

Definição de série estacionária:

"A stationary time series is one whose properties do not depend on the time at which the series is observed. Thus, time series with trends, or with seasonality, are not stationary — the trend and seasonality will affect the value of the time series at different times." [https://otexts.com/fpp2/stationarity.html#fn14]

Em outras palavras, é uma série cujas propriedades estatísticas (e.g., média, variância, auto-correlação) são constantes ao longo do tempo.

Definição de série tendência-estacionária:

"In the statistical analysis of time series, a trend-stationary process is a stochastic process from which an underlying trend (function solely of time) can be removed, leaving a stationary process. The trend does not have to be linear." [https://en.wikipedia.org/wiki/Trend-stationary_process]*

Em outras palavras, é uma série que eliminada a tendência os resíduos têm um comportamento estacionário.

Outros materiais:

- Mais exemplos em python: https://machinelearningmastery.com/time-series-data-stationary-python/
- Vale a pena ler esse capítulo para entender mais sobre estacionariedade: https://otexts.com/fpp2/stationarity.html#fn14

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.tsa.stattools import adfuller

starbucks = pd.read_csv('starbucks.csv', index_col=["Date"], parse_dates=["Date"])
starbucks
```

Out[1]:

	Close	Volume
Date		
2015-01-02	38.0061	6906098
2015-01-05	37.2781	11623796
2015-01-06	36.9748	7664340
2015-01-07	37.8848	9732554
2015-01-08	38.4961	13170548
...
2018-12-24	60.5600	6323252
2018-12-26	63.0800	16646238
2018-12-27	63.2000	11308081
2018-12-28	63.3900	7712127
2018-12-31	64.4000	7690183

1006 rows × 2 columns

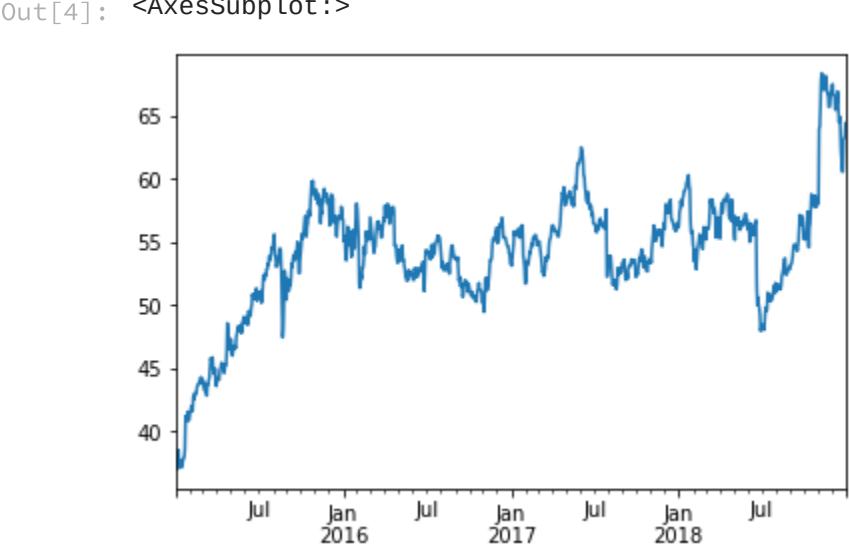
```
In [2]: starbucks.shape
```

Out[2]: (1006, 2)

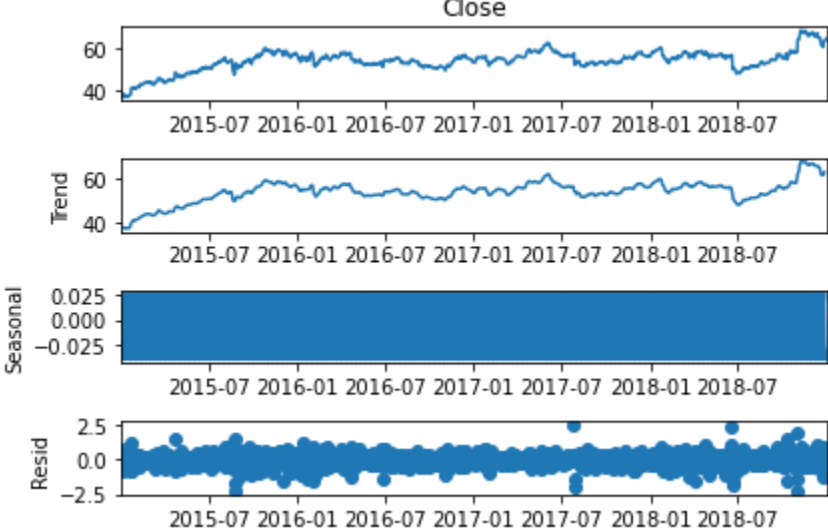
```
In [3]: idx = pd.date_range(start=starbucks.index.min(), end=starbucks.index.max(), freq="B")
starbucks = starbucks.reindex(idx)
starbucks.fillna(method='ffill', inplace=True)
starbucks.shape
```

Out[3]: (1042, 2)

