'])

print('\n\nADF Test: E_MART Time series')
adf_test(df_train['E_MART'])

print('\n\nADF Test: Green_Cross_Holdings_Corporation Time series')
adf_test(df_train['Green_Cross'])

print('\n\nADF Test: Hanmi_Pharm_Co_Ltd Time series')
adf_test(df_train['Hanmi_Pharm'])

print('\n\nADF Test: Hanmi_Semiconductor_Co_Ltd Time series')
adf_test(df_train['Hanmi_Semiconductor'])

print('\n\nADF Test: Hansol_Chemical_Co_Ltd Time series')
adf_test(df_train['Hansol_Chemical'])

print('\n\nADF Test: HMM Time series')
adf_test(df_train['HMM'])

print('\n\nADF Test: Hyundai_Heavy_Industries Time series')
adf_test(df_train['Hyundai_Heavy_Industries'])

print('\n\nADF Test: IS_DongSeo Time series')
adf_test(df_train['IS_DongSeo'])

print('\n\nADF Test: Kakao Time series')
adf_test(df_train['Kakao'])

print('\n\nADF Test: Kiwoom_Securities Time series')
adf_test(df_train['Kiwoom_Securities'])

print('\n\nADF Test: Korea_Petrochemical Time series')
adf_test(df_train['Korea_Petrochemical'])

print('\n\nADF Test: Lotte_Chilsung_Beverage Time series')
adf_test(df_train['Lotte_Chilsung_Beverage'])

print('\n\nADF Test: Shinsegae Time series')
adf_test(df_train['Shinsegae'])
```

```
print('\n\nADF Test: SKC Time series')
adf_test(df_train['SKC'])

print('\n\nADF Test: Solus_Advanced_Materials Time series')
adf_test(df_train['Solus_Advanced_Materials'])

print('\n\nADF Test: Youngone Time series')
adf_test(df_train['Youngone'])

print('\n\nADF Test: Yuhan Time series')
adf_test(df_train['Yuhan'])
```

```
ADF Test: LG_Household_Health_Care_Ltd Time series
ADF Statistics: -1.374178
p-value: 0.594591
Critical values:
        1%: -3.460
        5%: -2.874
        10%: -2.574


ADF Test: Amorepacific_Corp
ADF Statistics: -0.213532
p-value: 0.936889
Critical values:
        1%: -3.460
        5%: -2.874
        10%: -2.574


ADF Test: Bukwang_Pharmaceutical_Co_Ltd Time series
ADF Statistics: -2.444841
p-value: 0.129478
Critical values:
        1%: -3.460
        5%: -2.874
        10%: -2.574


ADF Test: Cj_Logistics_Corporation Time series
ADF Statistics: -2.082783
p-value: 0.251494
Critical values:
        1%: -3.460
        5%: -2.874
        10%: -2.574
```

```
ADF Test: CS_Wind_Corporation Time series
ADF Statistics: 0.611481
p-value: 0.987895
Critical values:
        1%: -3.460
        5%: -2.875
        10%: -2.574


ADF Test: E_MART Time series
ADF Statistics: -0.365986
p-value: 0.915625
Critical values:
        1%: -3.460
        5%: -2.875
        10%: -2.574


ADF Test: Green_Cross_Holdings_Corporation Time series
ADF Statistics: -1.000872
p-value: 0.752946
Critical values:
        1%: -3.461
        5%: -2.875
        10%: -2.574


ADF Test: Hanmi_Pharm_Co_Ltd Time series
ADF Statistics: -1.239902
p-value: 0.656200
Critical values:
        1%: -3.460
        5%: -2.874
        10%: -2.574


ADF Test: Hanmi_Semiconductor_Co_Ltd Time series
ADF Statistics: 1.801061
p-value: 0.998351
Critical values:
        1%: -3.460
        5%: -2.874
        10%: -2.574


ADF Test: Hansol_Chemical_Co_Ltd Time series
ADF Statistics: -0.608873
```

```
p-value: 0.868964
Critical values:
        1%: -3.461
        5%: -2.875
        10%: -2.574


ADF Test: HMM Time series
ADF Statistics: 0.161638
p-value: 0.970026
Critical values:
        1%: -3.461
        5%: -2.875
        10%: -2.574


ADF Test: Hyundai_Heavy_Industries Time series
ADF Statistics: -1.879521
p-value: 0.341755
Critical values:
        1%: -3.460
        5%: -2.874
        10%: -2.574


ADF Test: IS_DongSeo Time series
ADF Statistics: -0.363855
p-value: 0.915961
Critical values:
        1%: -3.460
        5%: -2.874
        10%: -2.574


ADF Test: Kakao Time series
ADF Statistics: -0.420772
p-value: 0.906572
Critical values:
        1%: -3.460
        5%: -2.874
        10%: -2.574


ADF Test: Kiwoom_Securities Time series
ADF Statistics: -0.212296
p-value: 0.937040
Critical values:
        1%: -3.460
        5%: -2.874
        10%: -2.574


ADF Test: Korea_Petrochemical Time series
ADF Statistics: 0.242654
p-value: 0.974555
Critical values:
        1%: -3.460
        5%: -2.874
        10%: -2.574


ADF Test: Lotte_Chilsung_Beverage Time series
ADF Statistics: -1.409847
p-value: 0.577546
Critical values:
        1%: -3.460
        5%: -2.874
        10%: -2.574


ADF Test: Shinsegae Time series
ADF Statistics: -2.731201
p-value: 0.068793
Critical values:
        1%: -3.460
        5%: -2.874
        10%: -2.574


ADF Test: SKC Time series
ADF Statistics: 0.486309
p-value: 0.984451
Critical values:
        1%: -3.460
        5%: -2.874
        10%: -2.574


ADF Test: Solus_Advanced_Materials Time series
ADF Statistics: -1.104715
p-value: 0.713188
Critical values:
        1%: -3.460
        5%: -2.874
        10%: -2.574
```

```
ADF Test: Youngone Time series
ADF Statistics: -1.735379
p-value: 0.412983
Critical values:
        1%: -3.460
        5%: -2.874
        10%: -2.574


ADF Test: Yuhan Time series
ADF Statistics: -0.850364
p-value: 0.803879
Critical values:
        1%: -3.460
        5%: -2.875
        10%: -2.574
```

# KPSS Test > Trend Stationary를 Test하는데 사용된다. 귀무 가설과 P값 해석은 ADH Test와 반대이다. 따라서 KPSS Test에서 P값이 유의값 0.05 보다 크면 Trend Stationary이다. 귀무가설은 기각된다 -> Stationary time-series

[41]:

```python
from statsmodels.tsa.stattools import kpss

# p-value<0.05이므로 귀무 가설을 기각할 수 없습니다. -> 안정x

def kpss_test(df):
    statistic, p_value, n_lags, critical_values = kpss(df.values)

    print(f'KPSS Statistic: {statistic}')
    print(f'p-value: {p_value}')
    print(f'num lags: {n_lags}')
    print('Critial Values:')
    for key, value in critical_values.items():
        print(f'{key} : {value}')

print('KPSS Test: LG_Household_Health_Care_Ltd Time series Time series')
kpss_test(df_train['LG'])

print('\n\nKPSS Test: Amorepacific_Corp Time series')
kpss_test(df_train['Amore'])

print('\n\nKPSS Test: Bukwang_Pharmaceutical_Co_Ltd Time series')
kpss_test(df_train['Bukwang'])

print('\n\nKPSS Test: Cj_Logistics_Corporation Time series')
kpss_test(df_train['Cj'])
```

```python
print('\n\nKPSS Test: CS_Wind_Corporation Time series')
kpss_test(df_train['CS'])

print('\n\nKPSS Test: E_MART Time series')
kpss_test(df_train['E_MART'])

print('\n\nKPSS Test: Green_Cross_Holdings_Corporation Time series')
kpss_test(df_train['Green_Cross'])

print('\n\nKPSS Test: Hanmi_Pharm_Co_Ltd Time series')
kpss_test(df_train['Hanmi_Pharm'])

print('\n\nKPSS Test: Hanmi_Semiconductor_Co_Ltd Time series')
kpss_test(df_train['Hanmi_Semiconductor'])

print('\n\nKPSS Test: Hansol_Chemical_Co_Ltd Time series')
kpss_test(df_train['Hansol_Chemical'])

print('\n\nKPSS Test: HMM Time series')
kpss_test(df_train['HMM'])

print('\n\nKPSS Test: Hyundai_Heavy_Industries Time series')
kpss_test(df_train['Hyundai_Heavy_Industries'])

print('\n\nKPSS Test: IS_DongSeo Time series')
kpss_test(df_train['IS_DongSeo'])

print('\n\nKPSS Test: Kakao Time series')
kpss_test(df_train['Kakao'])

print('\n\nKPSS Test: Kiwoom_Securities Time series')
kpss_test(df_train['Kiwoom_Securities'])

print('\n\nKPSS Test: Korea_Petrochemical Time series')
kpss_test(df_train['Korea_Petrochemical'])

print('\n\nKPSS Test: Lotte_Chilsung_Beverage Time series')
kpss_test(df_train['Lotte_Chilsung_Beverage'])

print('\n\nKPSS Test: Shinsegae Time series')
kpss_test(df_train['Shinsegae'])

print('\n\nKPSS Test: SKC Time series')
kpss_test(df_train['SKC'])

print('\n\nKPSS Test: Solus_Advanced_Materials Time series')
kpss_test(df_train['Solus_Advanced_Materials'])
```

