



Published on *STAT 510* (<https://onlinecourses.science.psu.edu/stat510>)

[Home](#) > Lesson 5: Smoothing and Decomposition Methods and More Practice with ARIMA models

# Lesson 5: Smoothing and Decomposition Methods and More Practice with ARIMA models

## Assignments:

- Read Section 2.4 of your text.
- Read through the Lesson 5 online notes that follow.
- Complete Lesson 5 Assignment.

## Overview:

This week we'll continue with ARIMA models and look at decomposition and smoothing methods.

## Learning Objectives:

After successfully completing this lesson, you should be able to:

- Identify and interpret additive and multiplicative decompositions
- Decompose a time series
- Apply a lowess smoother
- Apply a moving averages smoother
- Apply a single exponential smoother

## 5.1 Decomposition Models

Decomposition procedures are used in time series to describe the trend and seasonal factors in a time series. More extensive decompositions might also include long-run cycles, holiday effects, day of week effects and so on. Here, we'll only consider trend and seasonal decompositions.

One of the main objectives for a decomposition is to estimate seasonal effects that can be used to create and present seasonally adjusted values. A seasonally adjusted value removes the seasonal effect from a value so that trends can be seen more clearly. For instance, in many regions of the U.S. unemployment tends to decrease in the summer due to increased employment in agricultural areas. Thus a drop in the unemployment rate in June compared to May doesn't necessarily indicate that there's a trend toward lower unemployment in the country. To see whether there is a real trend, we should adjust for the fact that unemployment is always lower in June than in May.

### Basic Structures

The following two structures are considered for basic decomposition models:

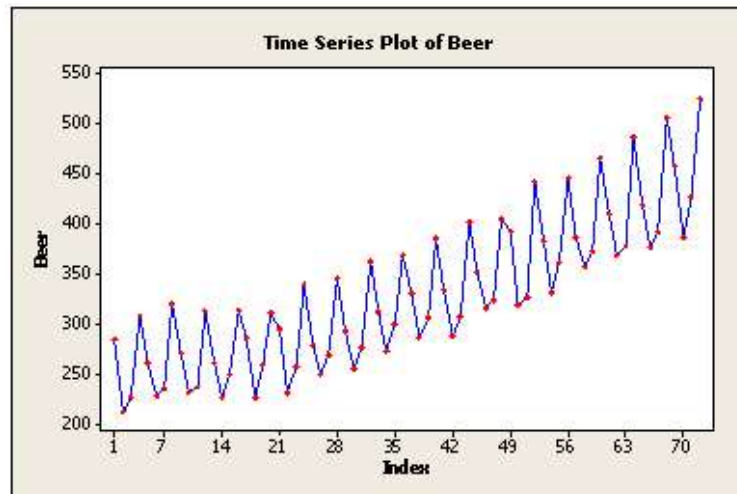
1. Additive:  $x_t = \text{Trend} + \text{Seasonal} + \text{Random}$
2. Multiplicative:  $x_t = \text{Trend} * \text{Seasonal} * \text{Random}$

The “Random” term is often called “Irregular” in software for decompositions.

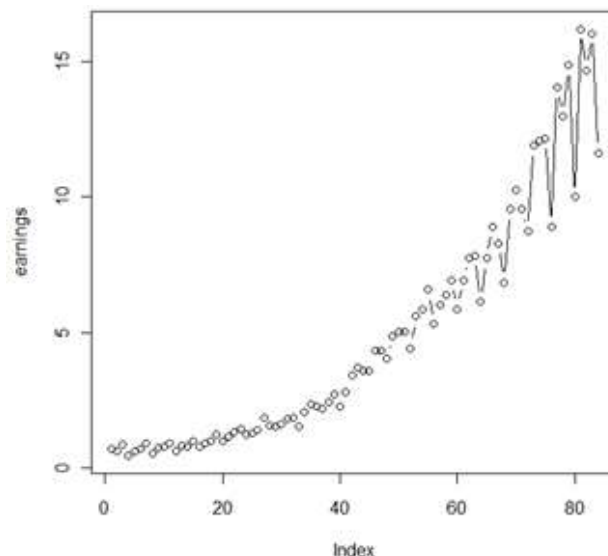
## How to Choose Between Additive and Multiplicative Decompositions

- The additive model is useful when the seasonal variation is relatively constant over time.
- The multiplicative model is useful when the seasonal variation increases over time.

**Example 1:** In Lesson 1.1, we looked at quarterly beer production in Australia. The seasonal variation looked to be about the same magnitude across time, so an additive decomposition might be good. Here’s the time series plot:



**Example 2:** We’ve seen at least one example so far in the course where a multiplicative decomposition would be good – the quarterly earnings data for the Johnson and Johnson Corporations. The seasonal variation increases as we move across time. A multiplicative decomposition could be useful. Here’s the plot of the data:



## Basic Steps in Decomposition

1. The first step is to estimate the trend. Two different approaches could be used for this (with many variations of each).

- One approach is to estimate the trend with a smoothing procedure such as moving averages. (See Lesson 5.2 for more on that.) With this approach no equation is used to describe trend.
- The second approach is to model the trend with a regression equation.

2. The second step is to “de-trend” the series. For an additive decomposition, this is done by subtracting the trend estimates from the series. For a multiplicative decomposition, this is done by dividing the series by the trend values.

3. Next, seasonal factors are estimated using the de-trended series. For monthly data, this entails estimating an effect for each month of the year. For quarterly data, this entails estimating an effect for each quarter. The simplest method for estimating these effects is to average the de-trended values for a specific season. For instance, to get a seasonal effect for January, we average the de-trended values for all Januarys in the series, and so on. (Minitab uses medians rather than means, by the way.)

The seasonal effects are usually adjusted so that they average to 0 for an additive decomposition or they average to 1 for a multiplicative decomposition.

4. The final step is to determine the random (irregular) component.

For the additive model,  $\text{random} = \text{series} - \text{trend} - \text{seasonal}$ .

For the multiplicative model,  $\text{random} = \text{series} / (\text{trend} * \text{seasonal})$

The random component could be analyzed for such things as mean absolute size, or mean squared size (variance), or possibly even for whether the component is actually random or might be modeled with an ARIMA model.

A few programs iterate through the steps 1 to 3. For example, after step 3 we could use the seasonal factors to de-seasonalize the series and then return to step 1 to estimate the trend based on the de-seasonalized series. Minitab does this (and estimates the trend with a straight line in the iteration).

## Decomposition in R

The basic command is `decompose`.

For an additive model `decompose(name of series, type = "additive")`.

For a multiplicative decomposition `decompose(name of series, type = "multiplicative")`.

**Important first step:** As a preliminary you have to use a `ts` command to define the seasonal span for a series.

For quarterly data, it might be `name of series = ts(name of series, freq = 4)`.

For monthly data, it might be `name of series = ts(name of series, freq = 12)`.

You can plot the elements of the decomposition by putting the `decompose` command as an argument of a plot command.

As an example, `plot (decompose(earnings, type="multiplicative"))`.

Another way to plot is to store the results of the decomposition into a named object and then plot the object. As an example,

```
decompearn = decompose (earnings, type = "multiplicative")
plot (decompearn)
```

To see all elements of a stored object, simply type its name. For instance, entering `decompearn` will show all elements of the decomposition in the example above.

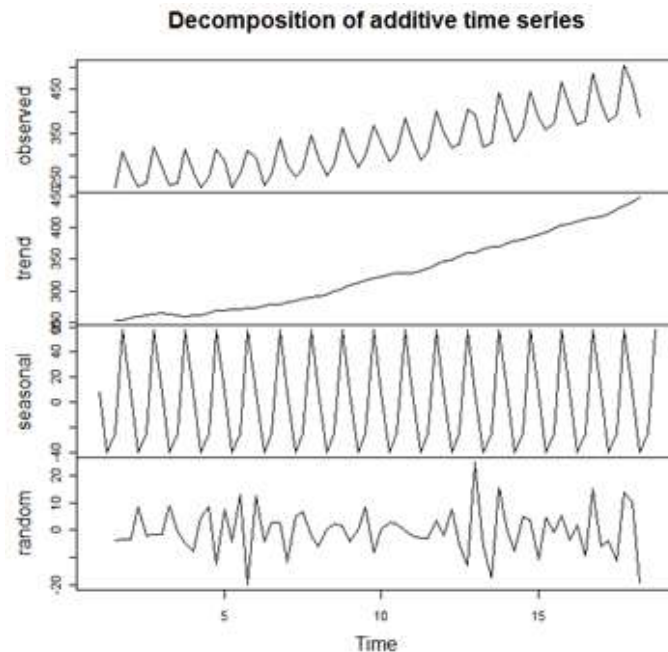
When the decomposition is stored in an object, you also have access to the various elements of the decomposition. For instance, in the example just given, `decompearn$figure` contains the seasonal effects values for four quarters.

You could “print” the seasonal figures simply by entering `decompearn$figure`. You could plot them using `plot (decompearn$figure)`.

## Example 1 Continued: Additive Decomposition for Beer Production

The following commands produced the graph and numerical output that follows for the beer Australian beer production series.

```
beerprod = scan("beerprod.dat")
beerprod = ts(beerprod, freq = 4)
decompbeer = decompose (beerprod, type="additive")
plot (decompbeer)
decompbeer
```



The plot shows the observed series, the smoothed trend line, the seasonal pattern and the random part of the series. Note that the seasonal pattern is a regularly repeating pattern.

Here's the numerical output:

```
$seasonal
  Qtr1    Qtr2    Qtr3    Qtr4
1 7.896324 -40.678676 -24.650735 57.433088
2 7.896324 -40.678676 -24.650735 57.433088
3 7.896324 -40.678676 -24.650735 57.433088
4 7.896324 -40.678676 -24.650735 57.433088
5 7.896324 -40.678676 -24.650735 57.433088
... same rows down to 18 ...
```

```
$trend
  Qtr1    Qtr2    Qtr3    Qtr4
1 NA      NA      255.3250 254.4125
2 257.4500 260.1000 262.8375 264.6875
3 265.4125 264.6500 262.4625 260.4000
4 261.2625 262.9875 266.1875 269.2375
5 270.5125 271.4625 272.1750 274.0125
6 274.3750 277.4500 278.9750 279.1750
7 282.9000 285.2875 287.9375 290.3875
and so on to the 18th row
```

```
$random
      Qtr1      Qtr2      Qtr3      Qtr4
1      NA      NA      -3.77426471 -3.44558824
2 -3.34632353  8.47867647 -2.08676471 -1.72058824
3 -1.40882353  8.82867647 -0.81176471 -4.43308824
4  7.75882353  4.49117647  8.36323529 -12.37058824
5  7.69117647 -4.28382353 12.87573529 -20.04558824
6 12.42867647 -4.17132353  2.87573529  2.59191176
7 -11.69632353  5.19117647  6.51323529 -2.12058824
and so on to the 18th row
```

```
$figure
[1] 7.896324 -40.678676 -24.650735 57.433088
```

## Using the Seasonal Values

The elements of `$figure` are the effects for the four quarters. Note that the seasonal effect values are repeated each year (row) in the `$seasonal` object at the top of this page.

The seasonal values are used to seasonally adjust future values. Suppose for example that the next quarter 4 seasonal value past the end of the series has the value 535. The quarter 4 seasonal effect is 57.433088, or about 57.43. Thus for this future value, the “de-seasonalized” or seasonally adjusted value =  $535 - 57.43 = 477.57$ .

## How the Trend Values Were Calculated

The trend values were determined as “centered” moving averages of span 4 (because there are four quarters per year). Here’s how the centered moving average for time = 3 would be calculated.

Average the observed data values at times 1 to 4:

$$\frac{1}{4}(x_1 + x_2 + x_3 + x_4)$$

Average the values at times 2 to 5:

$$\frac{1}{4}(x_2 + x_3 + x_4 + x_5)$$

Then average those two averages:

$$\frac{1}{2} \left[ \frac{1}{4}(x_1 + x_2 + x_3 + x_4) + \frac{1}{4}(x_2 + x_3 + x_4 + x_5) \right] = \frac{1}{8}x_1 + \frac{1}{4}x_2 + \frac{1}{4}x_3 + \frac{1}{4}x_4 + \frac{1}{8}x_5.$$

More generally, the centered moving average smoother for time  $t$  (with 4 quarters) is

$$\frac{1}{8}x_{t-2} + \frac{1}{4}x_{t-1} + \frac{1}{4}x_t + \frac{1}{4}x_{t+1} + \frac{1}{8}x_{t+2}.$$

Following are the first 8 values in the observed series. The smoothed trend value for time 3 in the series (Qtr 3 of year 1) is 255.325 and the smoothed trend value for time 4 is 254.4125. Use the data below to verify these values (and your understanding of the procedure).

```
Qtr1 Qtr2 Qtr3 Qtr4
```

```
1 284.4 212.8 226.9 308.4
2 262.0 227.9 236.1 320.4
```

For monthly data the centered moving average smoother for time  $t$  will be

$$\frac{1}{24}x_{t-6} + \left( \sum_{j=-5}^5 \frac{1}{12}x_{t+j} \right) + \frac{1}{24}x_{t+6}$$

### Example 1 Continued: Multiplicative Decomposition for Beer Production

The following two commands will do a multiplicative decomposition of the beer production series and print the seasonal effects.

```
decombeermult = decompose (beerprod, type = "multiplicative")
decombeermult$figure
```

The seasonal (quarterly) effects are:

```
1.0237877 0.8753662 0.9233315 1.1775147
```

To seasonally adjust a value, divide the observed value of the series by the seasonal factors. For example if a future quarter 4 value is 535, the seasonally adjusted value =  $535/1.1775147 = 454.34677$ .

### Lowess Seasonal and Trend Decomposition

A lowess smoother essentially replaces values with a “locally weighted” robust regression estimate of the value. The R command `stl` does an additive decomposition in which a lowess smoother is used to estimate the trend and (potentially) the seasonal effects as well. There are several parameters that can be adjusted, but the default does a fairly good job.

For our beer production example, the following command works:

```
stl (beerprod, "periodic")
```

The “periodic” parameter essentially causes the seasonal effects to be estimated in the usual way, as averages of de-trended values. The alternative to this is a `s.window` = some odd number of lags, which uses lowess smoothing procedures to estimate the seasonal effects based on a number of years = the specified `s.window` value. When you do this, the seasonal effects will change as you move through the series.

Here’s a piece of the result of the `stl(beerprod, "periodic")` command. Note that the word “remainder” is used rather than “random.” (Maybe it’s not really random!)

	seasonal	trend	remainder
1 Q1	8.06289	267.3569	8.9802489
1 Q2	-41.58529	261.8710	-7.4856917
1 Q3	-24.68456	257.1444	-5.5598227
1 Q4	58.20698	253.8595	-3.6664533
2 Q1	8.06289	257.4133	-3.4761521
2 Q2	-41.58529	260.6083	8.8769848
2 Q3	-24.68456	262.9967	-2.2121517
2 Q4	58.20698	264.2030	-2.0099463
3 Q1	8.06289	265.5992	-1.7621096
3 Q2	-41.58529	265.2844	9.1008543
3 Q3	-24.68456	262.6567	-0.9721191
3 Q4	58.20698	259.6218	-4.4287360

and so on for years 4 to 16

```
17 Q1    8.06289 417.3459 -6.2088201
17 Q2 -41.58529 420.2610 -1.9757578
17 Q3 -24.68456 428.1381 -10.6535322
17 Q4  58.20698 435.8692  12.0238431
18 Q1    8.06289 441.0740   9.2631448
18 Q2 -41.58529 447.1144 -18.1291496
18 Q3 -24.68456 453.0243  -1.4397012
18 Q4  58.20698 459.3832   7.4098529
```

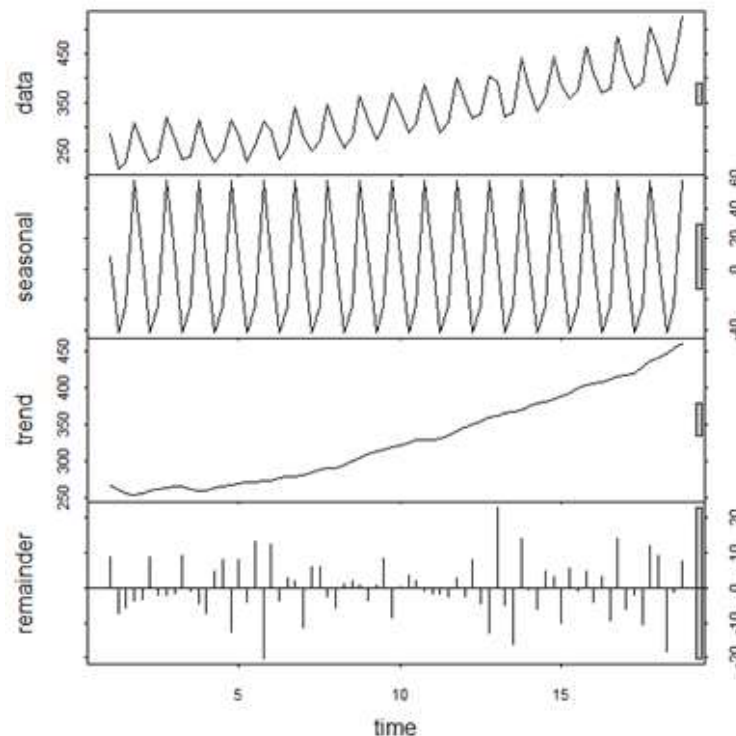
The additive seasonal effects are 8.06289, -41.58529, -24.68456, 58.20698.

These aren't much different than what we got from the additive decompose. Those seasonal values were

```
$figure
```

```
[1] 7.896324 -40.678676 -24.650735 57.433088
```

The command `plot(stl(beerprod, "periodic"))` gave the following plot.



## 5.2 Smoothing Time Series

Smoothing is usually done to help us better see patterns, trends for example, in time series. Generally smooth out the irregular roughness to see a clearer signal. For seasonal data, we might smooth out the seasonality so that we can identify the trend. Smoothing doesn't provide us with a model, but it can be a good first step in describing various components of the series.

The term **filter** is sometimes used to describe a smoothing procedure. For instance, if the smoothed value for a particular time is calculated as a linear combination of observations for surrounding times, it might be said that we've applied a linear filter to the data (not the same as saying the result is a straight line, by the way).

## Moving Averages

The traditional use of the term moving average is that at each point in time we determine (possibly weighted) averages of observed values that surround a particular time.

For instance, at time  $t$ , a “centered moving average of length 3” with equal weights would be the average of values at times  $t-1$ ,  $t$ , and  $t+1$ .

To take away seasonality from a series, so we can better see trend, we would use a moving average with a length = seasonal span. Thus in the smoothed series, each smoothed value has been averaged across all seasons. This might be done by looking at a “one-sided” moving average in which you average all values for the previous year’s worth of data or a centered moving average in which you use values both before and after the current time.

For quarterly data, for example, we could define a smoothed value for time  $t$  as  $(x_t + x_{t-1} + x_{t-2} + x_{t-3})/4$ , the average of this time and the previous 3 quarters. In R code this will be a one-sided filter.

A centered moving average creates a bit of a difficulty when we have an even number of time periods in the seasonal span (as we usually do).

To smooth away seasonality in **quarterly data**, in order to identify trend, the usual convention is to use the moving average smoothed at time  $t$  is

$$\frac{1}{8}x_{t-2} + \frac{1}{4}x_{t-1} + \frac{1}{4}x_t + \frac{1}{4}x_{t+1} + \frac{1}{8}x_{t+2}$$

To smooth away seasonality in **monthly data**, in order to identify trend, the usual convention is to use the moving average smoothed at time  $t$  is

$$\frac{1}{24}x_{t-6} + \frac{1}{12}x_{t-5} + \frac{1}{12}x_{t-4} + \cdots + \frac{1}{12}x_{t+4} + \frac{1}{12}x_{t+5} + \frac{1}{24}x_{t+6}$$

That is, we apply weight  $1/24$  to values at times  $t-6$  and  $t+6$  and weight  $1/12$  to all values at all times between  $t-5$  and  $t+5$ .

In the R `filter` command, we’ll specify a two-sided filter when we want to use values that come both before and after the time for which we’re smoothing.

Note that on page 71 of our book, the authors apply equal weights across a centered seasonal moving average. That’s okay too. For instance, a quarterly smoother might be smoothed at time  $t$  is

$$\frac{1}{5}x_{t-2} + \frac{1}{5}x_{t-1} + \frac{1}{5}x_t + \frac{1}{5}x_{t+1} + \frac{1}{5}x_{t+2}$$

A monthly smoother might apply a weight of  $1/13$  to all values from times  $t-6$  to  $t+6$ .

The code the authors use on page 72 takes advantage of a `rep` command that repeats a value a certain number of times. They don’t use the “filter” parameter within the filter command.

### Example 1 Quarterly Beer Production in Australia

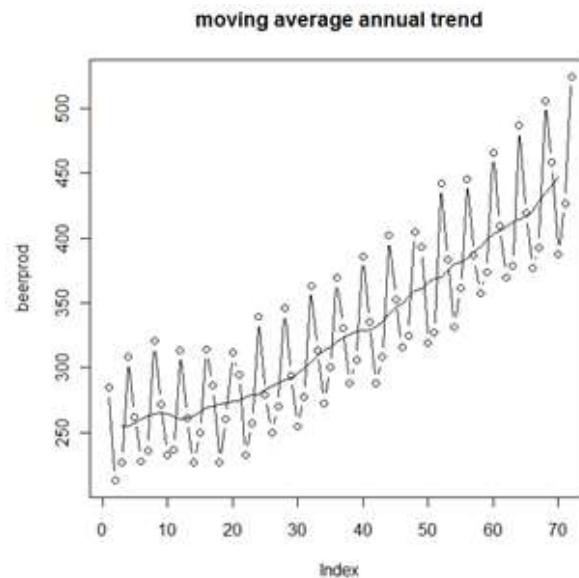
In both Lesson 1 and Lesson 4, we looked at a series of quarterly beer production in Australia. The



following R code creates a smoothed series that lets us see the trend pattern, and plots this trend pattern on the same graph as the time series. The second command creates and stores the smoothed series in the object called `trendpattern`. Note that within the `filter` command, the parameter named `filter` gives the coefficients for our smoothing and `sides = 2` causes a centered smooth to be calculated.

```
beerprod = scan("beerprod.dat")
trendpattern = filter (beerprod, filter = c(1/8, 1/4, 1/4, 1/4, 1/8), sides=2)
plot (beerprod, type= "b", main = "moving average annual trend")
lines (trendpattern)
```

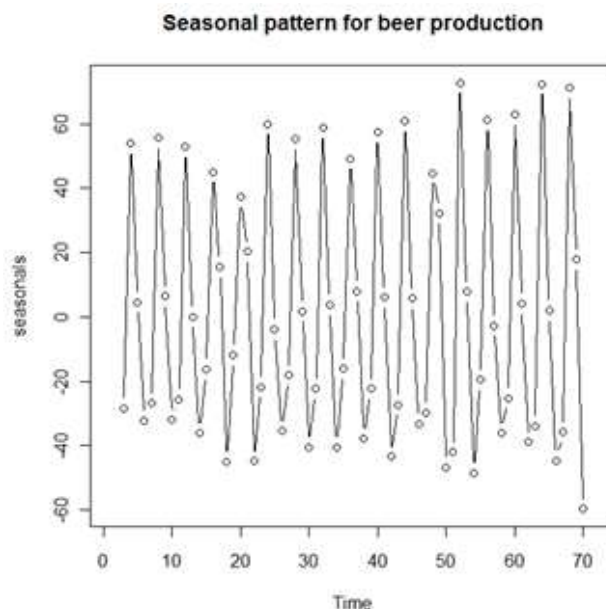
Here's the result:



We might subtract the trend pattern from the data values to get a better look at seasonality. Here's how that would be done:

```
seasonals = beerprod - trendpattern
plot (seasonals, type = "b", main = "Seasonal pattern for beer production")
```

The result follows:



Another possibility for smoothing series to see trend is the one-sided filter

```
trendpattern2 = filter (beerprod, filter = c(1/4, 1/4, 1/4, 1/4), sides=1)
```

With this, the smoothed value is the average of the past year.

### Example 2: U.S. Monthly Unemployment

In the homework for week 4 you looked at a monthly series of U.S. Unemployment for 1948-1978. Here's a smoothing done to look at the trend.

```
trendunemploy=filter(unemploy,
filter=c(1/24,1/12,1/12,1/12,1/12,1/12,1/12,1/12,1/12,1/12,1/12,1/12,1/24), sides=2)
trendunemploy = ts(trendunemploy, start = c(1948,1), freq = 12)
plot(trendunemploy, main="Trend in U.S. Unemployment, 1948-1978", xlab = "Year")
```

Only the smoothed trend is plotted. The second command identifies the calendar time characteristics of the series. That makes the plot have a more meaningful axis. The plot follows.



### Non-Seasonal Series

For non-seasonal series, you aren't bound to smooth over any particular span. For smoothing you should experiment with moving averages of different spans. Those spans of time could be relatively short. The objective is to knock off the rough edges to see what trend or pattern might be there.

### Other Smoothing Methods (Section 2.4)

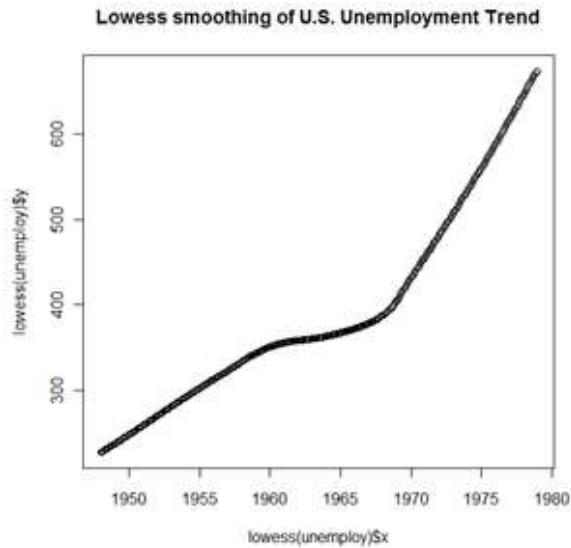
Section 2.4 describes several sophisticated and useful alternatives to moving average smoothing. The details may seem sketchy, but that's okay because we don't want to get bogged down in lots of details for those methods. Of the alternative methods described in Section 2.4, lowess (locally weighted regression) may be the most widely used.

### Example 2 Continued

The following plot is smoothed trend line for the U.S. Unemployment series, found using a lowess smoother in which a substantial amount (2/3) contributed to each smoothed estimate. Note that this smoothed the series more aggressively than the moving average.

The commands used were

```
unemploy = ts(unemploy, start = c(1948,1), freq=12)
plot(lowess(unemploy, f = 2/3), main = "Lowess smoothing of U.S. Unemployment Trend")
```



## Single Exponential Smoothing

The basic forecasting equation for single exponential smoothing is often given as

$$\hat{x}_{t+1} = \alpha x_t + (1 - \alpha) \hat{x}_t \quad (1)$$

We forecast the value of  $x$  at time  $t+1$  to be a weighted combination of the observed value at time  $t$  and the forecasted value at time  $t$ . Although the method is called a smoothing method, it's principally used for short run forecasting.

The value of  $\alpha$  is called the **smoothing constant**. For whatever reason,  $\alpha = 0.2$  is a popular default choice of programs. This puts a weight of .2 on the most recent observation and a weight of  $1 - .2 = .8$  on the most recent forecast. With a relatively small value of  $\alpha$ , the smoothing will be relatively more extensive. With a relatively large value of  $\alpha$ , the smoothing is relatively less extensive as more weight will be put on the observed value.

This is simple one-step ahead forecasting method that at first glance seems not to require a model for the data. In fact, this method is equivalent to the use of an ARIMA(0,1,1) model with no constant.

The optimal procedure is to fit an ARIMA (0,1,1) model to the observed dataset and use the results to determine the value of  $\alpha$ . This is "optimal" in the sense of creating the best  $\alpha$  for the data already observed.

Although the goal is smoothing and one step ahead forecasting, the equivalence to the ARIMA(0,1,1) model does bring up a good point. We shouldn't blindly apply exponential smoothing because the underlying process might not be well modeled by an ARIMA(0,1,1).

## ARIMA(0,1,1) and Exponential Smoothing Equivalence

Consider an ARIMA(0,1,1) with mean  $\mu = 0$  for the first differences,  $x_t - x_{t-1}$ :

The model is  $x_t - x_{t-1} = w_t + \theta_1 w_{t-1}$ .

Equivalently,  $x_t = x_{t-1} + w_t + \theta_1 w_{t-1}$ .

To forecast at time  $t+1$ , we consider  $x_{t+1} = x_t + w_{t+1} + \theta_1 w_t$ .

Because  $w_{t+1} = x_{t+1} - \hat{x}_{t+1}$ ,

$$\begin{aligned}\hat{x}_{t+1} &= x_t + \theta_1 w_t \\ &= x_t + \theta_1 (x_t - \hat{x}_t) \\ &= (1 + \theta_1)x_t - \theta_1 \hat{x}_t\end{aligned}$$

If we let  $\alpha = (1 + \theta_1)$  and thus  $-\theta_1 = 1 - \alpha$ , we see the equivalence to equation (1) above.

### Why the Method is Called Exponential Smoothing

Starting with  $\hat{x}_{t+1} = \alpha x_t + (1 - \alpha)\hat{x}_t$ , we can substitute for  $\hat{x}_t$ .

This yields the following:

$$\begin{aligned}\hat{x}_{t+1} &= \alpha x_t + (1 - \alpha)[\alpha x_{t-1} + (1 - \alpha)\hat{x}_{t-1}] \\ &= \alpha x_t + \alpha(1 - \alpha)x_{t-1} + (1 - \alpha)^2 \hat{x}_{t-1}\end{aligned}$$

Continue in this fashion by successively substituting for the forecasted value on the right side of the equation. This leads to:

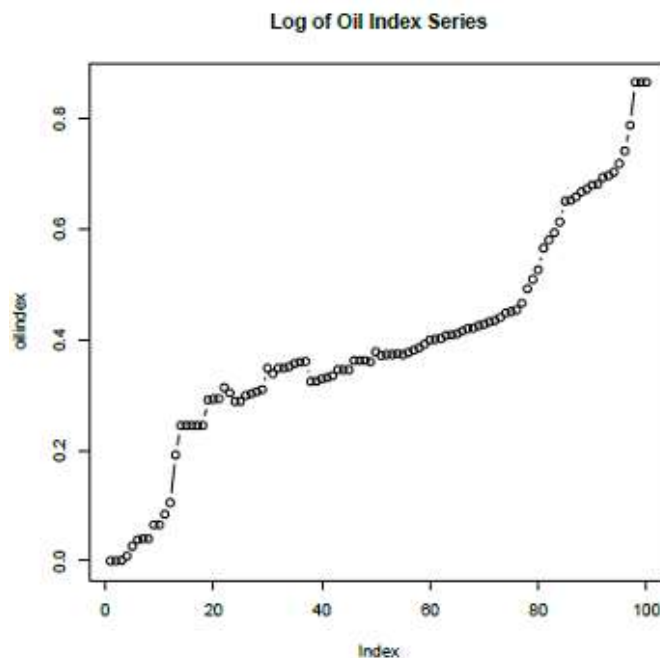
$$\hat{x}_{t+1} = \alpha x_t + \alpha(1 - \alpha)x_{t-1} + \alpha(1 - \alpha)^2 x_{t-2} + \cdots + \alpha(1 - \alpha)^j x_{t-j} + \cdots + \alpha(1 - \alpha)^{t-1} x_1 \quad (2)$$

Equation 2 shows that the *forecasted value is a weighted average* of all past values of the series, with *exponentially changing weights* as we move back in the series.

### Optimal Exponential Smoothing in R

Basically, we just fit an ARIMA(0,1,1) to the data and determine the  $\alpha$  coefficient. We can examine the fit of the smooth by comparing the predicted values to the actual series. Exponential smoothing tends to be used more as a forecasting tool than a true smoother, so we're looking to see if we have a good fit.

**Example 3:**  $n = 100$  monthly observations of the logarithm of an oil price index in the United States. The data series is:



An ARIMA(0,1,1) fit in R gave an MA(1) coefficient = 0.3877. Thus  $\alpha = (1 + \theta_1) = 1.3877$  and  $1 - \alpha = -0.3877$ . The exponential smoothing forecasting equation is

$$\hat{x}_{t+1} = 1.3877x_t - 0.3877\hat{x}_t$$

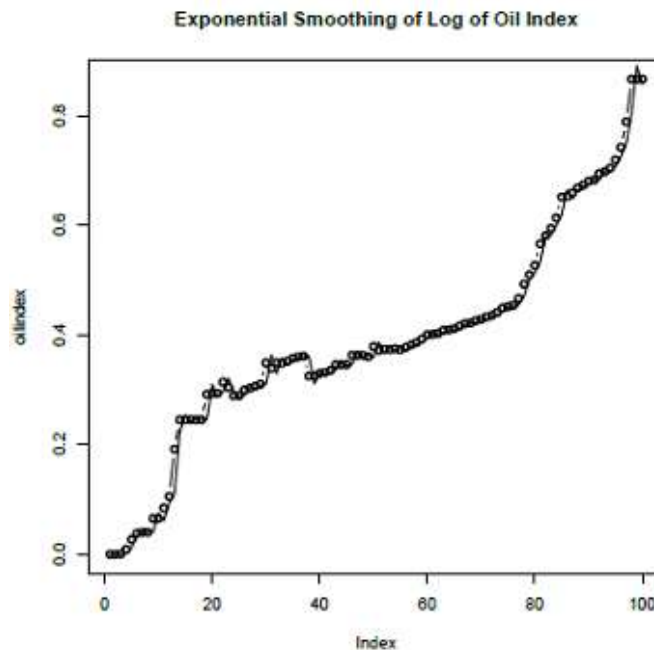
At time 100, the observed value of the series is  $x_{100} = 0.86601$ . The predicted value for the series at that time is

$$\hat{x}_{100} = 0.856789$$

Thus the forecast for time 101 is

$$\hat{x}_{101} = 1.3877x_{100} - 0.3877\hat{x}_{100} = 1.3877(0.86601) - 0.3877(0.856789) = 0.8696$$

Following is how well the smoother fits the series. It's a good fit. That's a good sign for forecasting, the main purpose for this "smoother."



Here are the commands used to generate the output for this example:

```
oilindex = scan("oildata.dat")
plot (oilindex, type = "b", main = "Log of Oil Index Series")
expsmoothfit = arima (oilindex, order = c(0,1,1))
expsmoothfit # to see the arima results
predicted = oilindex - expsmoothfit$residuals # predicted values
plot (oilindex, type="b", main = "Exponential Smoothing of Log of Oil Index")
lines (predicted)
1.3877*oilindex[100]-0.3877*predicted[100] # forecast for time 101
```

## Double Exponential Smoothing

Double exponential smoothing might be used when there's trend (either long run or short run), but no seasonality.

Essentially the method creates a forecast by combining exponentially smoothed estimates of the trend (slope of a straight line) and the level (basically, the intercept of a straight line).

Two different weights, or smoothing parameters, are used to update these two components at each time.

The smoothed “level” is more or less equivalent to a simple exponential smoothing of the data values and the smoothed trend is more or less equivalent to a simple exponential smoothing of the first differences.

The procedure is equivalent to fitting an ARIMA(0,2,2) model, with no constant; it can be carried out with an ARIMA(0,2,2) fit.

$$(1 - B)^2 x_t = (1 + \theta_1 B + \theta_2 B^2) w_t.$$

Source URL: <https://onlinecourses.science.psu.edu/stat510/node/51>