Using Exponential Smoothing Models in Python

Martin Burger
STATS PROGRAMMING TUTOR

www.r-tutorials.com

Exponential Smoothing

The most common analytical tool for univariate time series besides ARIMA

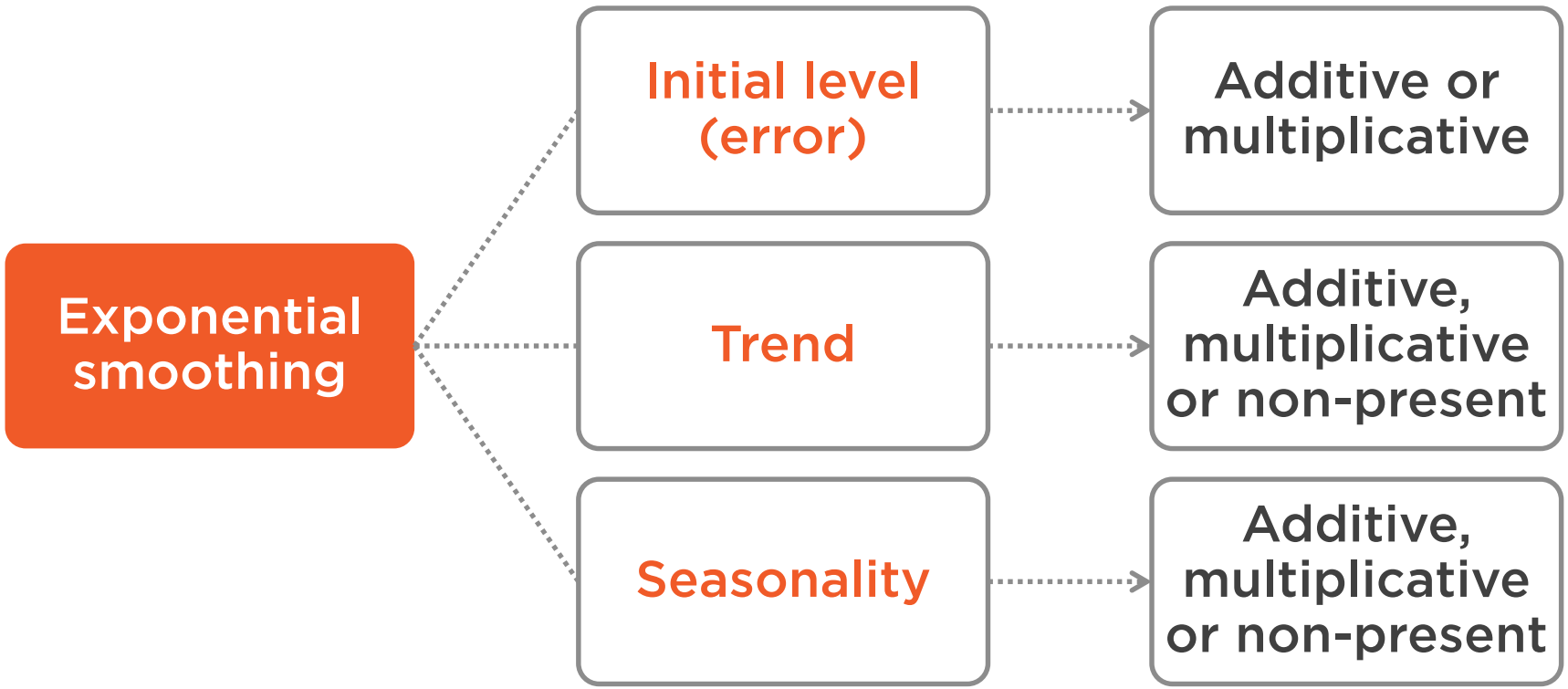
Exponential smoothing models theory

Demo in Python with the nottem dataset

- Identifying the model parameters
- Modulating the trend component
- Forecasting

