ETS 1.0

Allin Cottrell and Ignacio Díaz-Emparanza

2026-11-07

1 Introduction

'ETS' can be thought of as indicating "Exponential Time-series Smoothing" or "Error, Trend, Seasonal". This package implements exponential smoothing for time series data, in three variants:

1. *Simple* exponential smoothing: a level component only.
2. *Holt's* method (Holt, 1957), also known as double exponential smoothing: level and trend components.
3. The *Holt-Winters* method (Winters, 1960), also known as triple exponential smoothing: level, trend and seasonal components, where the seasonal component can be either additive or multiplicative.

The seasonal variants require data with a seasonal dimension. Throughout this document m denotes the periodicity of the data, that is, the number of observations per seasonal cycle (gretl's `$pd`): $m = 4$ for quarterly data and $m = 12$ for monthly data, which are the best bet for suitability. Daily or hourly data might be suitable, subject to the qualifications noted below. Weekly data ($m = 52$) are not supported: the seasonal period is not an integer number of weeks.

The package combines and supersedes the earlier Holt and Winters packages, and adds simple exponential smoothing for completeness. See section 8 for the history of the package and the differences from those packages.

Daily and hourly data

The seasonal models may be applicable to daily data, with $m = 5$, 6 or 7 days per week, where 'seasonality' means intra-week variability. However, the results will be meaningful only if the data are complete, in the sense that each week in the sample range includes m valid daily observations. Gretl also supports hourly data ($m = 24$) and the seasonal models may work if there's seasonality in the form of intra-day variability. Again, however, the series in question has to be complete over the given sample range.

2 The models

The filtering recursions and forecasting equations are shown below. The notation is based on Hyndman and Athanasopoulos (2021, chapter 8): ℓ denotes the level, b the trend or slope, and s the seasonal; α, β and γ are the respective smoothing parameters. In addition m is the periodicity of the data, as defined in section 1, and k is the integer part of $(h-1)/m$ (= 0 when $h = 1$).

Simple:

$$\begin{aligned} \ell_t &= \alpha y_t + (1-\alpha)\ell_{t-1} \\ \hat{y}_{t+h|t} &= \ell_t \end{aligned}$$

Holt:

$$\begin{aligned} \ell_t &= \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\ b_t &= \beta(\ell_t - \ell_{t-1}) + (1-\beta)b_{t-1} \\ \hat{y}_{t+h|t} &= \ell_t + hb_t \end{aligned}$$

Holt-Winters, additive:

$$\begin{aligned}
\ell_t &= \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\
b_t &= \beta(\ell_t - \ell_{t-1}) + (1-\beta)b_{t-1} \\
s_t &= \gamma(y_t - \ell_t) + (1-\gamma)s_{t-m} \\
\hat{y}_{t+h|t} &= \ell_t + hb_t + s_{t+h-m(k+1)}
\end{aligned}$$

Holt-Winters, multiplicative:

$$\begin{aligned}
\ell_t &= \alpha(y_t / s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\
b_t &= \beta(\ell_t - \ell_{t-1}) + (1-\beta)b_{t-1} \\
s_t &= \gamma\, y_t/\ell_t + (1-\gamma)s_{t-m} \\
\hat{y}_{t+h|t} &= (\ell_t + hb_t)\, s_{t+h-m(k+1)}
\end{aligned}$$

3 The public functions

The primary function has the following signature:

```
function bundle ETS (const series y, const bundle specs[null])
```

The required argument `y` specifies the series to be processed. The second argument—which can be omitted, as indicated by the `null` modifier—may contain one or more of the following options:

