$$X(t,\Omega) = \{X(t,\omega_i)|\omega_i \in \Omega\} = \{x_1(t), x_2(t), \dots\}$$

$$X(t,\omega_i) = x_i(t)$$

$$X(t_0,\Omega) = \{X(t,\omega_i)|\omega_i \in \Omega\} = \{x_1(t_0), x_2(t_0), \dots\}$$

$$X(t_0,\omega_i) = x_i(t_0)$$

$$y'_t = y_t - y_{t-1}$$

Si quiero eliminar una tendencia anual, puedo calcular la diferencia de orden 12 (meses):

$$y'_t = y_t - y_{t-12}$$

$$y''_t = y'_t - y'_{t-1} = y_t - y_{t-1} - y_{t-12} + y_{t-12-1}$$

$$By_t = y_{t-1}$$

$$B^{12}y_t = y_{t-12}$$

$$y'_t = y_t - By_t = (1-B)y_t$$

$$y''_t = (1-B)(1-B^{12})y_t$$

Exponential Smoothing

$$\hat{y}_{t+1|t} = \alpha y_t + \alpha(1-\alpha)y_{t-1} + \alpha(1-\alpha)^2 y_{t-2} + \dots$$

donde alpha es el parámetro de suavizado que se encuentra en el intervalo (0,1). Se requiere un valor inicial para t01.

Usando notación de componentes, y denotando el valor inicial como n_0 tenemos que:

$$\hat{y}_{t+h|t} = n_t$$

$$n_t = \alpha y_t + \alpha(1-\alpha)n_{t-1}$$

```
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.api import SimpleExpSmoothing

df = [ 420.735,392.943, 440.593, 450.037, 430.345, 471.033, 423.456, 458.989,
470.767, 420.368, 432.456, 487.409, 458.989, 467.765, 432.341, 399.563, 412.324,
398.452, 419.452, 470.567]

index=pd.date_range(start='2000', end='2020',freq='A')
data= pd.DataFrame(df, index)
data.plot()
```

```
plt.show()


#First Instance
ins1 = SimpleExpSmoothing(data).fit(smoothing_level=0.2,optimized=False)
ins_cast1 = ins1.forecast(3).rename('alpha=0.2')
ax = data.plot(marker='o', color='black', figsize=(12,8), legend=True)
ins_cast1.plot(marker='+', ax=ax, color='blue', legend=True)
ins1.fittedvalues.plot(marker='+', ax=ax, color='blue')

plt.show()

#Second Instance
ins2 = SimpleExpSmoothing(data).fit(smoothing_level=0.8,optimized=False)
ins_cast2 = ins2.forecast(3).rename('alpha=0.8')

#Third Instance
ins3 = SimpleExpSmoothing(data).fit()
ins_cast3 =
ins3.forecast(3).rename('alpha=%s'%ins3.model.params['smoothing_level'])

#After creating model we will visualize the plot
ax = data.plot(marker='o', color='black', figsize=(12,8), legend=True)

ins_cast1.plot(marker='+', ax=ax, color='blue', legend=True)
ins1.fittedvalues.plot(marker='+', ax=ax, color='blue')

ins_cast2.plot(marker='o', ax=ax, color='red', legend=True)
ins2.fittedvalues.plot(marker='o', ax=ax, color='red')

ins_cast3.plot(marker='*', ax=ax, color='green', legend=True)
ins3.fittedvalues.plot(marker='*', ax=ax, color='green')

plt.show()
```

```
#Holt lineal
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.api import SimpleExpSmoothing
from statsmodels.tsa.api import ExponentialSmoothing, SimpleExpSmoothing, Holt

data = [
    17.5534,
    21.86,
    23.8866,
    26.9293,
    26.8885,
    28.8314,
    30.0751,
    30.9535,
```

```
    30.1857,
    31.5797,
    32.5776,
    33.4774,
    39.0216,
    41.3864,
    41.5966,
]
index = pd.date_range(start="1990", end="2005", freq="A")
air = pd.Series(data, index)

fit1 = Holt(air).fit(smoothing_level=0.8, smoothing_slope=0.2, optimized=False)
fcast1 = fit1.forecast(5).rename("Holt's linear trend")
fit2 = Holt(air, exponential=True).fit(smoothing_level=0.8, smoothing_slope=0.4,
optimized=False)
fcast2 = fit2.forecast(5).rename("Exponential trend")
fit3 = Holt(air, damped=True).fit(smoothing_level=0.8, smoothing_slope=0.7,
optimized=False)
fcast3 = fit3.forecast(5).rename("Additive damped trend")

plt.figure(figsize=(12, 8))
plt.plot(air, marker="o", color="black")
plt.plot(fit1.fittedvalues, color="blue")
(line1,) = plt.plot(fcast1, marker="o", color="blue")
plt.plot(fit2.fittedvalues, color="red")
(line2,) = plt.plot(fcast2, marker="o", color="red")
plt.plot(fit3.fittedvalues, color="green")
(line3,) = plt.plot(fcast3, marker="o", color="green")
plt.legend([line1, line2, line3], [fcast1.name, fcast2.name, fcast3.name])
plt.show()
```

```
#Holt-Winters Seasonal

data = [
    41.7275,
    24.0418,
    32.3281,
    37.3287,
    46.2132,
    29.3463,
    36.4829,
    42.9777,
    48.9015,
    31.1802,
    37.7179,
    40.4202,
    51.2069,
    31.8872,
    40.9783,
    43.7725,
    55.5586,
    33.8509,
    42.0764,
    45.6423,
    59.7668,
    35.1919,
```

```
    44.3197,
    47.9137,
]
index = pd.date_range(start="2005", end="2010-Q4", freq="QS-OCT")
aust = pd.Series(data, index)

fit1 = ExponentialSmoothing(
   aust,
   seasonal='mul',
   seasonal_periods=12
).fit()
fit2 = ExponentialSmoothing(
   aust,
   seasonal='add',
   seasonal_periods=12
).fit()


ax = aust.plot(
   figsize=(10, 6),
   marker="o",
   color="black",
   title="Forecasts from Holt-Winters' multiplicative method",
)
ax.set_ylabel("International visitor night in Australia (millions)")
ax.set_xlabel("Year")
fit1.fittedvalues.plot(ax=ax, style="--", color="red")
fit2.fittedvalues.plot(ax=ax, style="--", color="green")

fit1.forecast(8).rename("Holt-Winters (add-add-seasonal)").plot(
   ax=ax, style="--", marker="o", color="red", legend=True
)
fit2.forecast(8).rename("Holt-Winters (add-mul-seasonal)").plot(
   ax=ax, style="--", marker="o", color="green", legend=True
)

plt.show()
```