STAT 5170: Applied Time Series

Course notes for part A of learning unit 7

Section 7A.1: Moving average smoothing of a time series.

In previous units we encountered ideas that are connected to a class of statistical techniques known as smoothing. For example, the term "smoothing" was used explicitly in the section of our discussion on periodograms where the aim was to smooth the periodogram. Shortly thereafter, we encountered the concept of a low-pass filter, which borrows ideas from periodogram smoothing but applies them to the purpose of smoothing the time series itself. These ideas were all discussed from the frequency-domain perspective. However, they are also relevant to the time-domain perspective, where a step in time series analysis is to detrend a time series under the decomposition $x_t = \mu_t + \epsilon_t$, or, if seasonality is present, the decomposition $x_t = \mu_t + s_t + \epsilon_t$. Here μ_t and s_t are deterministic components of the time series model, and (ϵ_t) is a stationary time series of residual deviations.

Smoothing that is implemented by the use of a low-pass filter may be understood through the formula

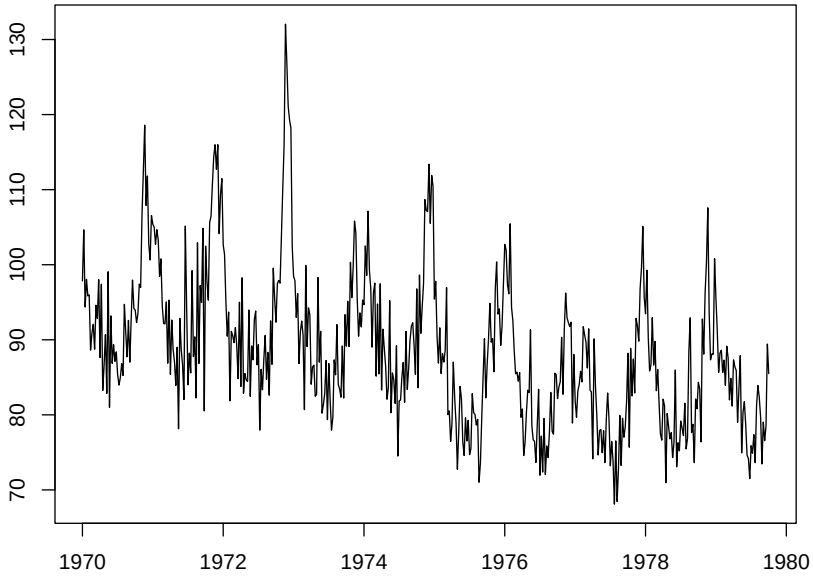
$$\hat{x}_t \;=\; \sum_{j=-m}^{m} w_j x_{t-j}.$$

This is a specialization of the linear-filter formula, where summation is doubly-infinite. The above formula emphasizes the case in which the w_j are defined from a kernel scheme, such as a Daniell kernel. In this specialized context, the w_j are constrained to be positive, $w_j \geq 0$, and to sum to one, $\sum_{j=-m}^{m} w_j = 1$. The approach is mathematically connected to linear filters, but in the time domain it is often referred to as *moving-average smoothing*, and the w_j, though defining the impulse response function of a linear filter, are simply called *weights*. The range of values $j = 0 \pm 1 \pm \cdots \pm m$ is sometimes called the *smoothing window*.

The parameter m, along with the weights, w_j, is a tuning parameter, which controls *how smooth* the graph of $(\hat{x}_t)$ appears, and how informative that graph is to the purpose of further modeling and analysis. The graph of $(\hat{x}_t)$ with m set to a large value will be more smooth than a graph with m set to a smaller value. This is illustrated in the following example.