- `trend` (integer): 0 or 1, to exclude or include a trend component. The default is 0, unless `seasonal` is non-zero, in which case it is 1.
- `seasonal` (integer): 0 for no seasonal component (the default), 1 for an additive seasonal component or 2 for a multiplicative one. A seasonal component requires a trend: setting `seasonal` non-zero along with `trend` = 0 is an error.
- `theta` (matrix): if supplied, this must be a vector holding the smoothing parameters, each in $[0, 1]$. The length depends on the model: 1 for simple smoothing, $\theta = (\alpha)$; 2 for Holt, $\theta = (\alpha, \beta)$; and 3 for Holt-Winters, $\theta = (\alpha, \beta, \gamma)$. If `theta` is not given the parameters are determined automatically.
- `init` (matrix): if supplied, this must be a vector holding the initial state: the level, followed by the trend if applicable, followed by the m seasonals (oldest first) if applicable, where m is the periodicity of the data. The length is thus 1 for simple smoothing, 2 for Holt and $m + 2$ for Holt-Winters. In the multiplicative case the level and seasonals must be positive. If `init` is given it is held fixed: it is neither estimated nor overwritten, and if `theta` is not given only the smoothing parameters are optimized. This is mainly useful for testing and for replicating results from other software.
- `ntest` (integer): specifies the number of observations at the end of the range of `y` values to reserve for testing via pseudo-out-of-sample forecasting. The default value is 0.
- `verbose` (integer): controls the verbosity of output, with levels 0, 1 (the default), 2 or 3.

Thus the model is selected by the combination of `trend` and `seasonal`:

trend	*seasonal*	*model*
0	0	simple exponential smoothing
1	0	Holt
1	1	Holt-Winters, additive
1	2	Holt-Winters, multiplicative

The return value from ETS is a bundle containing the following elements:

- theta: a vector holding the smoothing parameter values.
- init: a vector holding the initial state: the level, followed by the trend if applicable, followed by m seasonals if applicable.
- trend, seasonal: the model flags as described above.
- ntest: a non-negative integer indicating how many observations were reserved for pseudo-out-of-sample forecasting.
- S: a matrix with a row for each observation in the range from t_1 to the last forecast. Column 1 holds the data passed as y; the next columns hold the estimated level, then the trend (if applicable), then the seasonal (if applicable); and the final column holds forecasts. Columns are named. The matrix is basically a matrix version of the table of results displayed when verbose is 1 or greater (see section 5).
- RMSE_train, MAE_train: respectively, the root mean squared error and mean absolute error for the 1-step ahead forecasts made using the training data.
- yname: the name of the input series.
- limits: a 3-vector holding the 1-based indices of the first training observation, the last training observation, and the last forecast. If there are no post-training forecasts (testing or out-of-sample) the third element is equal to the second.
- nparms: the number of estimated parameters (smoothing parameters plus free elements of the initial state), used in calculating degrees of freedom for confidence intervals.

If ntest is non-zero, two additional items are included in the bundle, namely the RMSE and MAE for the pseudo-out-of-sample forecasts, under the keys RMSE_test and MAE_test.

If any post-training forecasts are performed (either in response to ntest or out of sample) and the model is not multiplicative, the matrix S acquires a further column, holding standard errors for these forecasts. (In the multiplicative case there is no closed form for the forecast variance; see section 7.)

The standard-errors column of S, when present, is named stderr. If you want to grab the columns of S as a list of series you can do something like this:

```
bundle sb = ETS(Exports)
list L = mat2list(sb.S)
list L print # show all the series in L
```

If gretl is version 2026c or later the bundle also contains "virtual series" named after the input series: for a series y, y_level, y_fcast and y_irreg, plus y_trend and y_season where applicable.

Table 1 shows what is printed by ETS as a function of the verbose option.

A simple *plotting* function is also supplied:

```
function void ETS_plot (bundle *b, int ptype, matrix ci_levels[null])
```

The first argument should be a pointer to a bundle obtained from ETS and the second argument should be either 1 for a plot of actual and forecast values, or 2 for a plot of the components: level; trend and seasonal if applicable; and irregular. The third (optional) argument is referenced only when ptype = 1 and post-training forecasts have been performed, in which case confidence

Table 1: Effect of the verbose option

verbose	*cumulative effect*
0	Nothing is printed.
1	Parameters and initializer are reported, also a table of results (see section 5) and the SSE, RMSE and MAE for the training data.
2	If optimization is performed, partial information on progress of the iterative procedure is shown.
3	If optimization is performed, progress of the optimizer is shown in full.

intervals will be shown. If `ci_levels` is supplied it must be a 2-vector holding two distinct confidence levels, in percentage form and in the range 50 to 99. If this argument is not given the levels default to 80% and 95%. For the additive models the intervals are based on analytical standard errors; in the multiplicative case they are simulation-based. See section 7 for details.

