**Title** 

# tssmooth ma — Moving-average filter



## Syntax

*Moving average with uniform weights*

`tssmooth ma` [*type*] *newvar* `=` *exp* [*if*] [*in*] `,` `window(`$\#_l$ [$\#_c$ [$\#_f$]]`)` [`replace`]

*Moving average with specified weights*

`tssmooth ma` [*type*] *newvar* `=` *exp* [*if*] [*in*] `,` `weights(`[$numlist_l$] `<`$\#_c$`>` [$numlist_f$]`)`

[`replace`]

You must `tsset` your data before using `tssmooth ma`; see [TS] **tsset**.
*exp* may contain time-series operators; see [U] **11.4.4 Time-series varlists**.

## Menu

Statistics > Time series > Smoothers/univariate forecasters > Moving-average filter

## Description

`tssmooth ma` creates a new series in which each observation is an average of nearby observations in the original series.

In the first syntax, `window()` is required and specifies the span of the filter. `tssmooth ma` constructs a uniformly weighted moving average of the expression.

In the second syntax, `weights()` is required and specifies the weights to be used. `tssmooth ma` then applies the specified weights to construct a weighted moving average of the expression.

## Options

`window(`$\#_l$ [$\#_c$ [$\#_f$]]`)` describes the span of the uniformly weighted moving average.

$\#_l$ specifies the number of lagged terms to be included, $0 \le \#_l \le$ one-half the number of observations in the sample.

$\#_c$ is optional and specifies whether to include the current observation in the filter. A 0 indicates exclusion and 1, inclusion. The current observation is excluded by default.

$\#_f$ is optional and specifies the number of forward terms to be included, $0 \le \#_f \le$ one-half the number of observations in the sample.

`weights(`[*numlist*$_l$] `<`$\#_c$`>` [*numlist*$_f$]`)` is required for the weighted moving average and describes the span of the moving average, as well as the weights to be applied to each term in the average. The middle term literally is surrounded by `<` and `>`, so you might type `weights(1/2 <3> 2/1)`.

*numlist*$_l$ is optional and specifies the weights to be applied to the lagged terms when computing the moving average.

$\#_c$ is required and specifies the weight to be applied to the current term.

*numlist*$_f$ is optional and specifies the weights to be applied to the forward terms when computing the moving average.

The number of elements in each *numlist* is limited to one-half the number of observations in the sample.

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

# Remarks and examples

stata.com

Remarks are presented under the following headings:



## Overview

Moving averages are simple linear filters of the form

$$\widehat{x}_t = \frac{\sum_{i=-l}^{f} w_i x_{t+i}}{\sum_{i=-l}^{f} w_i}$$

where

| | |
|---|---|
| $\widehat{x}_t$ | is the moving average |
| $x_t$ | is the variable or expression to be smoothed |
| $w_i$ | are the weights being applied to the terms in the filter |
| $l$ | is the longest lag in the span of the filter |
| $f$ | is the longest lead in the span of the filter |

Moving averages are used primarily to reduce noise in time-series data. Using moving averages to isolate signals is problematic, however, because the moving averages themselves are serially correlated, even when the underlying data series is not. Still, Chatfield (2004) discusses moving-average filters and provides several specific moving-average filters for extracting certain trends.

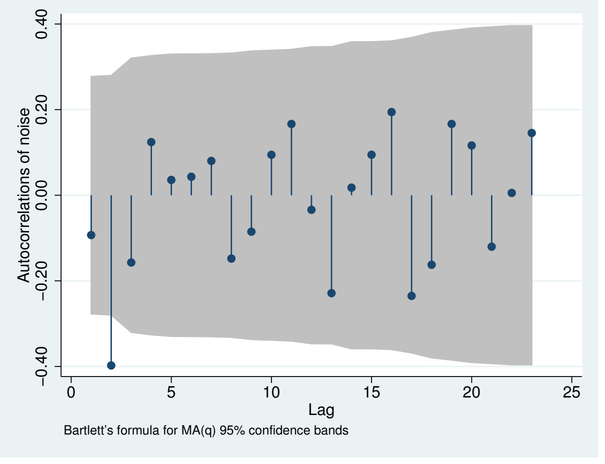
### ▷ Example 1: A symmetric moving-average filter with uniform weights

Suppose that we have a time series of sales data, and we want to separate the data into two components: signal and noise. To eliminate the noise, we apply a moving-average filter. In this example, we use a symmetric moving average with a span of 5. This means that we will average the first two lagged values, the current value, and the first two forward terms of the series, with each term in the average receiving a weight of 1.

```
. use http://www.stata-press.com/data/r13/sales1
. tsset
        time variable:  t, 1 to 50
                delta:  1 unit
. tssmooth ma sm1 = sales, window(2 1 2)
The smoother applied was
     (1/5)*[x(t-2) + x(t-1) + 1*x(t) + x(t+1) + x(t+2)]; x(t)= sales
```

We would like to smooth our series so that there is no autocorrelation in the noise. Below we compute the noise as the difference between the smoothed series and the series itself. Then we use `ac` (see [TS] **corrgram**) to check for autocorrelation in the noise.

```
. generate noise = sales-sm1
. ac noise
```




## ▷ Example 2: A symmetric moving-average filter with nonuniform weights

In the previous example, there is some evidence of negative second-order autocorrelation, possibly due to the uniform weighting or the length of the filter. We are going to specify a shorter filter in which the weights decline as the observations get farther away from the current observation.

The weighted moving-average filter requires that we supply the weights to apply to each element with the `weights()` option. In specifying the weights, we implicitly specify the span of the filter.

Below we use the filter

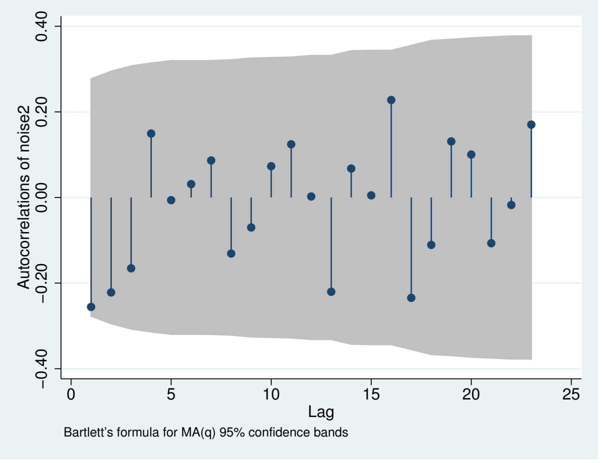
$$\widehat{x}_t = (1/9)(1x_{t-2} + 2x_{t-1} + 3x_t + 2x_{t+1} + 1x_{t+2})$$

In what follows, `1/2` does not mean one-half, it means the numlist `1 2`:

```
. tssmooth ma sm2 = sales, weights( 1/2 <3> 2/1)
The smoother applied was
     (1/9)*[1*x(t-2) + 2*x(t-1) + 3*x(t) + 2*x(t+1) + 1*x(t+2)]; x(t)= sales
. generate noise2 = sales-sm2
```

We compute the noise and use `ac` to check for autocorrelation.

```
. ac noise2
```




The graph shows no significant evidence of autocorrelation in the noise from the second filter.

◁

## ❑ Technical note

`tssmooth ma` gives any missing observations a coefficient of zero in both the uniformly weighted and weighted moving-average filters. This simply means that missing values or missing periods are excluded from the moving average.

Sample restrictions, via `if` and `in`, cause the expression smoothed by `tssmooth ma` to be missing for the excluded observations. Thus sample restrictions have the same effect as missing values in a variable that is filtered in the expression. Also, gaps in the data that are longer than the span of the filter will generate missing values in the filtered series.

Because the first $l$ observations and the last $f$ observations will be outside the span of the filter, those observations will be set to missing in the moving-average series.

❑

## Video example

Time series, part 6: Moving-average smoothers using `tssmooth`

# Stored results

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

Scalars

| | |
|---|---|
| `r(N)` | number of observations |
| `r(w0)` | weight on the current observation |
| `r(wlead#)` | weight on lead #, if leads are specified |
| `r(wlag#)` | weight on lag #, if lags are specified |

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 formula for moving averages is the same as previously given.

A truncated description of the specified moving-average filter labels the new variable. See [D] **label** for more information on labels.

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

# Reference

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

# Also see

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

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