```
print('\n\nKPSS Test: Youngone Time series')
kpss_test(df_train['Youngone'])

print('\n\nKPSS Test: Yuhan Time series')
kpss_test(df_train['Yuhan'])
```

```
KPSS Test: LG_Household_Health_Care_Ltd Time series Time series
KPSS Statistic: 1.253314255735552
p-value: 0.01
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: Amorepacific_Corp Time series
KPSS Statistic: 0.7609564456214484
p-value: 0.01
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: Bukwang_Pharmaceutical_Co_Ltd Time series
KPSS Statistic: 0.398425275093653
p-value: 0.07783393314928751
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: Cj_Logistics_Corporation Time series
KPSS Statistic: 0.9808360540961695
p-value: 0.01
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
```

```
1% : 0.739


KPSS Test: CS_Wind_Corporation Time series
KPSS Statistic: 1.4219858614411436
p-value: 0.01
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: E_MART Time series
KPSS Statistic: 1.3846683414307204
p-value: 0.01
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: Green_Cross_Holdings_Corporation Time series
KPSS Statistic: 1.1560036889248393
p-value: 0.01
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: Hanmi_Pharm_Co_Ltd Time series
KPSS Statistic: 1.0708150673849726
p-value: 0.01
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: Hanmi_Semiconductor_Co_Ltd Time series
KPSS Statistic: 1.170025925741159
p-value: 0.01
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: Hansol_Chemical_Co_Ltd Time series
KPSS Statistic: 1.32437104659351
p-value: 0.01
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: HMM Time series
KPSS Statistic: 1.3369667869788198
p-value: 0.01
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: Hyundai_Heavy_Industries Time series
KPSS Statistic: 0.3201939889988351
p-value: 0.1
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: IS_DongSeo Time series
KPSS Statistic: 1.3028755248117037
p-value: 0.01
num lags: 15
Critial Values:
```

```
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: Kakao Time series
KPSS Statistic: 1.3383234901501886
p-value: 0.01
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: Kiwoom_Securities Time series
KPSS Statistic: 1.3867705018494965
p-value: 0.01
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: Korea_Petrochemical Time series
KPSS Statistic: 1.3153878802730636
p-value: 0.01
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: Lotte_Chilsung_Beverage Time series
KPSS Statistic: 0.16775283807438976
p-value: 0.1
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739
```

```
KPSS Test: Shinsegae Time series
KPSS Statistic: 0.33090680185773796
p-value: 0.1
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: SKC Time series
KPSS Statistic: 1.3197197032843662
p-value: 0.01
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: Solus_Advanced_Materials Time series
KPSS Statistic: 1.0221080259033382
p-value: 0.01
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: Youngone Time series
KPSS Statistic: 0.7445536675572113
p-value: 0.01
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: Yuhan Time series
KPSS Statistic: 1.3633014954639013
```

```
p-value: 0.01
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739
```

```
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1850: FutureWarning: The behavior of using
nlags=None will change in release 0.13.Currently nlags=None is the same as
nlags="legacy", and so a sample-size lag length is used. After the next release,
the default will change to be the same as nlags="auto" which uses an automatic
lag length selection method. To silence this warning, either use "auto" or
"legacy"
  warnings.warn(msg, FutureWarning)
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1881: InterpolationWarning: The test
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is smaller than the p-value returned.

  warnings.warn(
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1881: InterpolationWarning: The test
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is smaller than the p-value returned.

  warnings.warn(
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1881: InterpolationWarning: The test
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is smaller than the p-value returned.

  warnings.warn(
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1881: InterpolationWarning: The test
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is smaller than the p-value returned.

  warnings.warn(
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1881: InterpolationWarning: The test
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is smaller than the p-value returned.

  warnings.warn(
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1881: InterpolationWarning: The test
```

```
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is smaller than the p-value returned.