```
In [4]: starbucks["Close"].plot()
```

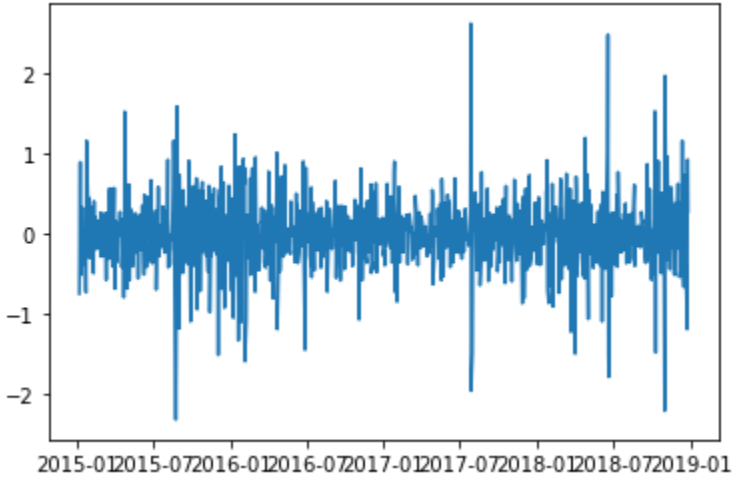


```
In [5]: decomp = seasonal_decompose(starbucks["Close"], model='add', period=5)
decomp.plot()
plt.show()
```



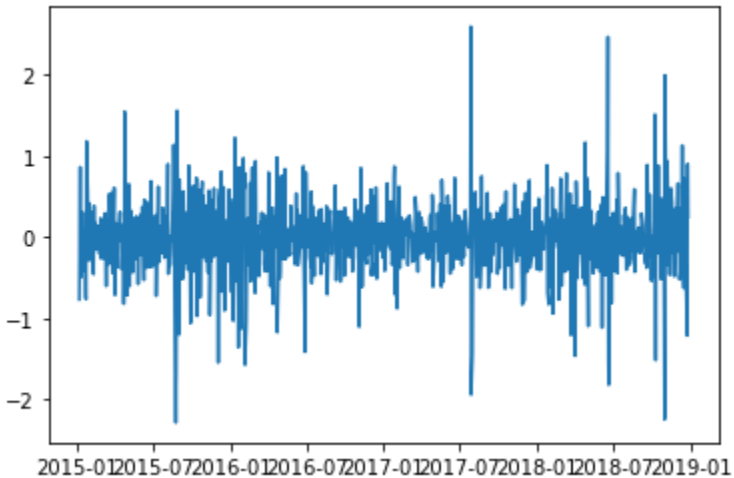
Subtraindo a tendência da série temporal, já dá para observar a estacionariedade:

```
In [6]: # The additive model is Y[t] = T[t] + S[t] + e[t] (fonte: https://www.statsmodels.org,
# removendo a tendencia
without_trend = decomp.observed - decomp.trend
plt.plot(without_trend)
plt.show()
```



O resíduo já nos mostra isso:

```
In [7]: plt.plot(decomp.resid)
plt.show()
```



Observe que os dois plots acima mostra que a série é tendência estacionária. Contudo, observe que são plots diferentes:

```
In [8]: print(decomp.resid[10:20]) # resíduo, e[t] = Y[t] - T[t] - S[t]
print(without_trend[10:20]) # série sem tendência, Z[t] = Y[t] - T[t]
```

```
2015-01-16    0.080057
2015-01-19    0.016261
2015-01-20   -0.047584
2015-01-21   -0.734600
2015-01-22   -0.764555
2015-01-23    1.180017
2015-01-26    0.565041
2015-01-27    0.022896
2015-01-28   -0.281440
2015-01-29    0.417965
Freq: B, Name: resid, dtype: float64
2015-01-16    0.05648
2015-01-19   -0.02472
2015-01-20   -0.03266
2015-01-21   -0.71264
2015-01-22   -0.73688
2015-01-23    1.15644
2015-01-26    0.52406
2015-01-27    0.03782
2015-01-28   -0.25948
2015-01-29    0.44564
Freq: B, dtype: float64
```

Como a série starbucks é tendência-estacionária rejeitamos H_0 no teste de Dickey–Fuller:

Como visto em aula:

H_0 : a série é não estacionária e contém uma raiz unitária

H_1 : a série é estacionária ou tendência-estacionária

- Quando o valor-p é pequeno ($p < 0.05$, por exemplo), rejeitamos H_0 e portanto há evidências de que a série é estacionária ou tendência-estacionária.
- Se o valor-p for grande ($p \geq 0.05$, por exemplo), não rejeitamos H_0 e a série não é estacionária e contém uma raiz unitária.

```
In [9]: result = adfuller(starbucks["Close"], autolag='AIC')
print('ADF Statistic: %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))
```

```
ADF Statistic: -3.035274
p-value: 0.031729
Critical Values:
1%: -3.437
5%: -2.864
10%: -2.568
```

Há evidências de que a série é estacionária ou tendência-estacionária.