A third public function provides confidence intervals for the post-training forecasts (it is an error to call it if there are none):

```
function matrix ETS_ci_matrix (const bundle b, matrix ci_levels[null])
```

The argument must be a bundle obtained from `ETS`. The return value is a matrix with four columns, and rows equal to the number of post-training forecasts. By default the first two columns hold the lower and upper limits for an 80 percent *confidence interval* and the second two hold the limits for a 95 percent interval. The optional `ci_levels` argument works just as with `ETS_plot`, and the intervals are calculated as described above.

4 Sample ranges and forecasts

Like most functions in gretl, `ETS` confines its operations to the sample range in force when the function is called. We'll refer to the (1-based) start and end of this range as t_1 and t_2, respectively.[1] In addition, let T denote the date of the last valid observation of the variable to be predicted, y.

There are two cases to be considered:

1. $T < t_2$: in this case there's a 'tail' of observations for which values of y_t are not available. `ETS` will by default produce predictions for these observations.
2. $T = t_2$: there's no tail for genuine out-of-sample prediction, but you can specify a number of observations, `ntest`, to reserve for pseudo-out-of-sample prediction. In that case the training recursion will be terminated at observation $T -$ `ntest`.

If you want case 1 but your dataset doesn't contain any extra trailing observations it's straightforward to add them using the `dataset addobs` command, or via the Add observations item under the Data menu in the graphical interface.

Conversely, if the dataset does contain extra trailing observations but you want case 2, specifying a positive value for `ntest` will result in the extra observations being ignored and pseudo-out-of-sample forecasts being produced.

[1] This range can be set via the `smpl` command. If `smpl` has not been invoked since a dataset was opened t_1 will equal 1 and t_2 will equal the total number of observations.

5 Examples

Script examples

We'll illustrate with replication of three examples from Hyndman and Athanasopoulos (2021). First, simple exponential smoothing applied to Algerian exports (section 8.1):

```
include ETS.gfn
open alg_exports.gdt --frompkg=ETS
dataset addobs 5
bundle sb = ETS(Exports)
ETS_plot(&sb, 1)
```

The `alg_exports` dataset contains 58 annual observations from 1960 to 2017, which are used for training the model. The five added observations, 2018 to 2022, are then used for out-of-sample forecasting. Since `theta` is not specified, both α and the initial level will be optimized automatically; see section 7 for details. Second, Holt's method applied to Australian population (section 8.2):

```
open auspop.gdt --frompkg=ETS
dataset addobs 10
bundle hb = ETS(pop, _(trend=1))
ETS_plot(&hb, 1)
ETS_plot(&hb, 2)
```

And third, the Holt-Winters method (multiplicative variant) applied to Australian tourism (section 8.3):

```
open aus_holidays.gdt --frompkg=ETS
dataset addobs 12
bundle wb = ETS(Trips, _(seasonal=2))
ETS_plot(&wb, 1)
ETS_plot(&wb, 2)
```

The `aus_holidays` dataset contains 80 quarterly observations from 1998:1 to 2017:4; the twelve added observations are for out-of-sample forecasting.

One might choose, instead, to use the `ntest` option to get `ETS` to generate (say) eight pseudo-out-of-sample forecasts:

```
open aus_holidays.gdt --frompkg=ETS
bundle wb = ETS(Trips, _(seasonal=2, ntest=8))
```

This generates forecasts for 2016:1 to 2017:4, and the RMSE and MAE for these testing forecasts are printed.

The data files referenced in these scripts, along with additional example scripts, can be found in the `examples` subdirectory of the `ETS` installation. You can access these resources under gretl's Packages menu: select On local machine, find `ETS`, right-click and select Resources.

Printed output

Examples of the table of results displayed when `verbose` is 1 or greater are presented in listings 1 and 2. See section 3 for the effect of the `verbose` option and for the relationship between this table and the matrix `S` in the returned bundle.

Listing 1: Partial listing of results from the simple exponential smoothing example in Section 5.

```
Simple exponential smoothing
Training range: 1960 to 2017
Out-of-sample forecast: 2018 to 2022

estimated parameter values:
  α = 0.839783

estimated initializer:
  level = 39.5381

training SSE = 1995.29, RMSE = 5.86528, MAE = 4.00111

                y        level     forecast

1960     39.04317     39.12248     39.53815
1961     46.24456     45.10348     39.12248
1962     19.79387     23.84890     45.10348
1963     24.68468     24.55078     23.84890
...
2014     30.21912     30.80023     33.84617
2015     23.17178     24.39399     30.80023
2016     20.86001     21.42621     24.39399
2017     22.63889     22.44460     21.42621
2018                              22.44460
2019                              22.44460
2020                              22.44460
2021                              22.44460
2022                              22.44460
```

Listing 2: Partial listing of results from the Holt-Winters example in Section 5: forecasts for 2018:1 and beyond are out-of-sample.

```
Holt-Winters, using multiplicative formula
Training range: 1998:1 to 2017:4
Out-of-sample forecast: 2018:1 to 2020:4

estimated parameter values:
  α = 0.246678, β = 0.122928, γ = 0

estimated initializer:
  level = 9.9096, trend = -0.0285393
  seasonals = {1.16264, 0.966858, 0.926865, 0.943636}

training SSE = 13.4951, RMSE = 0.410718, MAE = 0.307344

                  y        level        trend     seasonal     forecast

1998:1     11.80604      9.94851     -0.02025     1.162641     11.48812
1998:2      9.27566      9.84571     -0.03040     0.966858      9.59922
1998:3      8.64249      9.69423     -0.04528     0.926865      9.09748
1998:4      9.29952      9.69977     -0.03903     0.943636      9.10509
...
2017:1     12.40642     10.79711      0.08755     1.162641     12.60121
2017:2     10.47120     10.87121      0.08590     0.966858     10.52392
2017:3     10.49917     11.04850      0.09713     0.926865     10.15576
2017:4     11.21082     11.32689    0.1194143     0.943636     10.51742
2018:1                                                        13.30794
2018:2                                                        11.18241
2018:3                                                        10.83055
2018:4                                                        11.13920
...
```

6 GUI usage

When you install this package you should get the option of attaching it as the menu item Variable/Filter/ETS in gretl's main window. Clicking this item brings up a dialog box in which you choose the model (simple, Holt, or Holt-Winters additive or multiplicative), whether the smoothing parameters are to be estimated or pre-specified, the number of test observations and the verbosity level. See Figure 1.

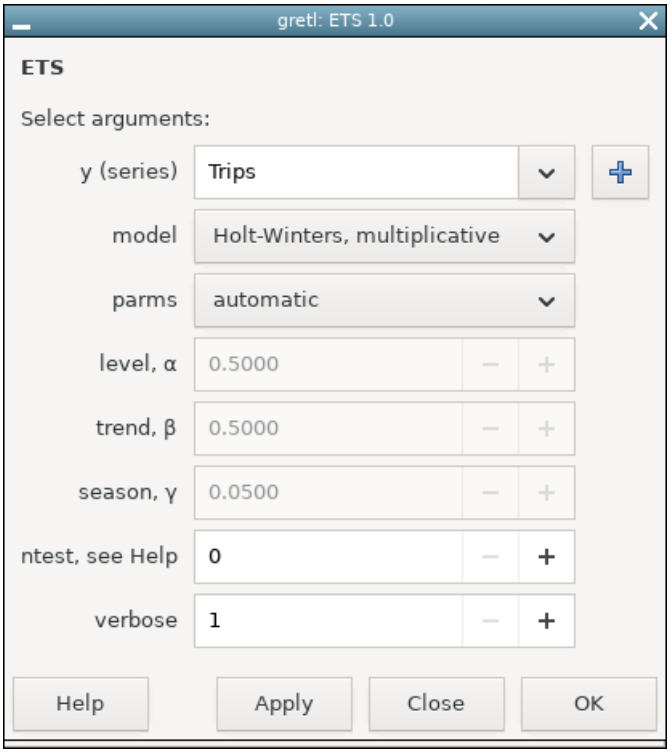


Figure 1: ETS dialog, under the Variable/Filter menu

Clicking the Help button will bring up a brief explanation of each control. Only those smoothing parameters relevant to the selected model are used when they are pre-specified: α for simple smoothing, α and β for Holt, all three for Holt-Winters. On clicking OK you should get a window showing the results. See Figure 2. The toolbar at the top of this window includes a Graph button which gives access to two types of plot, one showing actual and forecast y and one showing the components.

7 Technical details

This section offers some details on the initialization with which the recursions are started, the optimization that is carried out if `theta` is not given by the caller, and the calculation of forecast standard errors.

Initialization

This subsection describes the default ("rule-of-thumb") initial state. It is used as is only when it is not optimized or supplied by the caller—that is, for the non-seasonal models when `theta` is given but `init` is not, and, in the multiplicative seasonal case, as the starting point for optimization of the initializer (see below). In the additive models, when the initializer is optimized it is solved for directly and these values play no role.

Let τ denote the first period in which y is observed. For the non-seasonal models the basic initialization is as follows.

- The initial level, $\ell_{\tau-1}$, is set to the first observed value y_τ.
- In the Holt case the initial trend, $b_{\tau-1}$, is set to the first observed difference, $y_{\tau+1} - y_\tau$.

For the seasonal models we follow the rule-of-thumb suggested by Makridakis *et al.* (1998), which is also described in Hyndman (2010).

- The initial level, $\ell_{\tau-1}$, is set to the mean of the m values $y_\tau, y_{\tau+1}, \ldots, y_{\tau+m-1}$.
- The initial trend, $b_{\tau-1}$, is set to the mean of the first m seasonal differences:

$$m^{-1} \sum_{i=0}^{m-1} (y_{\tau+i+m} - y_{\tau+i})$$

- The initial seasonals are then set to the ratio of period-specific y_t to $\ell_{\tau-1}$ for the first m observations (multiplicative case) or the difference between y_t and $\ell_{\tau-1}$ (additive case).

Optimization

If neither `theta` nor `init` is pre-specified, the parameter values and initializer are optimized, with the criterion of minimizing the sum of squared one-step ahead forecast errors for the training range. The starting point for the parameters is determined via a coarse grid search, then optimization is performed using the L-BFGS-B algorithm (see the chapter on Numerical methods in the the *Gretl User's Guide* for details), with the smoothing parameters constrained to lie in $[0, 1]$.

In the additive models (simple, Holt and additive Holt-Winters) the recursion is linear in the state, so given the smoothing parameters the one-step forecasts are linear in the initial state. The initializer can therefore be concentrated out and solved for by OLS within each evaluation of the criterion, and L-BFGS-B need only search over the smoothing parameters.[2] In the seasonal case the seasonals are constrained to sum to zero, since the initial level and the seasonal mean are not separately identified. In the multiplicative case the initializer is optimized jointly with the smoothing parameters via L-BFGS-B, starting from the rule-of-thumb values, with the seasonals normalized to average 1.

If `theta` is given but `init` is not given, the treatment of the initializer depends on the model. For the seasonal models, it is optimized conditional on θ, using the same criterion: in the additive case this is a linear least-squares problem with a closed-form solution, while in the multiplicative case L-BFGS-B is used. For the non-seasonal models the initializer is just the basic value described above; no optimization is performed.

If `init` is given (with or without `theta`) it is used as is, and only the smoothing parameters, if not given, are optimized (by L-BFGS-B, for all four models). The supplied seasonals are not renormalized, so the `init` in the returned bundle is exactly what was passed in.

The various cases are summarized in Table 2.

[2] Thanks go to Jack Lucchetti for suggesting a particularly efficient approach to the optimization, making use of gretl's `varsimul` function.

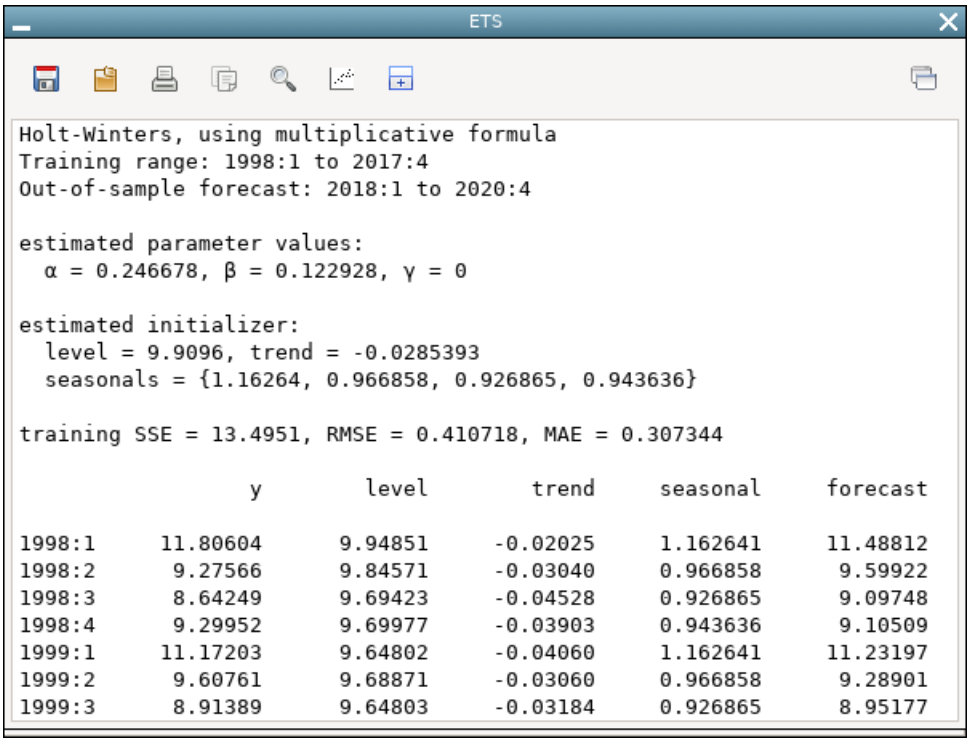


Figure 2: ETS output window

Table 2: Treatment of the initializer and smoothing parameters

theta given	*init given*	*simple and Holt*	*Holt-Winters, additive*	*Holt-Winters, multiplicative*
No	No	θ: L-BFGS-B; init: OLS	θ: L-BFGS-B; init: OLS	θ and init jointly: L-BFGS-B
Yes	No	init: basic	init: OLS	init: L-BFGS-B
No	Yes	θ: L-BFGS-B	θ: L-BFGS-B	θ: L-BFGS-B
Yes	Yes	no estimation	no estimation	no estimation

Forecast standard errors

For the additive models, the h-step ahead forecast error variance follows Hyndman and Athanasopoulos (2021, section 8.7):

$$\sigma_h^2 = \sigma^2 \left[1 + \sum_{j=1}^{h-1} c_j^2 \right], \quad c_j = \alpha + \beta^* j + \gamma^* I(j \bmod m = 0),$$

where σ^2 is estimated by the MSE for the training data and $\beta^* = \alpha\beta$ and $\gamma^* = (1-\alpha)\gamma$ are the trend and seasonal parameters of the equivalent ETS(A,·,·) model in its error-correction form (see Hyndman and Athanasopoulos 2021, sections 8.3 and 8.5, where the asterisks are used the other way round, to mark the component form used in our recursions). Terms that do not apply to the model in question (β^* in the simple case, γ^* in the non-seasonal cases) are zero. These expressions reduce to the familiar closed forms: for simple smoothing, $\sigma^2[1 + (h-1)\alpha^2]$, and for Holt,

$$\sigma_h^2 = \sigma^2 \left[1 + (h-1)\{\alpha^2 + \alpha\beta^* h + \frac{1}{6}\beta^{*2} h(2h-1)\} \right].$$

The intervals use quantiles of the $t(\nu)$ distribution, with ν equal to the number of training observations minus the number of estimated parameters (smoothing parameters plus free initial-state elements; this is the `nparms` element of the returned bundle).

In the multiplicative case there is no closed form for the forecast variance. The intervals are instead based on 10000 simulated paths. Each path starts from the final training state, with the estimated parameter values taken as given, and runs the recursions forward over the forecast range, with the simulated y_t equal to the point forecast plus an error drawn as the training RMSE times a $t(\nu)$ variate. The limits of the intervals at each horizon are the corresponding quantiles of the simulated values of y.

8 History and migration

The earlier versions of Holt (1.0, 2015) and Winters (1.1, 2022) were written by Ignacio Díaz-Emparanza; the versions 2.n on which this package builds were written by Allin Cottrell. Most of the work on the consolidated ETS package was done by Claude Sonnet 5.5.

Migrating from Holt and Winters

ETS replaces the separate Holt (2.0) and Winters (2.3) packages. For users of those packages the principal changes are as follows.

- There is a single primary function, `ETS()`. The call `Holt(y, specs)` becomes `ETS(y, _(trend=1, …))` while `Winters(y, _(additive=1, …))` becomes `ETS(y, _(seasonal=1, …))` and `Winters(y, _(additive=0, …))` becomes `ETS(y, _(seasonal=2, …))`. Note that `Winters` defaulted to the multiplicative variant; `ETS` requires you to say which seasonal variant you want.
- The plotting function is now `ETS_plot`. Confidence intervals are available via `ETS_ci_matrix`, which is new for the seasonal models and covers both the additive and multiplicative cases.
- The results matrix in the returned bundle is named `S`, in place of `H` (Holt) or `W` (Winters). Its columns are as before except that columns which do not apply to the model are omitted and the columns are named. In particular, the forecast is the last column other than the standard errors (column 4 for Holt, 5 for Holt-Winters); the virtual series keep the same suffixes as before.
- The bundle also contains the model flags `trend` and `seasonal` (replacing `additive`) and `nparms`.

- For the same model, parameters and options, the numerical results are the same as from the earlier packages (up to the tolerance of the optimizer). The forecast standard errors for the additive models are also identical.
- The `ETS_gui` dialog offers a single model selector in place of the formula selector in the Winters dialog.

Change log

- `v1.0`, 2026-11-07: Initial release.

References

Holt, C. C. (1957) 'Forecasting seasonals and trends by exponentially weighted moving averages'. ONR Memorandum, Vol. 52, Carnegie Institute of Technology. URL https://doi.org/10.1016/j.ijforecast.2003.09.015.

Hyndman, R. (2010) 'Initializing the Holt–Winters method'. https://robjhyndman.com/hyndsight/hw-initialization/. Accessed: 2026-03-09.

Hyndman, R. J. and G. Athanasopoulos (2021) 'Forecasting: principles and practice, 3e'. URL https://otexts.com/fpp3/. Accessed: 2026-09-14.

Makridakis, S., S. C. Wheelwright and R. Hyndman (1998) *Forecasting: Methods and Applications*, New York: Wiley.

Winters, P. R. (1960) 'Forecasting sales by exponentially weighted moving averages', *Management Science* 6(3): 324–342. URL https://doi.org/10.1287/mnsc.6.3.324.