  warnings.warn(
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1881: InterpolationWarning: The test
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is smaller than the p-value returned.

  warnings.warn(
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1881: InterpolationWarning: The test
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is smaller than the p-value returned.

  warnings.warn(
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1881: InterpolationWarning: The test
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is smaller than the p-value returned.

  warnings.warn(
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1881: InterpolationWarning: The test
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is smaller than the p-value returned.

  warnings.warn(
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1885: InterpolationWarning: The test
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.

  warnings.warn(
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1881: InterpolationWarning: The test
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is smaller than the p-value returned.

  warnings.warn(
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1881: InterpolationWarning: The test
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is smaller than the p-value returned.

  warnings.warn(
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1881: InterpolationWarning: The test
```

```
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is smaller than the p-value returned.

  warnings.warn(
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1881: InterpolationWarning: The test
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is smaller than the p-value returned.

  warnings.warn(
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1885: InterpolationWarning: The test
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.

  warnings.warn(
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1885: InterpolationWarning: The test
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.

  warnings.warn(
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1881: InterpolationWarning: The test
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is smaller than the p-value returned.

  warnings.warn(
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1881: InterpolationWarning: The test
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is smaller than the p-value returned.

  warnings.warn(
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1881: InterpolationWarning: The test
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is smaller than the p-value returned.

  warnings.warn(
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1881: InterpolationWarning: The test
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is smaller than the p-value returned.

  warnings.warn(
```

1 결론 : 22개의 시계열은 Non-stationary입니다

ADF Test와 KPSS Test를 모두 사용하여 시계열의 안정성에 대해 교차 확인합니다. 여기에 있는 모든 시계열 데이터가 안정적이지 않다는 결론을 내릴 수 있습니다. 차분 또는 비율로 시계열을 변환하여 안정적으로 만들어 보겠습니다.

2 일일 수익률로 변환하기

시계열을 안정적으로 만들기 위해서는 두개의 연속된 시계열 값의 차분(difference)을 취하는 방법이 가장 많이 사용되는 방법입니다. 하지만 주식 시계열 데이터에서는 차분 값보다는 수익률이 더욱 이해가 쉽기 때문에 일일 변화율을 계산하여 시계열 데이터를 안정화하는 방법을 추천합니다.

[42]:

```
for col in df.columns:
    df['r_'+col] = 100*df[col].pct_change()

df.dropna(inplace=True)
df.head()
```

[42]:

```
                 LG   Amore  Bukwang      Cj     CS  E_MART  Green_Cross  \
Date
2020-02-25  1294000  175000    11971  136500  17627  103500        17950
2020-02-26  1277000  171500    12108  137000  17651  106500        17950
2020-02-27  1246000  168500    11971  137500  17129  107500        18100
2020-02-28  1221000  161000    11468  133500  16868  105000        17550
2020-03-02  1210000  166500    12016  132000  16987  107000        17900

            Hanmi_Pharm  Hanmi_Semiconductor  Hansol_Chemical  ...  \
Date                                                          ...
2020-02-25       261164                 9670            96200  ...
2020-02-26       260673                 9680            96900  ...
2020-02-27       265091                 9110            96300  ...
2020-02-28       256255                 8660            92800  ...
2020-03-02       261655                 9120            93600  ...

            r_IS_DongSeo   r_Kakao  r_Kiwoom_Securities  \
Date
2020-02-25      4.494382  1.089918             1.005747
2020-02-26     -1.075269 -0.808625            -1.280228
2020-02-27      1.811594 -2.989130            -1.585014
2020-02-28     -4.626335 -3.641457            -3.221083
2020-03-02      0.373134  1.744186            -1.361573