Exponential Smoothing

Describing an Exponential Smoothing Model



Exponential Smoothing Model Types

Additive

Components are summed up

Multiplicative

Components are multiplied

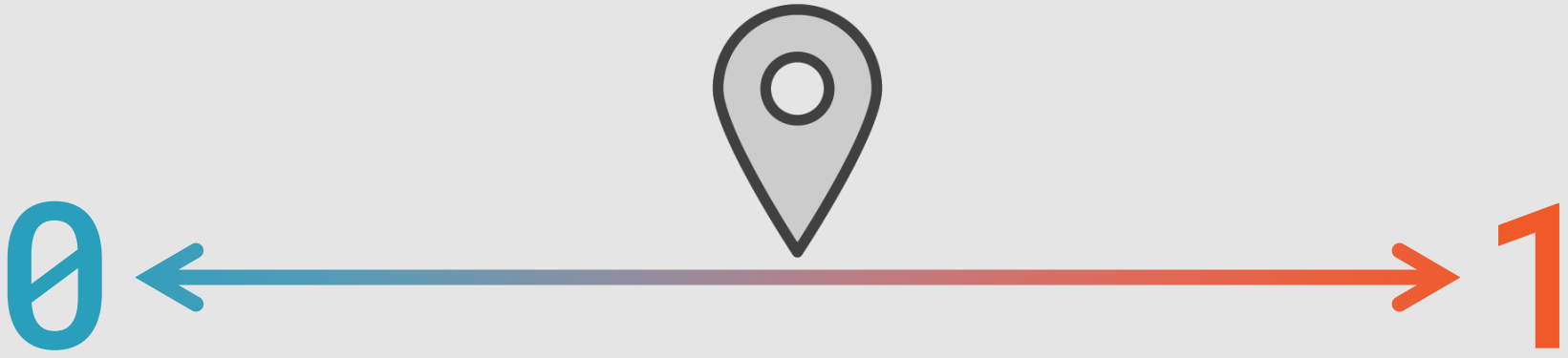
Non-present

Components are omitted

Exponential smoothing:

A model for univariate time series where fluctuations are smoothed out by weights which decrease exponentially over time.