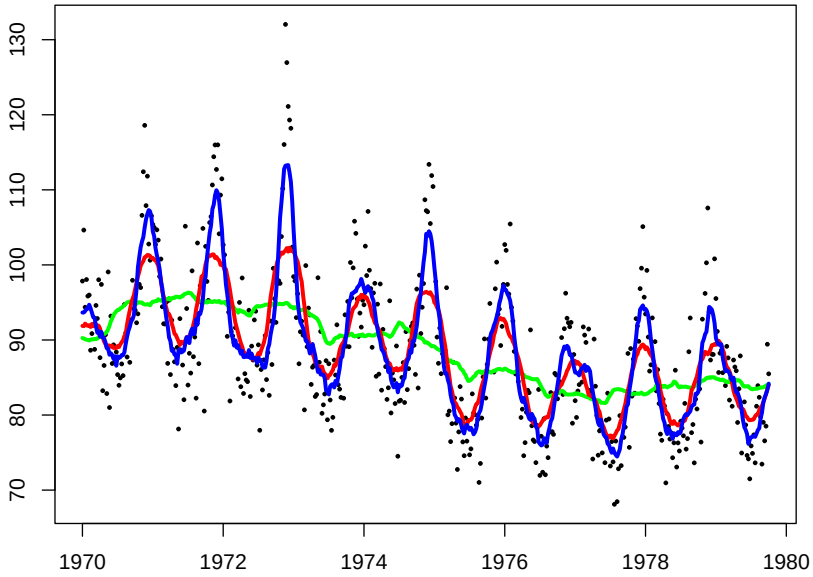
Example: Moving-average smoothing of cardiovascular mortality

In a previous learning unit we worked with a time series of weekly measurements of cardiovascular mortality collected in Los Angeles over a ten year period. A plot of this time series is as follows.



Among the notable patterns in these data are a slight decreasing trend, which may not be exactly linear, and a seasonal cycle with a period of about one year, or about $1/\omega = 52$ weeks.

A plot depicting the results of moving-average smoothing of cardiovascular mortality at three settings of m is shown below.



Here, the blue line is calculated with $m = 25$, the red line is calculated with $m = 12$, and the green line with $m = 5$. This corresponds to smoothing-window sizes of $L = 2m + 1 = 51, 25$, and 11. Given that the time series as a whole covers $n = 508$ weeks, these numbers correspond to about 10%, 5%, and 2% coverage of the time series by each smoothing window. In each calculation indicated in the above figure, smoothing is unweighted, for which each $w_j = 1/L$.

In these graphs, the "rougher" (more variable) curves, at $m = 5$ and 12 do well at tracking cyclic patterns, while the "smoother" (less variable) curve, at $m = 25$, emphasizes a long-term trend. □

Smoothing is largely motivated by its ability to clarify patterns in time series data, and generally to make graphical displays more informative. In our exploration of the frequency domain we also observed that smoothing can improve the inferential precision of a periodogram. A parallel property is at play in the time domain. In terms of variances, the central idea is that averaging reduces variance. Consider, for instance, an independent and identically distributed (white-noise) time series (x_t), wherein the variance of each measurement is $Var[x_t] = \sigma_x^2$, while the variance of the unweighted average of measurements $x_{t-m}, \ldots, x_t, \ldots, x_{t+m}$ is, writing $L = 2m + 1$,

$$Var\left[\frac{1}{L}\sum_{j=-m}^{m} x_{t-j}\right] = \frac{1}{L^2}\sum_{j=-m}^{m} Var\left[x_{t-j}\right] = \frac{1}{L^2}\sum_{j=-m}^{m}\sigma_x^2 = \frac{1}{L^2}L\sigma_x^2 = \frac{\sigma_x^2}{L}.$$

The moving-average has smaller variance than the variance of an individual measurement, hence inference based on the moving average is more precise. The situation is more complicated in the general case of a time series (x_t), which is possibly subject to complicated trends and dependencies. However, provided the smoothing window covers a small proportion of the time series, and still contains a fair number of points, smoothing a time series can improve inferential precision in a similar manner as smoothing a periodogram improves precision in the frequency domain.

In moving-average smoothing, non-uniform weights are typically applied using a *kernel strategy*. We have already examined one such kernel strategy in our exploration of periodogram smoothing, the Daniell kernel strategy. In general, a kernel strategy is implemented by specifying weights, w_j, from a *kernel function*, $K(\cdot)$, which is often a rescaled density function of a continuous distribution. For example, a Gaussian kernel is defined from the standard-normal density function

$$p(z) = \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}z^2}.$$

For the smoothing window defined at points $j = 0 \pm 1 \pm \cdots \pm m$, the kernel function is defined from $p(\cdot)$ as

$$K(j) = p(j/b) \Big/ \sum_{k=-m}^{m} p(k/b)$$

wherein b is a *bandwidth parameter* that controls how sharply the kernel function decrease at values far from zero. Subsequently, the weights are assigned according to the formula $w_j \propto K(j)$. Rescaling the density function ensures that the weights sum to one, $\sum_{j=-m}^{m} w_j = 1$.