            r_Korea_Petrochemical  r_Lotte_Chilsung_Beverage  r_Shinsegae  \
Date
2020-02-25               0.880088                   0.431034     4.089980
2020-02-26               0.327154                   0.000000    -0.196464
```

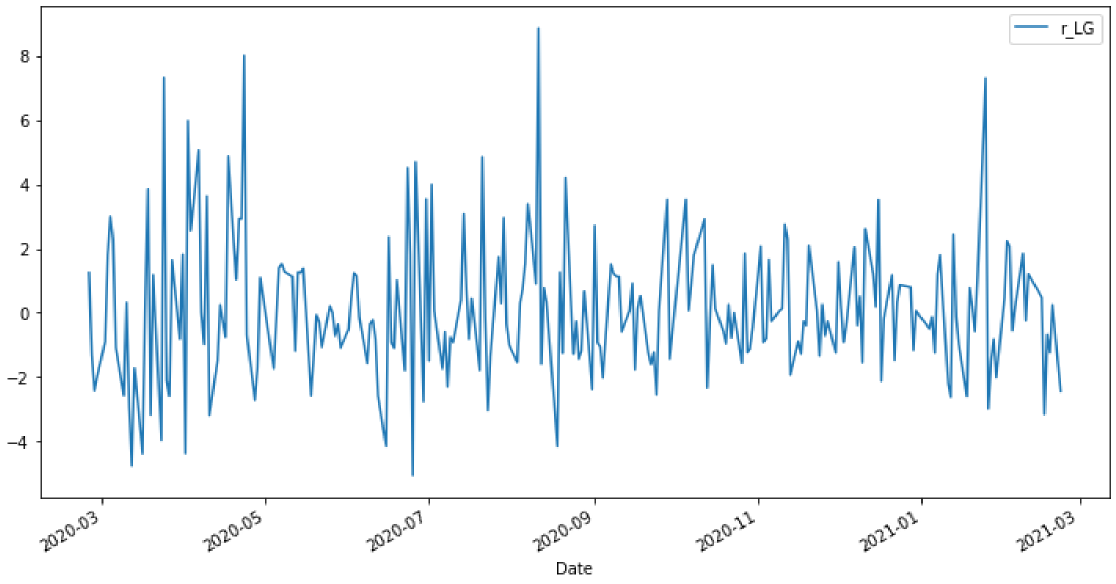
```
2020-02-27             -1.413043                     -1.716738    -1.968504
2020-02-28             -5.622933                     -1.746725    -5.220884
2020-03-02              0.000000                      0.888889     2.966102

               r_SKC  r_Solus_Advanced_Materials  r_Youngone    r_Yuhan
Date
2020-02-25  3.676471                    4.606526    3.859649  2.415691
2020-02-26 -1.950355                   -0.183486    3.885135 -0.234146
2020-02-27 -2.893309                   -4.411765    2.764228  2.364247
2020-02-28 -4.283054                   -4.807692   -0.316456 -1.156028
2020-03-02  1.945525                    5.454545    0.634921  3.972556

[5 rows x 44 columns]
```

[43]:

```
columns = ['r_LG']
df[columns].plot(figsize=(12.2,6.4))
```

[43]:

```
<AxesSubplot:xlabel='Date'>
```



[44]:

```
n_obs = 20
df_train, df_test = df[0:-n_obs], df[-n_obs:]

from statsmodels.tsa.stattools import adfuller

# ADF Test는 시계열이 안정적(Stationary)인지 여부를 확인하는데 이용됩니다.
# p-value>0.05이면 귀무 가설을 기각할 수 없습니다. -> 안정x
# 모든 주식이 귀무 가설을 기각할 수 없으므로 안정적이지 않습니다.
```

```
print('ADF Test: LG_Household_Health_Care_Ltd Time series')
adf_test(df_train['r_LG'])

print('\n\nADF Test: Amorepacific_Corp')
adf_test(df_train['r_Amore'])

print('\n\nADF Test: Bukwang_Pharmaceutical_Co_Ltd Time series')
adf_test(df_train['r_Bukwang'])

print('\n\nADF Test: Cj_Logistics_Corporation Time series')
adf_test(df_train['r_Cj'])

print('\n\nADF Test: CS_Wind_Corporation Time series')
adf_test(df_train['r_CS'])

print('\n\nADF Test: E_MART Time series')
adf_test(df_train['r_E_MART'])

print('\n\nADF Test: Green_Cross_Holdings_Corporation Time series')
adf_test(df_train['r_Green_Cross'])

print('\n\nADF Test: Hanmi_Pharm_Co_Ltd Time series')
adf_test(df_train['r_Hanmi_Pharm'])

print('\n\nADF Test: Hanmi_Semiconductor_Co_Ltd Time series')
adf_test(df_train['r_Hanmi_Semiconductor'])

print('\n\nADF Test: Hansol_Chemical_Co_Ltd Time series')
adf_test(df_train['r_Hansol_Chemical'])

print('\n\nADF Test: HMM Time series')
adf_test(df_train['r_HMM'])

print('\n\nADF Test: Hyundai_Heavy_Industries Time series')
adf_test(df_train['r_Hyundai_Heavy_Industries'])

print('\n\nADF Test: IS_DongSeo Time series')
adf_test(df_train['r_IS_DongSeo'])

print('\n\nADF Test: Kakao Time series')
adf_test(df_train['r_Kakao'])