Smoothing Coefficients Determine the Importance of a Time Lag



Smooth model

Reactive model

Smoothing Coefficients

Alpha: Initial level

Beta: Trend

Gamma: Seasonality

Phi: Damping parameter

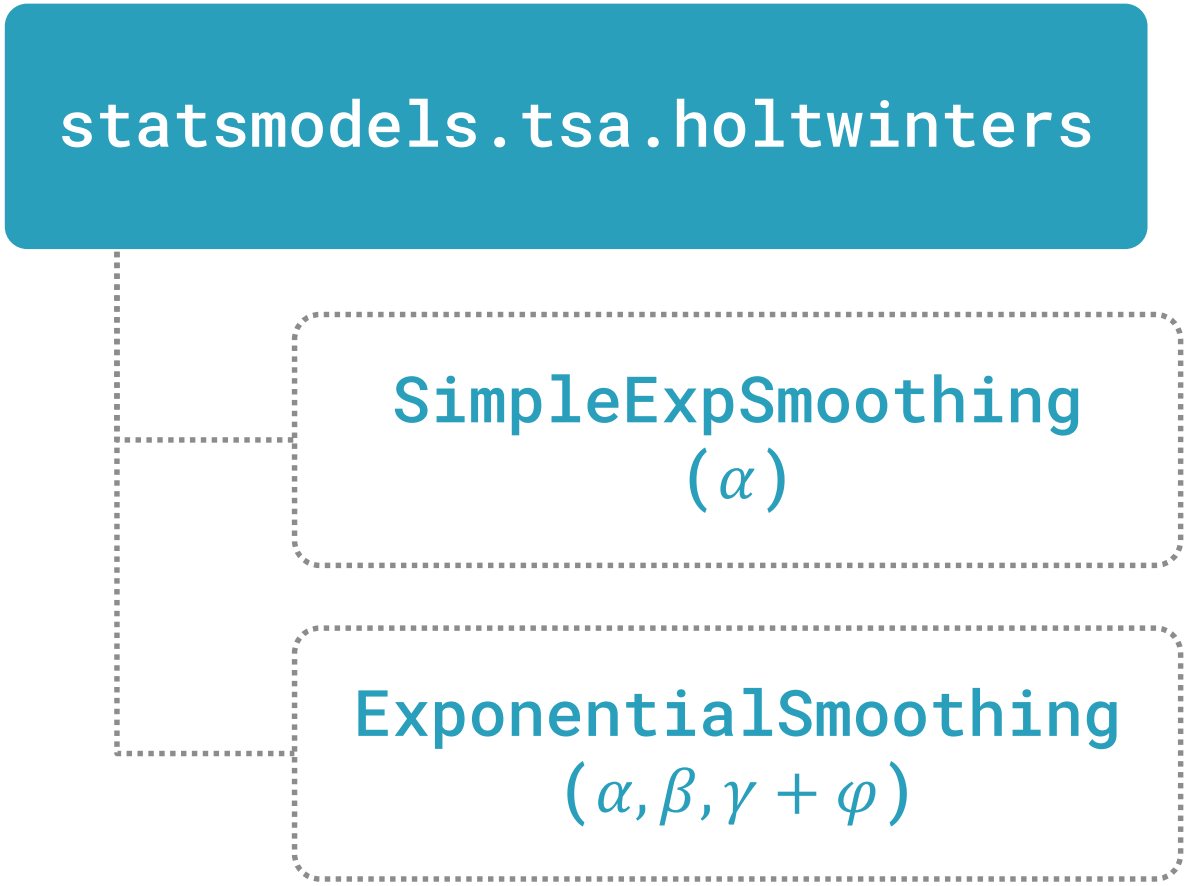
Damping Parameter (φ)

Trend must be present, but it is not constant (e.g. Holt-Winters trend model)

Parameter phi defines the degree of trend damping

- $\varphi \approx 1$: Constant trend model
- $\varphi \approx 0$: Flattened trend curve
- $\mathbf{0.8 < \varphi < 0.98}$

Python Tools for Exponential Smoothing



In [2]:
```
help(ExponentialSmoothing)
```

```
Help on class ExponentialSmoothing in module statsmodels.tsa.holtwinters:

class ExponentialSmoothing(statsmodels.tsa.base.tsa_model.TimeSeriesModel)
 |  Holt Winter's Exponential Smoothing
 |
 |  Parameters
 |  ----------
 |  endog : array-like
 |      Time series
 |  trend : {"add", "mul", "additive", "multiplicative", None}, optional
 |      Type of trend component.
 |  damped : bool, optional
 |      Should the trend component be damped.
 |  seasonal : {"add", "mul", "additive", "multiplicative", None}, optional
 |      Type of seasonal component.
 |  seasonal_periods : int, optional
 |      The number of seasons to consider for the holt winters.
 |
 |  Returns
 |  -------
 |  results : ExponentialSmoothing class
 |
```

In [2]:
```
help(ExponentialSmoothing)
```

```
 |  fit(self, smoothing_level=None, smoothing_slope=None, smoothing_seasonal
=None, damping_slope=None, optimized=True, use_boxcox=False, remove_bias=Fal
se, use_basinhopping=False)
 |      fit Holt Winter's Exponential Smoothing
 |
 |      Parameters
 |      ----------
 |      smoothing_level : float, optional
 |          The alpha value of the simple exponential smoothing, if the valu
e is
 |          set then this value will be used as the value.
 |      smoothing_slope :  float, optional
 |          The beta value of the holts trend method, if the value is
 |          set then this value will be used as the value.
 |      smoothing_seasonal : float, optional
 |          The gamma value of the holt winters seasonal method, if the valu
e is
 |          set then this value will be used as the value.
 |      damping_slope : float, optional
 |          The phi value of the damped method, if the value is
 |          set then this value will be used as the value.
 |      optimized : bool, optional
 |          Should the values that have not been set above be optimized
```