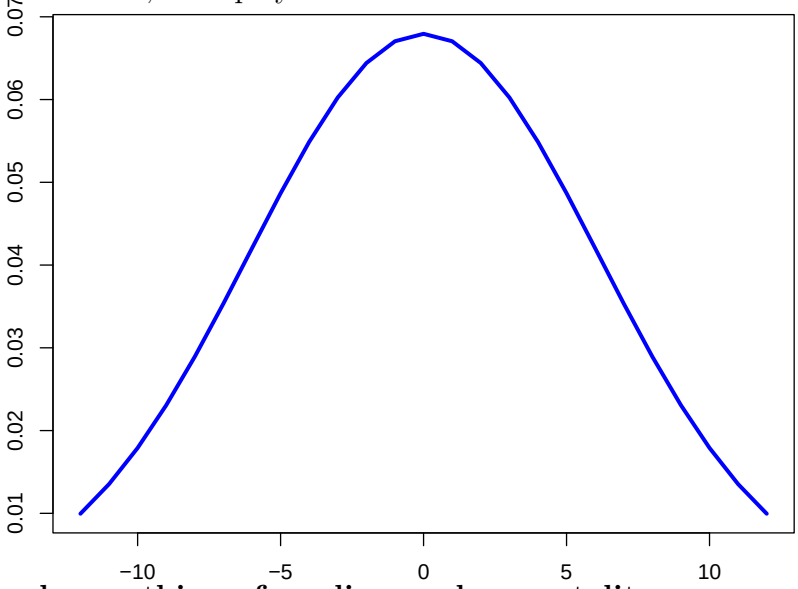
When working with a Gaussian kernel, it can be helpful to note that the bandwidth parameter b operates as the standard deviation parameter in the normal density formula

$$\frac{1}{b}p(z/b) = \frac{1}{\sqrt{2\pi b^2}}e^{-\frac{1}{2b^2}z^2}.$$

For example, this motivates a scheme for specifying b such that a specified pair of quantiles of a mean-zero normal distribution with standard deviation b would match the endpoints of

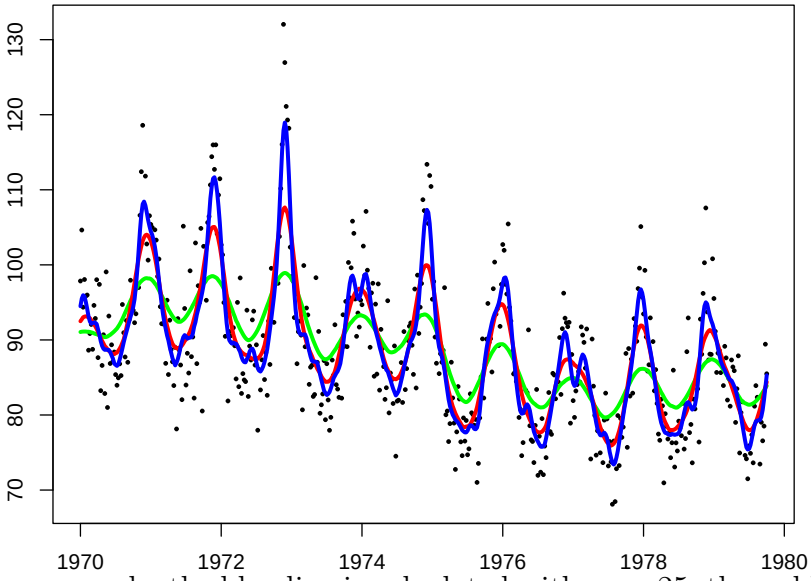
the smoothing window, $-m$ and m. An explicit formula is $b = m/z_{\alpha/2}$, for a given value α such as 0.05. Here, z_p denotes the pth quantile of a standard normal distribution. It follows that the $\alpha/2$ and $1 - \alpha/2$ quantiles of a $N(0, b^2)$ distribution are $\pm bz_{\alpha/2}$, and the given formula guarantees these match the endpoints $\pm m$.

A graph of a Gaussian kernel for $m = 12$, with the bandwidth parameter specified by the above scheme, at $\alpha = 0.05$, is displayed as follows.



Example: Kernel smoothing of cardiovascular mortality

The results of applying kernel smoothing to the cardiovascular mortality in Los Angeles time series is shown below.



As in the previous example, the blue line is calculated with $m = 25$, the red line is calculated with $m = 12$, and the green line with $m = 5$. The weights are defined from from a Gaussian kernel with the bandwidth parameter set to $b = m/z_{\alpha/2}$ for $\alpha = 0.05$.