print('\n\nADF Test: Kiwoom_Securities Time series')
adf_test(df_train['r_Kiwoom_Securities'])

print('\n\nADF Test: Korea_Petrochemical Time series')
```

```python
adf_test(df_train['r_Korea_Petrochemical'])

print('\n\nADF Test: Lotte_Chilsung_Beverage Time series')
adf_test(df_train['r_Lotte_Chilsung_Beverage'])

print('\n\nADF Test: Shinsegae Time series')
adf_test(df_train['r_Shinsegae'])

print('\n\nADF Test: SKC Time series')
adf_test(df_train['r_SKC'])

print('\n\nADF Test: Solus_Advanced_Materials Time series')
adf_test(df_train['r_Solus_Advanced_Materials'])

print('\n\nADF Test: Youngone Time series')
adf_test(df_train['r_Youngone'])

print('\n\nADF Test: Yuhan Time series')
adf_test(df_train['r_Yuhan'])
```

```
ADF Test: LG_Household_Health_Care_Ltd Time series
ADF Statistics: -7.506768
p-value: 0.000000
Critical values:
        1%: -3.460
        5%: -2.875
        10%: -2.574


ADF Test: Amorepacific_Corp
ADF Statistics: -17.243076
p-value: 0.000000
Critical values:
        1%: -3.460
        5%: -2.874
        10%: -2.574


ADF Test: Bukwang_Pharmaceutical_Co_Ltd Time series
ADF Statistics: -16.889573
p-value: 0.000000
Critical values:
        1%: -3.460
        5%: -2.874
        10%: -2.574


ADF Test: Cj_Logistics_Corporation Time series
ADF Statistics: -8.856934
p-value: 0.000000
Critical values:
        1%: -3.460
        5%: -2.875
        10%: -2.574


ADF Test: CS_Wind_Corporation Time series
ADF Statistics: -13.531500
p-value: 0.000000
Critical values:
        1%: -3.460
        5%: -2.874
        10%: -2.574


ADF Test: E_MART Time series
ADF Statistics: -10.353843
p-value: 0.000000
Critical values:
        1%: -3.460
        5%: -2.875
        10%: -2.574


ADF Test: Green_Cross_Holdings_Corporation Time series
ADF Statistics: -13.477417
p-value: 0.000000
Critical values:
        1%: -3.460
        5%: -2.874
        10%: -2.574


ADF Test: Hanmi_Pharm_Co_Ltd Time series
ADF Statistics: -14.797304
p-value: 0.000000
Critical values:
        1%: -3.460
        5%: -2.874
        10%: -2.574


ADF Test: Hanmi_Semiconductor_Co_Ltd Time series
ADF Statistics: -7.054677
p-value: 0.000000
Critical values:
        1%: -3.460
        5%: -2.875
        10%: -2.574


ADF Test: Hansol_Chemical_Co_Ltd Time series
ADF Statistics: -16.130220
p-value: 0.000000
Critical values:
        1%: -3.460
        5%: -2.874
        10%: -2.574


ADF Test: HMM Time series
ADF Statistics: -6.984588
p-value: 0.000000
Critical values:
        1%: -3.460
        5%: -2.875
        10%: -2.574


ADF Test: Hyundai_Heavy_Industries Time series
ADF Statistics: -14.536133
p-value: 0.000000
Critical values:
        1%: -3.460
        5%: -2.874
        10%: -2.574


ADF Test: IS_DongSeo Time series
ADF Statistics: -16.046344
p-value: 0.000000
Critical values:
        1%: -3.460
        5%: -2.874
        10%: -2.574


ADF Test: Kakao Time series
ADF Statistics: -9.046992
p-value: 0.000000
Critical values:
        1%: -3.460
        5%: -2.875
        10%: -2.574
```

```
ADF Test: Kiwoom_Securities Time series
ADF Statistics: -14.740345
p-value: 0.000000
Critical values:
        1%: -3.460
        5%: -2.874
        10%: -2.574


ADF Test: Korea_Petrochemical Time series
ADF Statistics: -7.139231
p-value: 0.000000
Critical values:
        1%: -3.460
        5%: -2.875
        10%: -2.574


ADF Test: Lotte_Chilsung_Beverage Time series
ADF Statistics: -16.881165
p-value: 0.000000
Critical values:
        1%: -3.460
        5%: -2.874
        10%: -2.574


ADF Test: Shinsegae Time series
ADF Statistics: -16.587463
p-value: 0.000000
Critical values:
        1%: -3.460
        5%: -2.874
        10%: -2.574


ADF Test: SKC Time series
ADF Statistics: -14.424068
p-value: 0.000000
Critical values:
        1%: -3.460
        5%: -2.874
        10%: -2.574


ADF Test: Solus_Advanced_Materials Time series
ADF Statistics: -7.285311
p-value: 0.000000
Critical values:
        1%: -3.460
        5%: -2.875
        10%: -2.574


ADF Test: Youngone Time series
ADF Statistics: -16.034723
p-value: 0.000000
Critical values:
        1%: -3.460
        5%: -2.874
        10%: -2.574


ADF Test: Yuhan Time series
ADF Statistics: -15.885768
p-value: 0.000000
Critical values:
        1%: -3.460
        5%: -2.874
        10%: -2.574
```

[45]:

```python
from statsmodels.tsa.stattools import kpss

# p-value<0.05이므로 귀무 가설을 기각할 수 없습니다. -> 안정x

def kpss_test(df):
    statistic, p_value, n_lags, critical_values = kpss(df.values)

    print(f'KPSS Statistic: {statistic}')
    print(f'p-value: {p_value}')
    print(f'num lags: {n_lags}')
    print('Critial Values:')
    for key, value in critical_values.items():
        print(f'{key} : {value}')

print('KPSS Test: LG_Household_Health_Care_Ltd Time series Time series')
kpss_test(df_train['r_LG'])

print('\n\nKPSS Test: Amorepacific_Corp Time series')
kpss_test(df_train['r_Amore'])

print('\n\nKPSS Test: Bukwang_Pharmaceutical_Co_Ltd Time series')
kpss_test(df_train['r_Bukwang'])
```

```
print('\n\nKPSS Test: Cj_Logistics_Corporation Time series')
kpss_test(df_train['r_Cj'])

print('\n\nKPSS Test: CS_Wind_Corporation Time series')
kpss_test(df_train['r_CS'])

print('\n\nKPSS Test: E_MART Time series')
kpss_test(df_train['r_E_MART'])

print('\n\nKPSS Test: Green_Cross_Holdings_Corporation Time series')
kpss_test(df_train['r_Green_Cross'])

print('\n\nKPSS Test: Hanmi_Pharm_Co_Ltd Time series')
kpss_test(df_train['r_Hanmi_Pharm'])

print('\n\nKPSS Test: Hanmi_Semiconductor_Co_Ltd Time series')
kpss_test(df_train['r_Hanmi_Semiconductor'])

print('\n\nKPSS Test: Hansol_Chemical_Co_Ltd Time series')
kpss_test(df_train['r_Hansol_Chemical'])

print('\n\nKPSS Test: HMM Time series')
kpss_test(df_train['r_HMM'])

print('\n\nKPSS Test: Hyundai_Heavy_Industries Time series')
kpss_test(df_train['r_Hyundai_Heavy_Industries'])

print('\n\nKPSS Test: IS_DongSeo Time series')
kpss_test(df_train['r_IS_DongSeo'])

print('\n\nKPSS Test: Kakao Time series')
kpss_test(df_train['r_Kakao'])

print('\n\nKPSS Test: Kiwoom_Securities Time series')
kpss_test(df_train['r_Kiwoom_Securities'])

print('\n\nKPSS Test: Korea_Petrochemical Time series')
kpss_test(df_train['r_Korea_Petrochemical'])

print('\n\nKPSS Test: Lotte_Chilsung_Beverage Time series')
kpss_test(df_train['r_Lotte_Chilsung_Beverage'])

print('\n\nKPSS Test: Shinsegae Time series')
kpss_test(df_train['r_Shinsegae'])

print('\n\nKPSS Test: SKC Time series')
```

```
kpss_test(df_train['r_SKC'])

print('\n\nKPSS Test: Solus_Advanced_Materials Time series')
kpss_test(df_train['r_Solus_Advanced_Materials'])

print('\n\nKPSS Test: Youngone Time series')
kpss_test(df_train['r_Youngone'])

print('\n\nKPSS Test: Yuhan Time series')
kpss_test(df_train['r_Yuhan'])
```

```
KPSS Test: LG_Household_Health_Care_Ltd Time series Time series
KPSS Statistic: 0.051607902255553
p-value: 0.1
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: Amorepacific_Corp Time series
KPSS Statistic: 0.3192828485306775
p-value: 0.1
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: Bukwang_Pharmaceutical_Co_Ltd Time series
KPSS Statistic: 0.52704989829505
p-value: 0.03557434723084459
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: Cj_Logistics_Corporation Time series
KPSS Statistic: 0.07159157445973315
p-value: 0.1
num lags: 15
```