Exponential Smoothing Model Variations

SES

Deducts the present observation by the last few observations

Holt

Initial level (α) and trend (β) with optional damping (φ)

Holt-Winters

A Holt model with seasonality (γ)

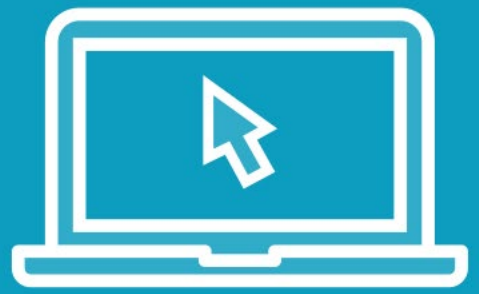
Drawbacks of Exponential Smoothing Models

Sensitivity towards outliers

Cannot handle multiple trends

Changes in the intercept cannot be handled accurately

Demo



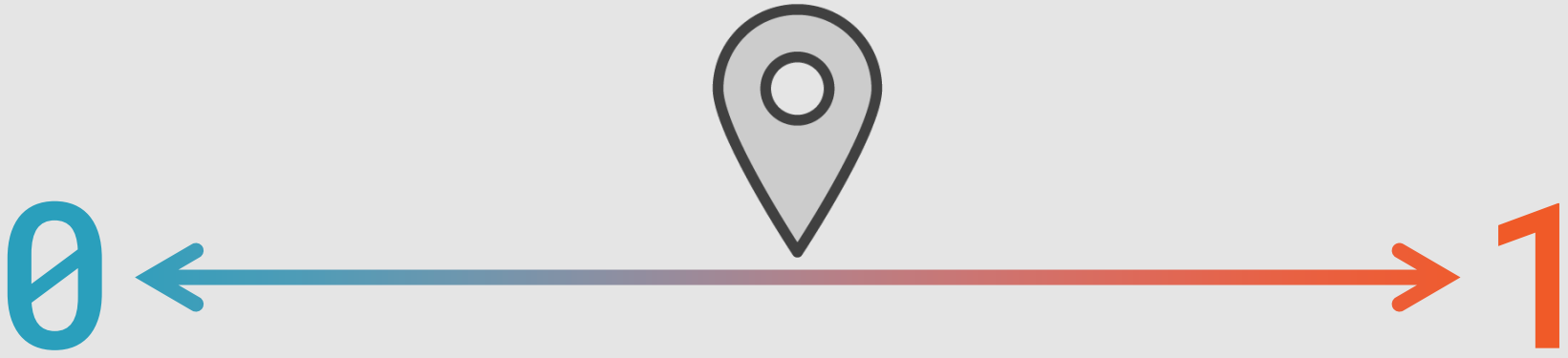
Modeling the nottem seasonal dataset

- No trend and constant seasonality

Model: Holt-Winters(A,N,A)

- Additive level, no trend, additive seasonality

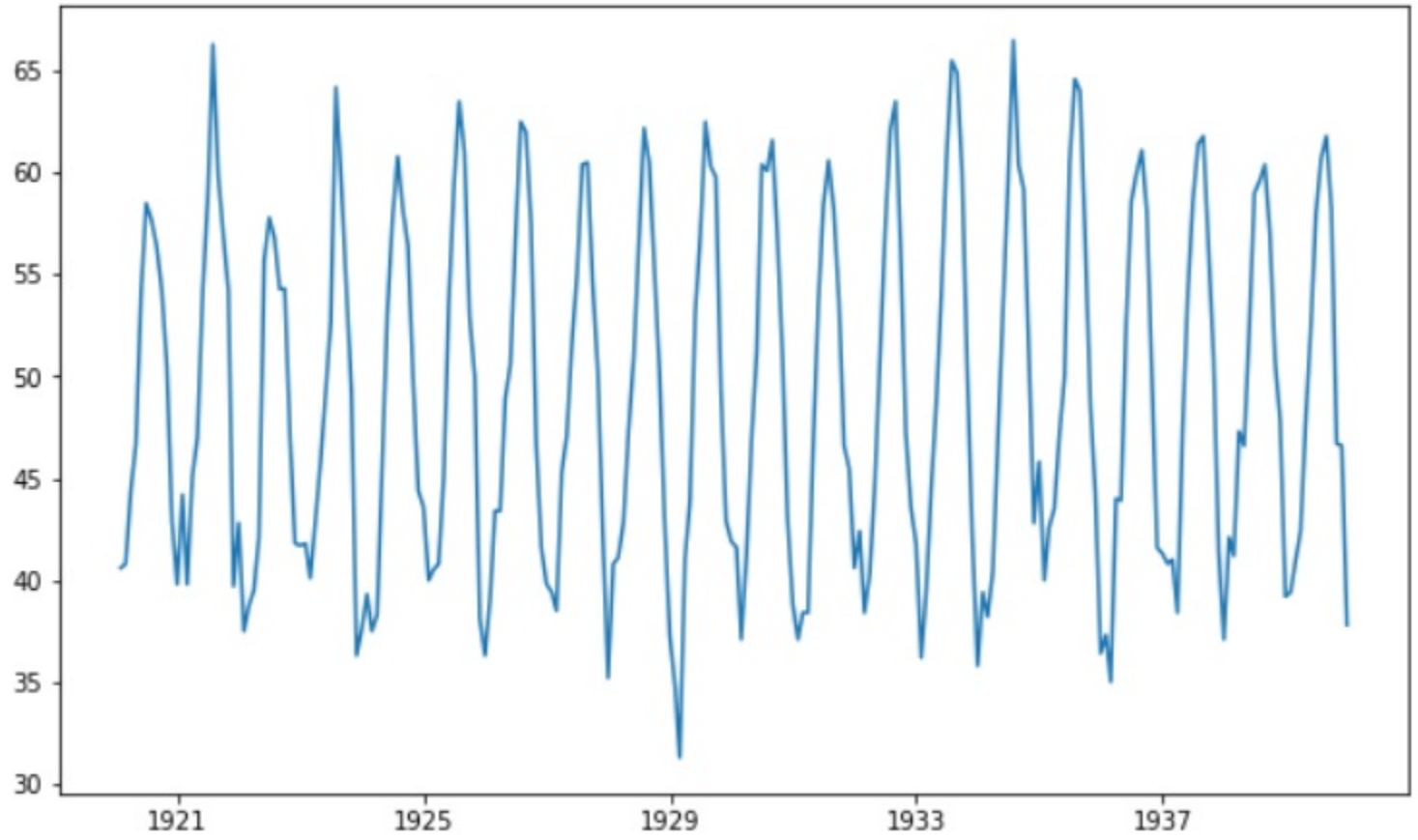
Smoothing Coefficients Determine the Importance of a Time Lag



Smooth model

Reactive model

The Nottem Dataset



```
expsfcast = model.predict(start = 240, end = 251)

# or

expsfcast = model.forecast(steps = 12)
```

Choosing a Forecast Method

StatsModels offers two tools for predictive modeling

- Method ‘predict’
- Method ‘forecast’

Exponential Smoothing Model

Model setup: `ExponentialSmoothing()`

Fitting the model: `model.fit()`

Prediction with the fitted model:
`model.predict(start = , end = )`

Exponential Smoothing

Exponential smoothing implements trend, seasonality and the initial level

Demo on a non-trending seasonal dataset

Smoothing coefficients: alpha, beta and gamma

- Determine the number of lags to be included in the model
- Additional trend dampening parameter phi