Title

stata.com

tssmooth exponential — Single-exponential smoothing

Description

`tssmooth exponential` models the trend of a variable whose change from the previous value is serially correlated. More precisely, it models a variable whose first difference follows a low-order, moving-average process.

Quick start

Create `smooth` using a single-exponential smoother over `y` with `tsset` data

```
tssmooth exponential smooth=y
```

Same as above, but forecast 10 periods out of sample

```
tssmooth exponential smooth=y, forecast(10)
```

Same as above, but use 111 as the initial value for the recursion

```
tssmooth exponential smooth=y, forecast(10) s0(111)
```

Same as above, but use 0.5 as the smoothing parameter

```
tssmooth exponential smooth=y, forecast(10) s0(111) parms(.5)
```

Note: The above commands can also be used to apply the smoother separately to each panel of a panel dataset when a *panelvar* has been specified using `tsset` or `xtset`.

Menu

Statistics > Time series > Smoothers/univariate forecasters > Single-exponential smoothing

Syntax

```
tssmooth exponential [type] newvar = exp [if] [in] [, options]
```

options	Description
Main	
`replace`	replace *newvar* if it already exists
`parms(#`$_\alpha$`)`	use #$_\alpha$ as smoothing parameter
`samp0(#)`	use # observations to obtain initial value for recursion
`s0(#)`	use # as initial value for recursion
`forecast(#)`	use # periods for the out-of-sample forecast

You must `tsset` your data before using `tssmooth exponential`; see [TS] **tsset**.
exp may contain time-series operators; see [U] **11.4.4 Time-series varlists**.
`collect` is allowed; see [U] **11.1.10 Prefix commands**.

Options

Main

`replace` replaces *newvar* if it already exists.

`parms(#`$_\alpha$`)` specifies the parameter α for the exponential smoother; $0 < \#_\alpha < 1$. If `parms(#`$_\alpha$`)` is not specified, the smoothing parameter is chosen to minimize the in-sample sum-of-squared forecast errors.

`samp0(#)` and `s0(#)` are mutually exclusive ways of specifying the initial value for the recursion.

`samp0(#)` specifies that the initial value be obtained by calculating the mean over the first # observations of the sample.

`s0(#)` specifies the initial value to be used.

If neither option is specified, the default is to use the mean calculated over the first half of the sample.

`forecast(#)` gives the number of observations for the out-of-sample prediction; $0 \leq \# \leq 500$. The default is `forecast(0)` and is equivalent to not forecasting out of sample.

Remarks and examples

stata.com

Introduction

Exponential smoothing can be viewed either as an adaptive-forecasting algorithm or, equivalently, as a geometrically weighted moving-average filter. Exponential smoothing is most appropriate when used with time-series data that exhibit no linear or higher-order trends but that do exhibit low-velocity, aperiodic variation in the mean. Abraham and Ledolter (1983), Bowerman, O'Connell, and Koehler (2005), and Montgomery, Johnson, and Gardiner (1990) all provide good introductions to single-exponential smoothing. Chatfield (2001, 2004) discusses how single-exponential smoothing

relates to modern time-series methods. For example, simple exponential smoothing produces optimal forecasts for several underlying models, including ARIMA(0,1,1) and the random-walk-plus-noise state-space model. (See Chatfield [2001, sec. 4.3.1].)

The exponential filter with smoothing parameter α creates the series S_t, where

$$S_t = \alpha X_t + (1-\alpha)S_{t-1} \qquad \text{for } t = 1, \ldots, T$$

and S_0 is the initial value. This is the adaptive forecast-updating form of the exponential smoother. This implies that

$$S_t = \alpha \sum_{k=0}^{T-1} (1-\alpha)^K X_{T-k} + (1-\alpha)^T S_0$$

which is the weighted moving-average representation, with geometrically declining weights. The choice of the smoothing constant α determines how quickly the smoothed series or forecast will adjust to changes in the mean of the unfiltered series. For small values of α, the response will be slow because more weight is placed on the previous estimate of the mean of the unfiltered series, whereas larger values of α will put more emphasis on the most recently observed value of the unfiltered series.

Examples

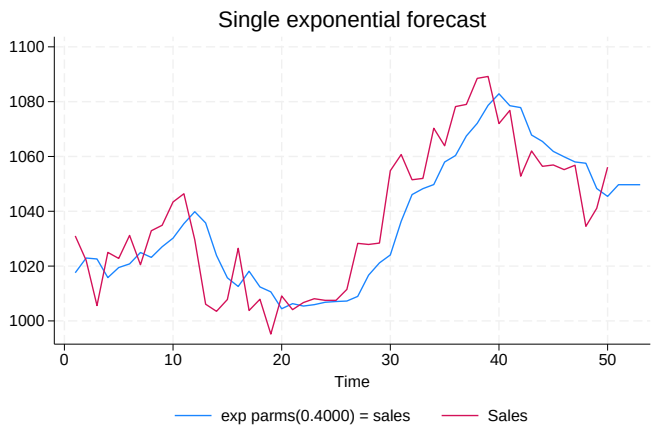
Example 1: Smoothing a series for specified parameters

Let's consider some examples using sales data. Here we forecast sales for three periods with a smoothing parameter of 0.4:

```
. use https://www.stata-press.com/data/r18/sales1
. tssmooth exponential sm1=sales, parms(.4) forecast(3)
exponential coefficient  =       0.4000
sum-of-squared residuals =         8345
root mean squared error  =       12.919
```

To compare our forecast with the actual data, we graph the series and the forecasted series over time.

```
. tsline sm1 sales, title("Single exponential forecast")
```

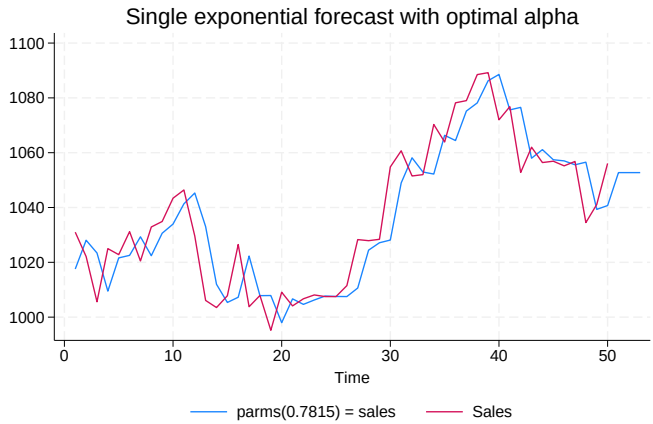


The graph indicates that our forecasted series may not be adjusting rapidly enough to the changes in the actual series. The smoothing parameter α controls the rate at which the forecast adjusts. Smaller values of α adjust the forecasts more slowly. Thus we suspect that our chosen value of 0.4 is too small. One way to investigate this suspicion is to ask `tssmooth exponential` to choose the smoothing parameter that minimizes the sum-of-squared forecast errors.

```
. tssmooth exponential sm2=sales, forecast(3)
computing optimal exponential  coefficient (0,1)
optimal exponential coefficient =        0.7815
sum-of-squared residuals        =     6727.7056
root mean squared error         =     11.599746
```

The output suggests that the value of $\alpha = 0.4$ is too small. The graph below indicates that the new forecast tracks the series much more closely than the previous forecast.

```
. tsline sm2 sales, title("Single exponential forecast with optimal alpha")
```



We noted above that simple exponential forecasts are optimal for an ARIMA (0,1,1) model. (See [TS] **arima** for fitting ARIMA models in Stata.) Chatfield (2001, 90) gives the following useful derivation that relates the MA coefficient in an ARIMA (0,1,1) model to the smoothing parameter in single-exponential smoothing. An ARIMA (0,1,1) is given by

$$x_t - x_{t-1} = \epsilon_t + \theta\epsilon_{t-1}$$

where ϵ_t is an independent and identically distributed white-noise error term. Thus given $\widehat{\theta}$, an estimate of θ, an optimal one-step prediction of $\widehat{x}_{t+1}$ is $\widehat{x}_{t+1} = x_t + \widehat{\theta}\epsilon_t$. Because ϵ_t is not observable, it can be replaced by

$$\widehat{\epsilon_t} = x_t - \widehat{x}_{t-1}$$

yielding

$$\widehat{x}_{t+1} = x_t + \widehat{\theta}(x_t - \widehat{x}_{t-1})$$

Letting $\widehat{\alpha} = 1 + \widehat{\theta}$ and doing more rearranging implies that

$$\widehat{x}_{t+1} = (1 + \widehat{\theta})x_t - \widehat{\theta}\widehat{x}_{t-1}$$
$$\widehat{x}_{t+1} = \widehat{\alpha}x_t - (1 - \widehat{\alpha})\widehat{x}_{t-1}$$

Example 2: Comparing ARIMA to exponential smoothing

Let's compare the estimate of the optimal smoothing parameter of 0.7815 with the one we could obtain using [TS] **arima**. Below we fit an ARIMA(0,1,1) to the sales data and then remove the estimate of α. The two estimates of α are quite close, given the large estimated standard error of $\widehat{\theta}$.

```
. arima sales, arima(0,1,1)
(setting optimization to BHHH)
Iteration 0:  Log likelihood = -189.91037
Iteration 1:  Log likelihood = -189.62405
Iteration 2:  Log likelihood = -189.60468
Iteration 3:  Log likelihood = -189.60352
Iteration 4:  Log likelihood = -189.60343
(switching optimization to BFGS)
Iteration 5:  Log likelihood = -189.60342
ARIMA regression
Sample: 2 thru 50                               Number of obs     =         49
                                                Wald chi2(1)      =       1.41
Log likelihood = -189.6034                      Prob > chi2       =     0.2347
```

D.sales	Coefficient	OPG std. err.	z	P>|z|	[95% conf.	interval]
sales						
_cons	.5025469	1.382727	0.36	0.716	-2.207548	3.212641
ARMA						
ma						
L1.	-.1986561	.1671699	-1.19	0.235	-.5263031	.1289908
/sigma	11.58992	1.240607	9.34	0.000	9.158378	14.02147

```
Note: The test of the variance against zero is one sided, and the two-sided
      confidence interval is truncated at zero.
. di 1 + _b[ARMA:L.ma]
.80134387
```

Example 3: Handling panel data

`tssmooth exponential` automatically detects panel data. Suppose that we had sales figures for five companies in long form. Running `tssmooth exponential` on the variable that contains all five series puts the smoothed series and the predictions in one variable in long form. When the smoothing parameter is chosen to minimize the squared prediction error, an optimal value for the smoothing parameter is chosen separately for each panel.

```
. use https://www.stata-press.com/data/r18/sales_cert, clear
. tsset
Panel variable: id (strongly balanced)
 Time variable: t, 1 to 100
         Delta: 1 unit
. tssmooth exponential sm5=sales, forecast(3)
-------------------------------------------------------------------------------
-> id = 1
computing optimal exponential  coefficient (0,1)
optimal exponential coefficient =        0.8702
sum-of-squared residuals        =     16070.567
root mean squared error         =     12.676974
-------------------------------------------------------------------------------
-> id = 2
computing optimal exponential  coefficient (0,1)
optimal exponential coefficient =        0.7003
sum-of-squared residuals        =     20792.393
root mean squared error         =     14.419568
-------------------------------------------------------------------------------
-> id = 3
computing optimal exponential  coefficient (0,1)
optimal exponential coefficient =        0.6927
sum-of-squared residuals        =      21629
root mean squared error         =     14.706801
-------------------------------------------------------------------------------
-> id = 4
computing optimal exponential  coefficient (0,1)
optimal exponential coefficient =        0.3866
sum-of-squared residuals        =     22321.334
root mean squared error         =     14.940326
-------------------------------------------------------------------------------
-> id = 5
computing optimal exponential  coefficient (0,1)
optimal exponential coefficient =        0.4540
sum-of-squared residuals        =     20714.095
root mean squared error         =     14.392392
```

`tssmooth exponential` computed starting values and chose an optimal α for each panel individually. ◁

Treatment of missing values

Missing values in the middle of the data are filled in with the one-step-ahead prediction using the previous values. Missing values at the beginning or end of the data are treated as if the observations were not there.

`tssmooth exponential` treats observations excluded from the sample by `if` and `in` just as if they were missing.

▷ Example 4: Handling missing data in the middle of a sample

Here the 28th observation is missing. The prediction for the 29th observation is repeated in the new series.

```
. use https://www.stata-press.com/data/r18/sales1, clear
. tssmooth exponential sm1=sales, parms(.7) forecast(3)
  (output omitted)
. generate sales2=sales if t!=28
(4 missing values generated)
. tssmooth exponential sm3=sales2, parms(.7) forecast(3)
exponential coefficient  =       0.7000
sum-of-squared residuals =       6842.4
root mean squared error  =       11.817
. list t sales2 sm3 if t>25 & t<31
```

	t	sales2	sm3
26.	26	1011.5	1007.5
27.	27	1028.3	1010.3
28.	28	.	1022.9
29.	29	1028.4	1022.9
30.	30	1054.8	1026.75

Because the data for $t = 28$ are missing, the prediction for period 28 has been used in its place. This implies that the updating equation for period 29 is

$$S_{29} = \alpha S_{28} + (1 - \alpha)S_{28} = S_{28}$$

which explains why the prediction for $t = 28$ is repeated.

Because this is a single-exponential procedure, the loss of that one observation will not be noticed several periods later.

```
. generate diff = sm3-sm1 if t>28
(28 missing values generated)
. list t diff if t>28 & t<39
```

	t	diff
29.	29	-3.5
30.	30	-1.050049
31.	31	-.3150635
32.	32	-.0946045
33.	33	-.0283203
34.	34	-.0085449
35.	35	-.0025635
36.	36	-.0008545
37.	37	-.0003662
38.	38	-.0001221

◁

▷ Example 5: Handling missing data at the beginning and end of a sample

Now consider an example in which there are data missing at the beginning and end of the sample.

```
. generate sales3=sales if t>2 & t<49
(7 missing values generated)
. tssmooth exponential sm4=sales3, parms(.7) forecast(3)
exponential coefficient  =       0.7000
sum-of-squared residuals =       6215.3
root mean squared error  =       11.624
. list t sales sales3 sm4 if t<5 | t>45
```

	t	sales	sales3	sm4
1.	1	1031	.	.
2.	2	1022.1	.	.
3.	3	1005.6	1005.6	1016.787
4.	4	1025	1025	1008.956
46.	46	1055.2	1055.2	1057.2
47.	47	1056.8	1056.8	1055.8
48.	48	1034.5	1034.5	1056.5
49.	49	1041.1	.	1041.1
50.	50	1056.1	.	1041.1
51.	51	.	.	1041.1
52.	52	.	.	1041.1
53.	53	.	.	1041.1

The output above illustrates that missing values at the beginning or end of the sample cause the sample to be truncated. The new series begins with nonmissing data and begins predicting immediately after it stops.

One period after the actual data concludes, the exponential forecast becomes a constant. After the actual end of the data, the forecast at period t is substituted for the missing data. This also illustrates why the forecasted series is a constant.

◁

Stored results

`tssmooth exponential` stores the following in `r()`:

Scalars

`r(N)`	number of observations
`r(alpha)`	α smoothing parameter
`r(rss)`	sum-of-squared prediction errors
`r(rmse)`	root mean squared error
`r(N_pre)`	number of observations used in calculating starting values
`r(s1_0)`	initial value for S_t

Macros

`r(method)`	smoothing method
`r(exp)`	expression specified
`r(timevar)`	time variable specified in `tsset`
`r(panelvar)`	panel variable specified in `tsset`

Methods and formulas

The formulas for deriving smoothed series are as given in the text. When the value of α is not specified, an optimal value is found that minimizes the mean squared forecast error. A method of bisection is used to find the solution to this optimization problem.

A truncated description of the specified exponential filter is used to label the new variable. See [D] **label** for more information about labels.

An untruncated description of the specified exponential filter is saved in the characteristic `tssmooth` for the new variable. See [P] **char** for more information about characteristics.

References

Abraham, B., and J. Ledolter. 1983. *Statistical Methods for Forecasting*. New York: Wiley.

Bowerman, B. L., R. T. O'Connell, and A. B. Koehler. 2005. *Forecasting, Time Series, and Regression: An Applied Approach*. 4th ed. Pacific Grove, CA: Brooks/Cole.

Chatfield, C. 2001. *Time-Series Forecasting*. London: Chapman and Hall/CRC.

——. 2004. *The Analysis of Time Series: An Introduction*. 6th ed. Boca Raton, FL: Chapman and Hall/CRC.

Chatfield, C., and M. Yar. 1988. Holt-Winters forecasting: Some practical issues. *Statistician* 37: 129–140. https://doi.org/10.2307/2348687.

Holt, C. C. 2004. Forecasting seasonals and trends by exponentially weighted moving averages. *International Journal of Forecasting* 20: 5–10. https://doi.org/10.1016/j.ijforecast.2003.09.015.

Montgomery, D. C., L. A. Johnson, and J. S. Gardiner. 1990. *Forecasting and Time Series Analysis*. 2nd ed. New York: McGraw–Hill.

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

Also see

[TS] **tsset** — Declare data to be time-series data

[TS] **tssmooth** — Smooth and forecast univariate time-series data

Stata, Stata Press, and Mata are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2023 StataCorp LLC, College Station, TX, USA. All rights reserved.