```
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: CS_Wind_Corporation Time series
KPSS Statistic: 0.1541838296403735
p-value: 0.1
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: E_MART Time series
KPSS Statistic: 0.07245173919939465
p-value: 0.1
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: Green_Cross_Holdings_Corporation Time series
KPSS Statistic: 0.06293890269144133
p-value: 0.1
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: Hanmi_Pharm_Co_Ltd Time series
KPSS Statistic: 0.1005597271332984
p-value: 0.1
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
```

```
1% : 0.739


KPSS Test: Hanmi_Semiconductor_Co_Ltd Time series
KPSS Statistic: 0.20475392372172097
p-value: 0.1
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: Hansol_Chemical_Co_Ltd Time series
KPSS Statistic: 0.1248132590633164
p-value: 0.1
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: HMM Time series
KPSS Statistic: 0.0886693285277386
p-value: 0.1
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: Hyundai_Heavy_Industries Time series
KPSS Statistic: 0.0749945538717231
p-value: 0.1
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: IS_DongSeo Time series
KPSS Statistic: 0.06358449761171903
p-value: 0.1
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: Kakao Time series
KPSS Statistic: 0.11750460209371519
p-value: 0.1
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: Kiwoom_Securities Time series
KPSS Statistic: 0.06414772158967477
p-value: 0.1
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: Korea_Petrochemical Time series
KPSS Statistic: 0.07020737883346326
p-value: 0.1
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: Lotte_Chilsung_Beverage Time series
KPSS Statistic: 0.26998823534369554
p-value: 0.1
num lags: 15
Critial Values:
```

```
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: Shinsegae Time series
KPSS Statistic: 0.12989840609159523
p-value: 0.1
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: SKC Time series
KPSS Statistic: 0.10256917710442039
p-value: 0.1
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: Solus_Advanced_Materials Time series
KPSS Statistic: 0.0510554335007387
p-value: 0.1
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: Youngone Time series
KPSS Statistic: 0.05649380035620086
p-value: 0.1
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739
```

```
KPSS Test: Yuhan Time series
KPSS Statistic: 0.046599077124183375
p-value: 0.1
num lags: 15
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739
```

```
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1850: FutureWarning: The behavior of using
nlags=None will change in release 0.13.Currently nlags=None is the same as
nlags="legacy", and so a sample-size lag length is used. After the next release,
the default will change to be the same as nlags="auto" which uses an automatic
lag length selection method. To silence this warning, either use "auto" or
"legacy"
  warnings.warn(msg, FutureWarning)
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1885: InterpolationWarning: The test
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.

  warnings.warn(
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1885: InterpolationWarning: The test
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.

  warnings.warn(
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1885: InterpolationWarning: The test
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.

  warnings.warn(
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1885: InterpolationWarning: The test
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.

  warnings.warn(
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1885: InterpolationWarning: The test
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

```
  warnings.warn(
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1885: InterpolationWarning: The test
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.

  warnings.warn(
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1885: InterpolationWarning: The test
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.

  warnings.warn(
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1885: InterpolationWarning: The test
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.

  warnings.warn(
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1885: InterpolationWarning: The test
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.

  warnings.warn(
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1885: InterpolationWarning: The test
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.

  warnings.warn(
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1885: InterpolationWarning: The test
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.

  warnings.warn(
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1885: InterpolationWarning: The test
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.

  warnings.warn(
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1885: InterpolationWarning: The test
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

```
  warnings.warn(
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1885: InterpolationWarning: The test
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.

  warnings.warn(
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1885: InterpolationWarning: The test
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.

  warnings.warn(
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1885: InterpolationWarning: The test
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.

  warnings.warn(
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1885: InterpolationWarning: The test
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.

  warnings.warn(
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1885: InterpolationWarning: The test
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.

  warnings.warn(
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1885: InterpolationWarning: The test
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.

  warnings.warn(
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1885: InterpolationWarning: The test
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.

  warnings.warn(
C:\ProgramData\Anaconda3\envs\muiiya\lib\site-
packages\statsmodels\tsa\stattools.py:1885: InterpolationWarning: The test
statistic is outside of the range of p-values available in the
look-up table. The actual p-value is greater than the p-value returned.
```

```
  warnings.warn(
```

[46]:

```
from dateutil.parser import parse
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import pandas as pd
```

[ ]: