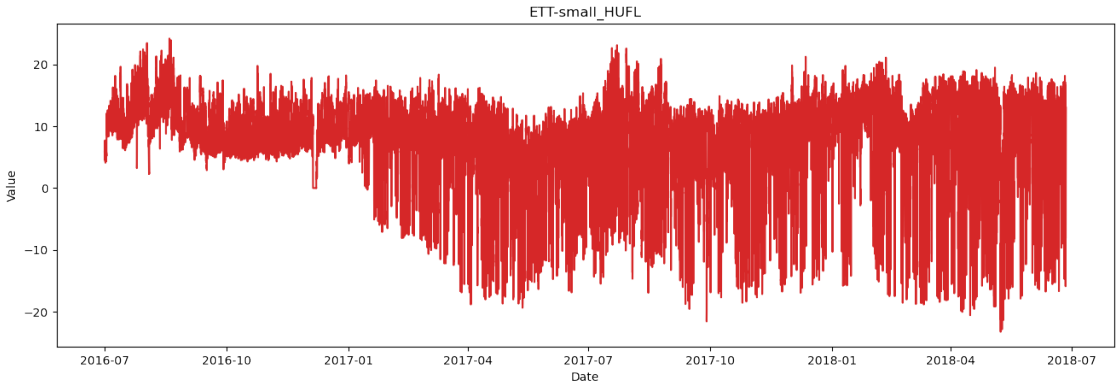
Stationary_Test_Informer_ETDataset_ETT-small_ETTm2.csv

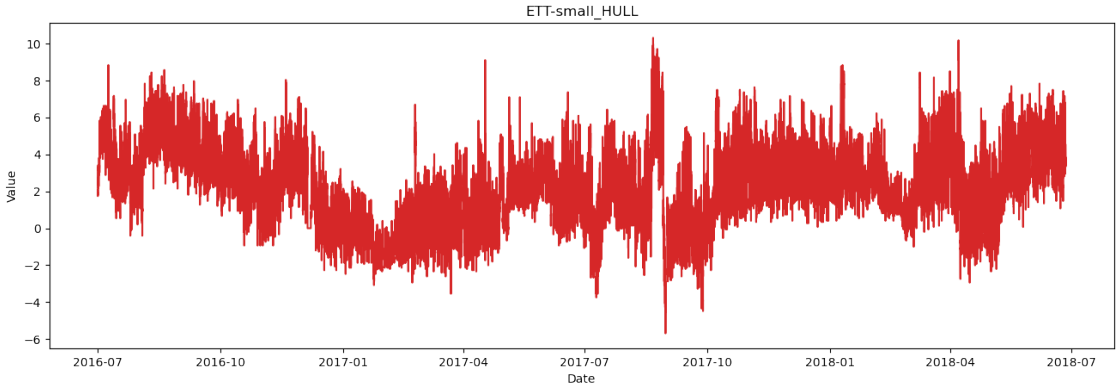
February 23, 2021

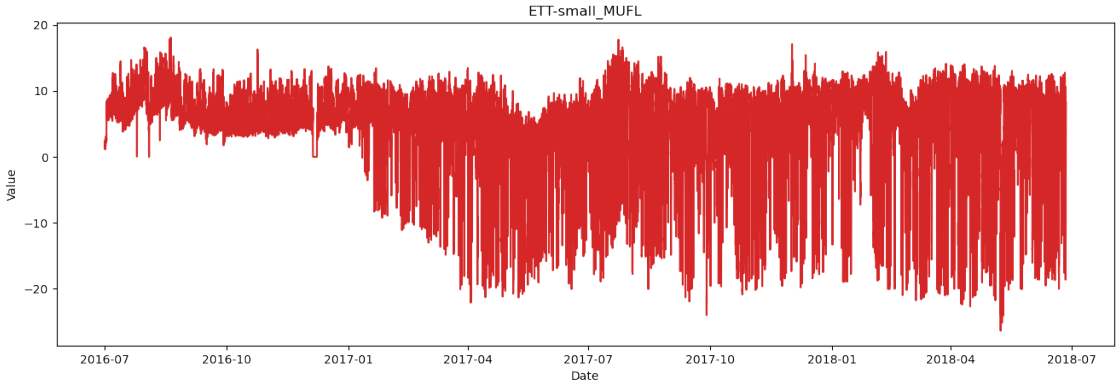
[1]:
```
from dateutil.parser import parse
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import pandas as pd
```

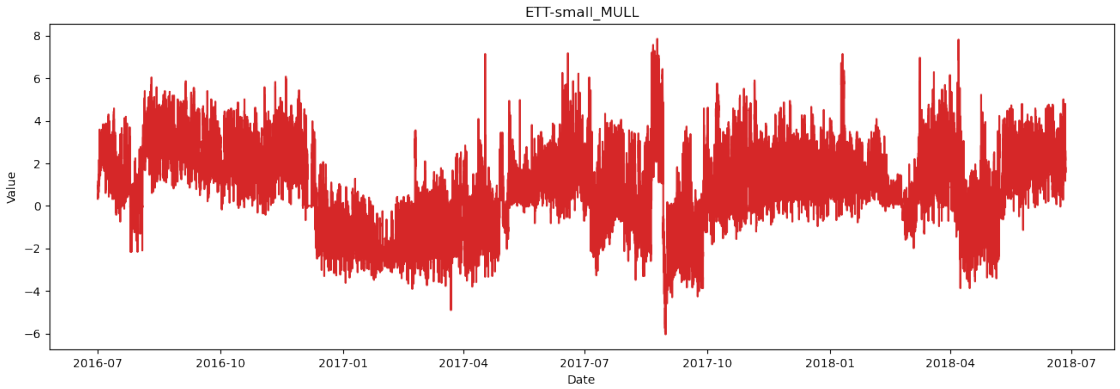
[2]:
```
df = pd.read_csv('https://raw.githubusercontent.com/zhouhaoyi/ETDataset/
↪11ab373cf9c9f5be7698e219a5a170e1b1c8a930/ETT-small/ETTm2.csv',
parse_dates=['date'], index_col='date')
```

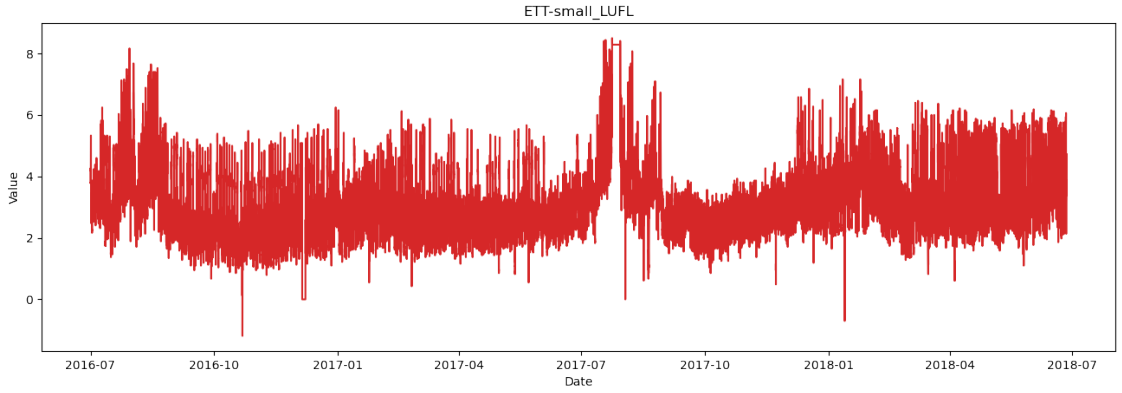
[3]:
```
def plot_df(df, x, y, title="", xlabel='Date', ylabel='Value', dpi=100):
    plt.figure(figsize=(16,5), dpi=dpi)
    plt.plot(x, y, color='tab:red')
    plt.gca().set(title=title, xlabel=xlabel, ylabel=ylabel)
    plt.show()

plot_df(df, x=df.index, y=df.HUFL, title='ETT-small_HUFL')
plot_df(df, x=df.index, y=df.HULL, title='ETT-small_HULL')
plot_df(df, x=df.index, y=df.MUFL, title='ETT-small_MUFL')
plot_df(df, x=df.index, y=df.MULL, title='ETT-small_MULL')
plot_df(df, x=df.index, y=df.LUFL, title='ETT-small_LUFL')
plot_df(df, x=df.index, y=df.LULL, title='ETT-small_LULL')
plot_df(df, x=df.index, y=df.OT, title='ETT-small_OT')
```

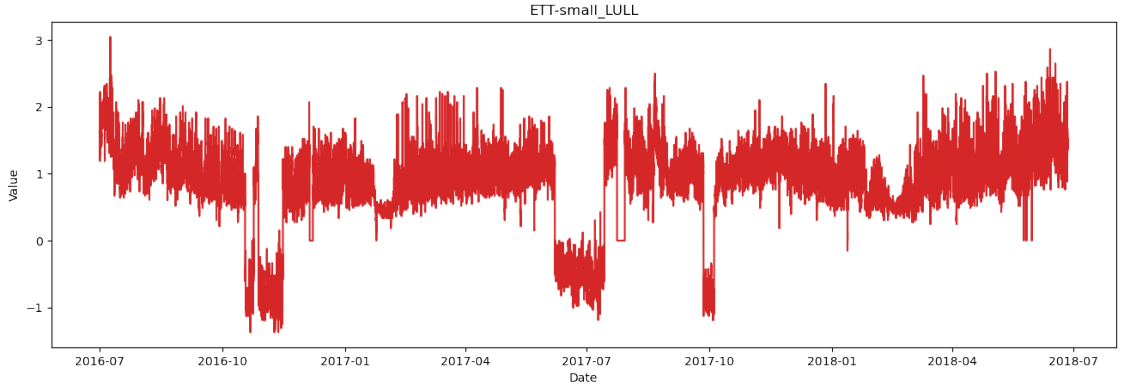


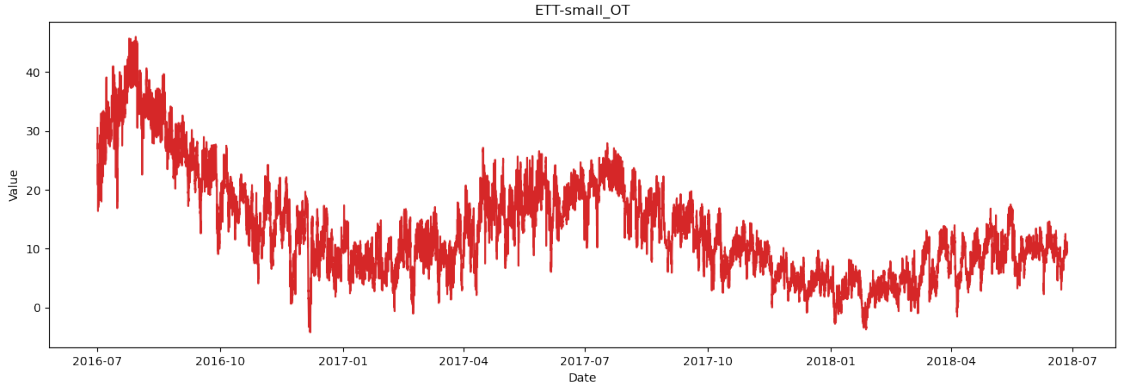

ETT-small_HULL
Value
Date
2016-07
2016-10
2017-01
2017-04
2017-07
2017-10
2018-01
2018-04
2018-07


ETT-small_MUFL
Value
Date
2016-07
2016-10
2017-01
2017-04
2017-07
2017-10
2018-01
2018-04
2018-07


ETT-small_MULL
Value
Date
2016-07
2016-10
2017-01
2017-04
2017-07
2017-10
2018-01
2018-04
2018-07







[4]:
```
n_obs = 20
df_train, df_test = df[0:-n_obs], df[-n_obs:]

from statsmodels.tsa.stattools import adfuller
```

```
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: ETT-small_HUFL Time series')
adf_test(df_train['HUFL'])

print('\n\nADF Test: ETT-small_HULL Time series')
adf_test(df_train['HULL'])

print('\n\nADF Test: ETT-small_MUFL Time series')
adf_test(df_train['MUFL'])

print('\n\nADF Test: ETT-small_MULL Time series')
adf_test(df_train['MULL'])

print('\n\nADF Test: ETT-small_LUFL Time series')
adf_test(df_train['LUFL'])

print('\n\nADF Test: ETT-small_LULL Time series')
adf_test(df_train['LULL'])

print('\n\nADF Test: ETT-small_OT Time series')
adf_test(df_train['OT'])
```

```
ADF Test: ETT-small_HUFL Time series
ADF Statistics: -26.710571
p-value: 0.000000
Critical values:
        1%: -3.430
        5%: -2.862
        10%: -2.567


ADF Test: ETT-small_HULL Time series
ADF Statistics: -11.818831
p-value: 0.000000
Critical values:
        1%: -3.430
        5%: -2.862
        10%: -2.567
```

```
ADF Test: ETT-small_MUFL Time series
ADF Statistics: -27.317251
p-value: 0.000000
Critical values:
        1%: -3.430
        5%: -2.862
        10%: -2.567


ADF Test: ETT-small_MULL Time series
ADF Statistics: -11.414144
p-value: 0.000000
Critical values:
        1%: -3.430
        5%: -2.862
        10%: -2.567


ADF Test: ETT-small_LUFL Time series
ADF Statistics: -16.320195
p-value: 0.000000
Critical values:
        1%: -3.430
        5%: -2.862
        10%: -2.567


ADF Test: ETT-small_LULL Time series
ADF Statistics: -7.050702
p-value: 0.000000
Critical values:
        1%: -3.430
        5%: -2.862
        10%: -2.567


ADF Test: ETT-small_OT Time series
ADF Statistics: -4.193942
p-value: 0.000675
Critical values:
        1%: -3.430
        5%: -2.862
        10%: -2.567
```

[5]:
```
from statsmodels.tsa.stattools import kpss
```

```python
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: ETT-small_HUFL Time series')
kpss_test(df_train['HUFL'])

print('\n\nKPSS Test: ETT-small_HULL Time series')
kpss_test(df_train['HULL'])

print('\n\nKPSS Test: ETT-small_MUFL Time series')
kpss_test(df_train['MUFL'])

print('\n\nKPSS Test: ETT-small_MULL Time series')
kpss_test(df_train['MULL'])

print('\n\nKPSS Test: ETT-small_LUFL Time series')
kpss_test(df_train['LUFL'])

print('\n\nKPSS Test: ETT-small_LULL Time series')
kpss_test(df_train['LULL'])

print('\n\nKPSS Test: ETT-small_OT Time series')
kpss_test(df_train['OT'])
```

```
KPSS Test: ETT-small_HUFL Time series
KPSS Statistic: 11.446013640510193
p-value: 0.01
num lags: 62
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: ETT-small_HULL Time series
KPSS Statistic: 9.029131415993245
p-value: 0.01
num lags: 62
Critial Values:
10% : 0.347
```

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


KPSS Test: ETT-small_MUFL Time series
KPSS Statistic: 15.042353025571492
p-value: 0.01
num lags: 62
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: ETT-small_MULL Time series
KPSS Statistic: 7.223645559214652
p-value: 0.01
num lags: 62
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: ETT-small_LUFL Time series
KPSS Statistic: 6.67760239203023
p-value: 0.01
num lags: 62
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739


KPSS Test: ETT-small_LULL Time series
KPSS Statistic: 5.005220278182344
p-value: 0.01
num lags: 62
Critial Values:
10% : 0.347
5% : 0.463
2.5% : 0.574
1% : 0.739
```

```
KPSS Test: ETT-small_OT Time series
KPSS Statistic: 44.80126808525964
p-value: 0.01
num lags: 62
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(
```

[ ]: