

statsmodels.tsa.holtwinters.ExponentialSmoothing

```
class statsmodels.tsa.holtwinters.ExponentialSmoothing(endog, trend=None, damped_trend=False, seasonal=None, *, seasonal_periods=None, initialization_method='estimated',
initial_level=None, initial_trend=None, initial_seasonal=None, use_boxcox=False, bounds=None, dates=None, freq=None, missing='none')[source]
[../modules/statsmodels/tsa/holtwinters/model.html#ExponentialSmoothing]
```

Holt Winter's Exponential Smoothing

Parameters:

endog : [array_like](https://numpy.org/doc/stable/glossary.html#term-array_like) [https://numpy.org/doc/stable/glossary.html#term-array_like]

The time series to model.

trend : {"add", "mul", "additive", "multiplicative", [None](https://docs.python.org/3/library/constants.html#None) [https://docs.python.org/3/library/constants.html#None]}, optional

Type of trend component.

damped_trend : [bool](https://docs.python.org/3/library/stdtypes.html#bool) [https://docs.python.org/3/library/stdtypes.html#bool], optional

Should the trend component be damped.

seasonal : {"add", "mul", "additive", "multiplicative", [None](https://docs.python.org/3/library/constants.html#None) [https://docs.python.org/3/library/constants.html#None]}, optional

Type of seasonal component.

seasonal_periods : [int](https://docs.python.org/3/library/functions.html#int) [https://docs.python.org/3/library/functions.html#int], optional

The number of periods in a complete seasonal cycle, e.g., 4 for quarterly data or 7 for daily data with a weekly cycle.

initialization_method : [str](https://docs.python.org/3/library/stdtypes.html#str) [https://docs.python.org/3/library/stdtypes.html#str], optional

Method for initialize the recursions. One of:

- None
- 'estimated'
- 'heuristic'
- 'legacy-heuristic'
- 'known'

None defaults to the pre-0.12 behavior where initial values are passed as part of `fit`. If any of the other values are passed, then the initial values must also be set when constructing the model. If 'known' initialization is used, then `initial_level` must be passed, as well as `initial_trend` and `initial_seasonal` if applicable. Default is 'estimated'. "legacy-heuristic" uses the same values that were used in statsmodels 0.11 and earlier.

initial_level : [float](https://docs.python.org/3/library/functions.html#float) [https://docs.python.org/3/library/functions.html#float], optional

The initial level component. Required if estimation method is "known". If set using either "estimated" or "heuristic" this value is used. This allows one or more of the initial values to be set while deferring to the heuristic for others or estimating the unset parameters.

initial_trend : [float](https://docs.python.org/3/library/functions.html#float) [https://docs.python.org/3/library/functions.html#float], optional

The initial trend component. Required if estimation method is "known". If set using either "estimated" or "heuristic" this value is used. This allows one or more of the initial values to be set while deferring to the heuristic for others or estimating the unset parameters.

initial_seasonal : [array_like](https://numpy.org/doc/stable/glossary.html#term-array_like) [https://numpy.org/doc/stable/glossary.html#term-array_like], optional

The initial seasonal component. An array of length `seasonal` or length `seasonal - 1` (in which case the last initial value is computed to make the average effect zero). Only used if initialization is 'known'. Required if estimation method is "known". If set using either "estimated" or "heuristic" this value is used. This allows one or more of the initial values to be set while deferring to the heuristic for others or estimating the unset parameters.

use_boxcox : {[True](https://docs.python.org/3/library/constants.html#True) [https://docs.python.org/3/library/constants.html#True], [False](https://docs.python.org/3/library/constants.html#False) [https://docs.python.org/3/library/constants.html#False], 'log', [float](https://docs.python.org/3/library/functions.html#float) [https://docs.python.org/3/library/functions.html#float]}, optional

Should the Box-Cox transform be applied to the data first? If 'log' then apply the log. If float then use the value as lambda.

bounds : [dict](https://docs.python.org/3/library/stdtypes.html#dict) [https://docs.python.org/3/library/stdtypes.html#dict][[str](https://docs.python.org/3/library/stdtypes.html#str) [https://docs.python.org/3/library/stdtypes.html#str], [tuple](https://docs.python.org/3/library/stdtypes.html#tuple) [https://docs.python.org/3/library/stdtypes.html#tuple][[float](https://docs.python.org/3/library/functions.html#float) [https://docs.python.org/3/library/functions.html#float], [float](https://docs.python.org/3/library/functions.html#float) [https://docs.python.org/3/library/functions.html#float]]], optional

A dictionary containing bounds for the parameters in the model, excluding the initial values if estimated. The keys of the dictionary are the variable names, e.g., `smoothing_level` or `initial_slope`. The initial seasonal variables are labeled `initial_seasonal.<j>` for `j=0,...,m-1` where `m` is the number of period in a full season. Use `None` to indicate a non-binding constraint, e.g., `(0, None)` constrains a parameter to be non-negative.

dates : [array_like](https://numpy.org/doc/stable/glossary.html#term-array_like) [https://numpy.org/doc/stable/glossary.html#term-array_like] of [datetime](https://docs.python.org/3/library/datetime.html#module-datetime) [https://docs.python.org/3/library/datetime.html#module-datetime], optional

An array-like object of datetime objects. If a Pandas object is given for endog, it is assumed to have a `DatetimeIndex`.

freq : [str](https://docs.python.org/3/library/stdtypes.html#str) [https://docs.python.org/3/library/stdtypes.html#str], optional

The frequency of the time-series. A Pandas offset or 'B', 'D', 'W', 'M', 'A', or 'Q'. This is optional if dates are given.

missing : `str` [https://docs.python.org/3/library/stdtypes.html#str]

Available options are 'none', 'drop', and 'raise'. If 'none', no nan checking is done. If 'drop', any observations with nans are dropped. If 'raise', an error is raised. Default is 'none'.

Notes

This is a full implementation of the holt winters exponential smoothing as per [1] [#r2ac3c82b3ff9-1]. This includes all the unstable methods as well as the stable methods. The implementation of the library covers the functionality of the R library as much as possible whilst still being Pythonic.

References

[1 [#id1]]

Hyndman, Rob J., and George Athanasopoulos. Forecasting: principles and practice. OTexts, 2014.

Attributes:

endog_names [statsmodels.tsa.holtwinters.ExponentialSmoothing.endog_names.html#statsmodels.tsa.holtwinters.ExponentialSmoothing.endog_names]

Names of endogenous variables.

exog_names [statsmodels.tsa.holtwinters.ExponentialSmoothing.exog_names.html#statsmodels.tsa.holtwinters.ExponentialSmoothing.exog_names]

The names of the exogenous variables.

Methods

fit [statsmodels.tsa.holtwinters.ExponentialSmoothing.fit.html#statsmodels.tsa.holtwinters.ExponentialSmoothing.fit](smoothing_level, smoothing_trend, ...)	Fit the model
fix_params [statsmodels.tsa.holtwinters.ExponentialSmoothing.fix_params.html#statsmodels.tsa.holtwinters.ExponentialSmoothing.fix_params](values)	Temporarily fix parameters for estimation.
from_formula [statsmodels.tsa.holtwinters.ExponentialSmoothing.from_formula.html#statsmodels.tsa.holtwinters.ExponentialSmoothing.from_formula](formula, data[, subset, drop_cols])	Create a Model from a formula and dataframe.
hessian [statsmodels.tsa.holtwinters.ExponentialSmoothing.hessian.html#statsmodels.tsa.holtwinters.ExponentialSmoothing.hessian](params)	The Hessian matrix of the model.
information [statsmodels.tsa.holtwinters.ExponentialSmoothing.information.html#statsmodels.tsa.holtwinters.ExponentialSmoothing.information](params)	Fisher information matrix of model.
initial_values [statsmodels.tsa.holtwinters.ExponentialSmoothing.initial_values.html#statsmodels.tsa.holtwinters.ExponentialSmoothing.initial_values]([initial_level, ...])	Compute initial values used in the exponential smoothing recursions.
initialize [statsmodels.tsa.holtwinters.ExponentialSmoothing.initialize.html#statsmodels.tsa.holtwinters.ExponentialSmoothing.initialize]()	Initialize (possibly re-initialize) a Model instance.
loglike [statsmodels.tsa.holtwinters.ExponentialSmoothing.loglike.html#statsmodels.tsa.holtwinters.ExponentialSmoothing.loglike](params)	Log-likelihood of model.
predict [statsmodels.tsa.holtwinters.ExponentialSmoothing.predict.html#statsmodels.tsa.holtwinters.ExponentialSmoothing.predict](params[, start, end])	In-sample and out-of-sample prediction.
score [statsmodels.tsa.holtwinters.ExponentialSmoothing.score.html#statsmodels.tsa.holtwinters.ExponentialSmoothing.score](params)	Score vector of model.

Properties

endog_names [statsmodels.tsa.holtwinters.ExponentialSmoothing.endog_names.html#statsmodels.tsa.holtwinters.ExponentialSmoothing.endog_names] Names of endogenous variables.

exog_names [statsmodels.tsa.holtwinters.ExponentialSmoothing.exog_names.html#statsmodels.tsa.holtwinters.ExponentialSmoothing.exog_names] The names of the exogenous variables.