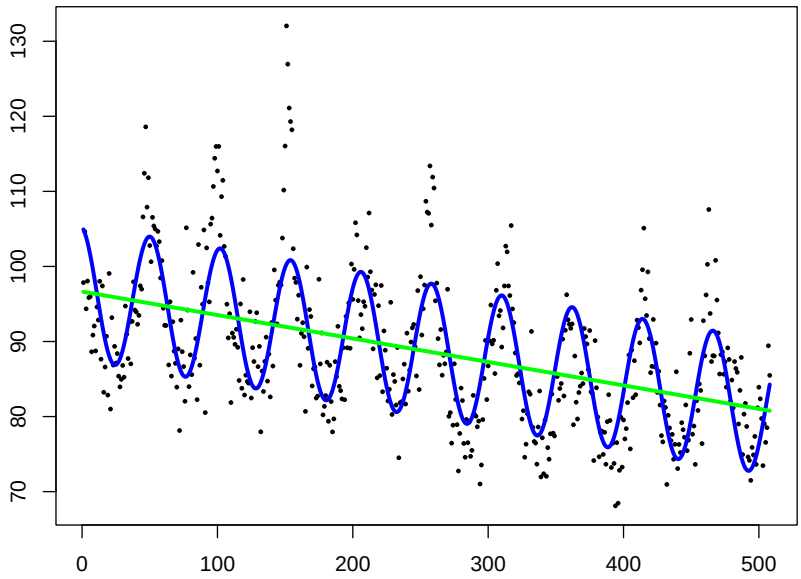
Observe that weighted smoothing more clearly captures the peaks and troughs of the cyclic patterns, compared to unweighted smoothing. □

Section 7A.2: Regression smoothing.

In a previous learning unit, we discussed the use of polynomial regression, augmented with seasonal variables, to de-trend a time series.

Example: De-trending cardiovascular mortality

The following displays a graph of the fitted mean from a regression calculation in on the cardiovascular mortality time series under two mean functions.



The green line is a graph of fitted values under the linear mean function.

$$\mu_t \quad = \quad \beta_1 + \beta_2 t.$$

The blue line is a graph of fitted values under the linear mean function augmented with two variables that capture a yearly seasonal cycle,

$$\mu_t \quad = \quad \beta_1 + \beta_2 t + \beta_2 \cos(2\pi\omega t) + \beta_4 \sin(2\pi\omega t).$$

Since this is weekly data, the yearly cycle is specified at frequency $\omega = 1/52$. □

Applying regression in the manner described above is sometimes called a *global* approach to capturing complex patterns in a time series. This refers to the use of functional forms that would apply across the entire time series, from its initial measurement to its final measurement. A number of techniques have been proposed that, in effect, apply regression within *local regions* of the time series, such that the fitting calculation in one region would be minimally influenced by the fitting calculations made in other regions, particularly those that are far away in time. This imbues flexibility in the fitting calculation as a whole.

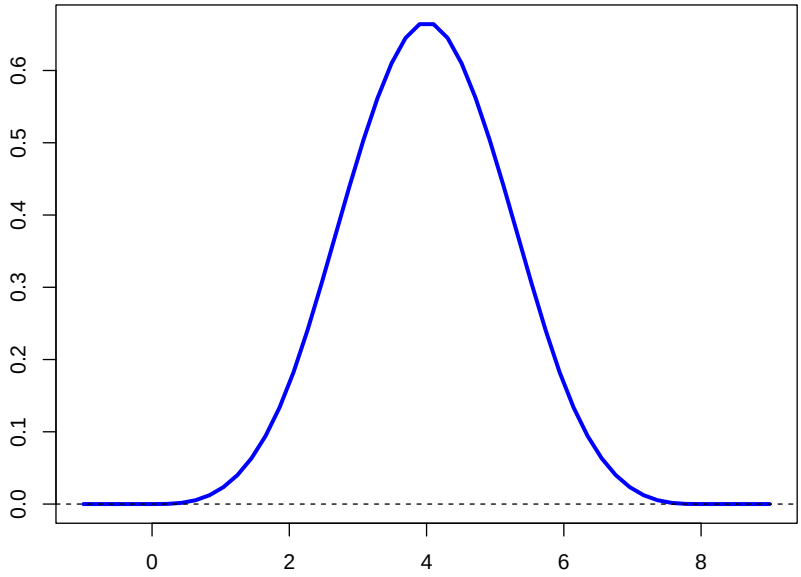
It is sometimes said that local regression is a way of de-trending data without specifying a mean function through the use of standard parametric functions, such as polynomials and trigonometric functions. Accordingly, this way of applying regression is said to be *non-parametric*.

One way to implement local regression is by way of incorporating a large set of flexible regression functions into the regression model. A wide variety of systems that defining such functions is available for this purpose. Once choice is the type of regression functions known as *cubic B-spline* functions. As we will see, they are particularly appealing for present purposes because they emphasize local regions of the time series.

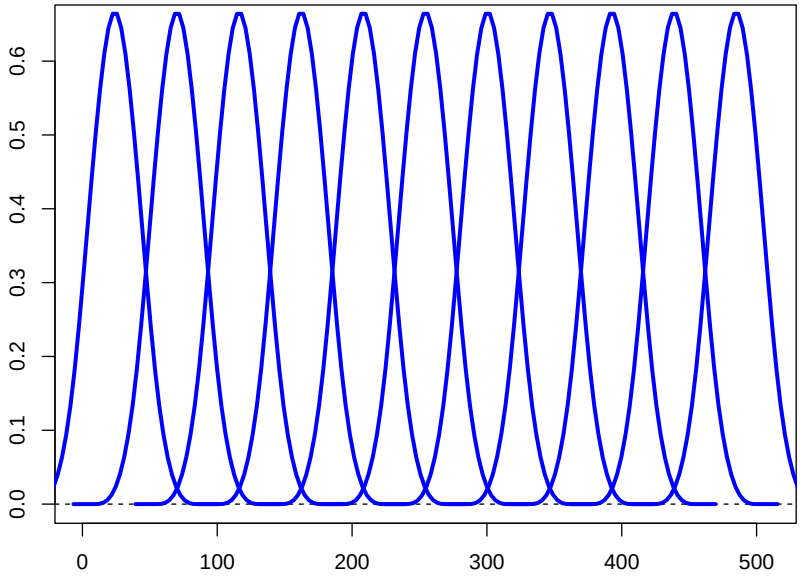
A cubic B-spline function is defined with respect to set of evenly-spaced reference time points known as *knots*. Each individual function need only reference five such knots, located around a *central knot*. Denote by t_0 the central knot, and $t_k = t_0 + \delta k$ the surrounding four knots, where $k = \pm 1$ or ± 2, and δ is the distance between knots. By this notation, an individual cubic B-spline function is defined according to

$$B(t; t_0, \delta) \quad = \quad \begin{cases} \frac{1}{6}u^3 & \text{if } t_{-2} \le t < t_{-1}, u = (t - t_{-2})/\delta \\ \frac{1}{6}(1 + 3u + 3u^2 - 3u^3) & \text{if } t_{-1} \le t < t_0, u = (t - t_{-1})/\delta \\ \frac{1}{6}(4 - 6u^2 + 3u^3) & \text{if } t_0 \le t < t_1, u = (t - t_0)/\delta \\ \frac{1}{6}(1 - 3u + 3u^2 - u^3) & \text{if } t_1 \le t < t_2, u = (t - t_1)/\delta \\ 0 & \text{otherwise} \end{cases}$$

The following displays the graph of the cubic B-spline function defined at the central knot $t_0 = 4$, and between-knot distance $\delta = 2$.



Local regression is accomplished by defining regression functions in correspondence with a set of cubic B-spline functions across the time series's entire time range, that set being defined along a sequence of evenly spaced central knots. The following displays overlaid graphs of one set of eleven cubic B-spline functions, covering the range of 508 time units.



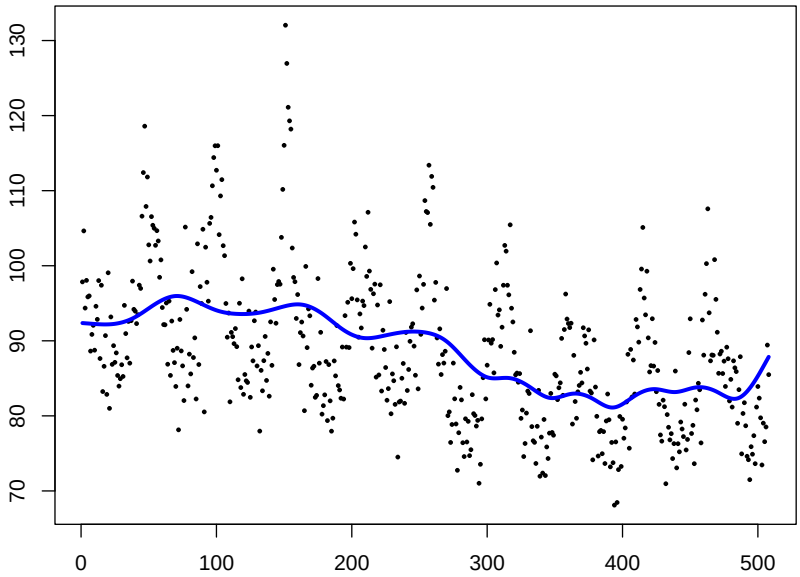
The knots that define an individual cubic B-spline function are spaced at 2/3 the distance between central knots.

To be clear, each regression function is defined from a single cubic B-spline function. Thus, huge sections of an individual regression function would take the value zero; non-zeros would only appear in a local region defined by the knots $t_k = 0, \pm 1, \pm 2$ that define the cubic B-spline function. Writing $B_c(\cdot)$ for the cubic B-spline function defined at the c'th of g central knots, the mean function of the time series is specified as

$$\mu_t \quad = \quad \beta_1 + \beta_2 B_1(t) + \cdots + \beta_{g+1} B_g(t).$$

Example: Cubic B-spline regression of cardiovascular mortality

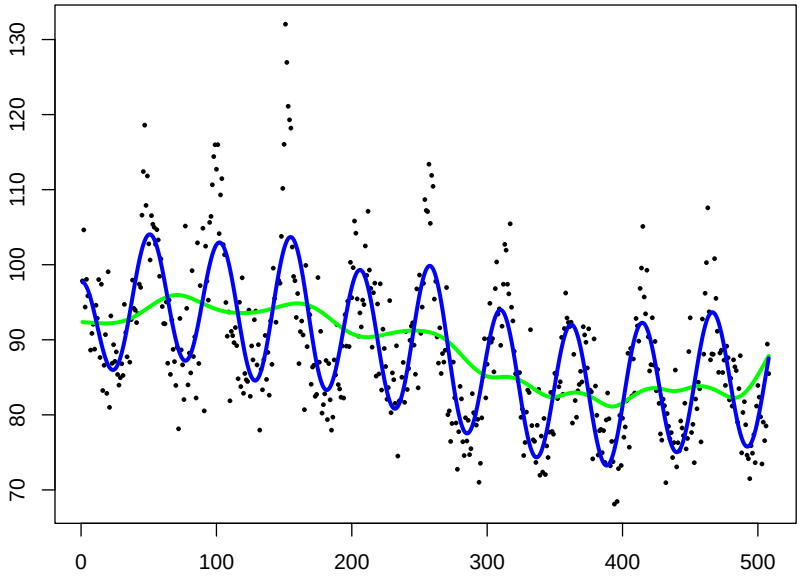
Fitted values from regression applied to the cardiovascular mortality in Los Angeles, defined from the $g = 11$ aforementioned cubic B-spline functions covering the $n = 508$ time points of the data series, are shown in the following figure.



These values are moderately successful at capturing non-parametric patterns of the time series's long term trend; however, it fails to adequately capture the seasonal cycles. An advantage of regression smoothing is that it is straightforward to extend the technique when it is thought that some patterns may be captured using ordinary parametric functions. In the present situation, our motivation is to consider augmenting the specified mean function by adding trigonometric functions to capture the yearly seasonal cycles, while continuing to model the long term trends non-parametrically. This augmented mean function is

$$\mu_t \;=\; \beta_1 + \beta_2 B_1(t) + \cdots + \beta_{g+1} B_g(t) + \beta_{g+2}\cos(2\pi\omega t) + \beta_{g+3}\sin(2\pi\omega t),$$

where $\omega = 1/52$. A plot of the fitted values that result from a regression analysis under this mean function, overlaid with the previous graph of fitted values associated with just the cubic B-spline functions, is shown below. The former set of fitted values are in blue, and the latter are in green.



This calculation does a much better job at capturing trends in the data than that using cubic B-spline functions alone. □

Cubic B-spline functions define one of many non-parameteric regression techniques. They are flexible and adapt to situations in which time points are not evenly spaced. However, computational efficiency is best in the case of evenly-spaced time points, especially when the central knots and knots that define individual cubic B-spline functions are all on the same grid of points, which admits use of algorithms that are effective at organizing computations for efficiency.

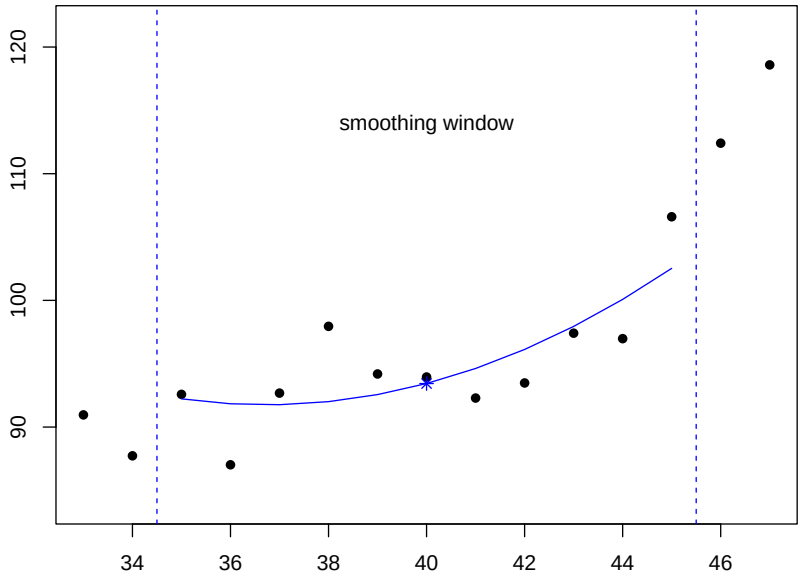
Cubic B-spline functions are a subclass arising from a system that formulates B-spline functions at any order, including linear, quadratic, cubic, quartic, *etc.*. The shape of a cubic B-spline function resembles a bell curve, and that is part of what makes them especially useful. They also carry the property of *bounded support*, which means that they are zero outside of a small range of values. This property can be exploited in the design of efficient computational algorithms.

A persistent problem in many smoothing and non-parametric regression techniques is that they often do not capture patterns in data near the edges of the time period in which the time series is measured, *i.e.*, for t near 1 or n. This is can be seen in many of the graphs throughout this set of notes, where fitted values at the beginning or end of the plot appear less than fully anchored, at least relative to values in the interior portions of the graph. Specialized methods have beed devised to improve the effectiveness of smoothing and non-parametric regression techniques near the edges of a time series, but these are not employed here.

The final smoothing technique that is discussed here is local polynomial regression, which might be viewed as a combination of moving-average smoothing, kernel smoothing, and local regression. The starting point would be moving-average smoothing, wherein fitted values are obtained from repeated statistical summarization of data within relatively narrow smoothing windows. In local polynomial regression, the first modification that is made is to replace the averaging operation $\hat{x}_t = \sum_{j=-m}^{m} w_j x_{t-j}$ with polynomial regression fitting at the midpoint of the smoothing window. Very often, quadratic regression, rather than higher-order regression such as cubic or quartic regression, is sufficient, whereby

$$\hat{x}_t \;=\; \hat{\beta}_1 + \hat{\beta}_2 t + \hat{\beta}_3 t^2$$

such that $\hat{\beta}_1$, $\hat{\beta}_2$, and $\hat{\beta}_3$ are fitted regression coefficients calculated from the data points within the smoothing window centered at time t; that is, the coefficients are calculated from x_{t-m}, x_{t-m+1}, ..., x_t, ..., x_{t+m-1}, x_{t+m}. Local quadratic regression is illustrated in the figure below.



The second modification is to incorporate kernel weighting in to the regression calculation. Having calculated the weights, $w_{-m}, \ldots, w_0, \ldots, w_m$, from a suitable kernel function, the usual regression calculation

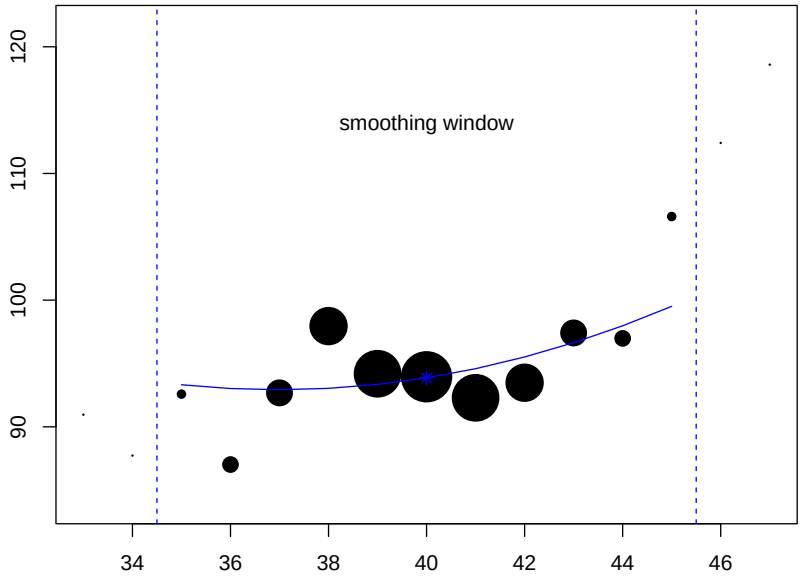
$$\hat{\boldsymbol{\beta}} \;=\; (\boldsymbol{Z}^T\boldsymbol{Z})^{-1}\boldsymbol{Z}^T\boldsymbol{x}$$

for the regression coefficient vector $\hat{\boldsymbol{\beta}} = [\hat{\beta}_1, \hat{\beta}_2, \hat{\beta}_3]^T$ is modified to

$$\hat{\boldsymbol{\beta}} \;=\; (\boldsymbol{Z}^T\boldsymbol{W}\boldsymbol{Z})^{-1}\boldsymbol{Z}^T\boldsymbol{W}\boldsymbol{x},$$

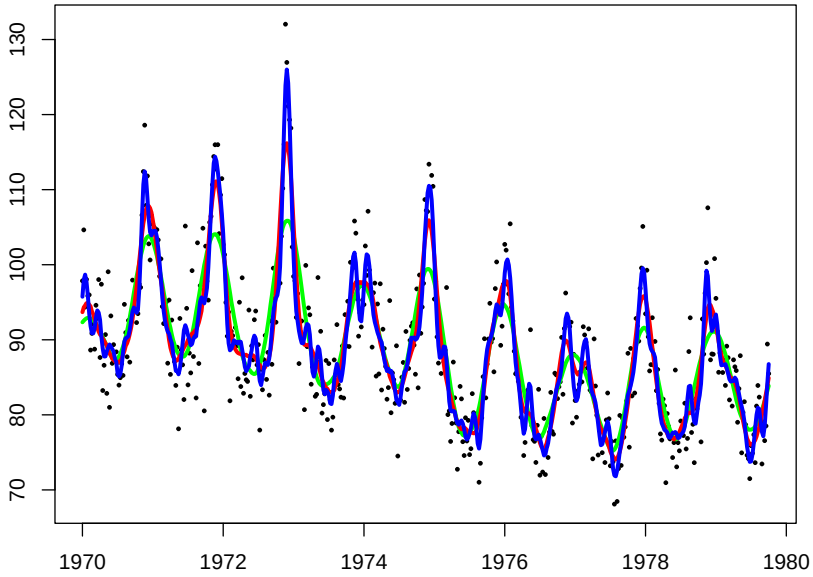
wherein $\boldsymbol{W} = diag(w_{-m}, \ldots, w_0, \ldots, w_m)$ is a diagonal matrix of weights. In either the unweighted or unweighted case, it is sufficient to define $\boldsymbol{Z}$ as the $(2m-1) \times 3$ matrix with its first column all ones, its second column consisting of the values $-m, -m+1, \ldots, 0, \ldots, m-1, m$, and its third column consisting of $m^2, (m-1)^2, \ldots, 0, \ldots, (m-1)^2, m^2$, in which case right-side of the fitted value calculation is shifted from time t to time $t = 0$; that is, $\hat{x}_t = \hat{\beta}_1 + \hat{\beta}_2 t + \hat{\beta}_3 t^2$ simplifies to $\hat{x}_t = \hat{\beta}_1$. Modifying the calculation by incorporating weights gives greater influence on the fitted-value calculation to data values x_{t-j} that are assigned larger weight values, w_j, which are typically those closer in time to the center of the smoothing window.

The figure below offers an illustration of weighted local quadratic regression, having used a Gaussian kernel to define the weights; data values that are assigned a larger weight value are displayed as a larger dot.

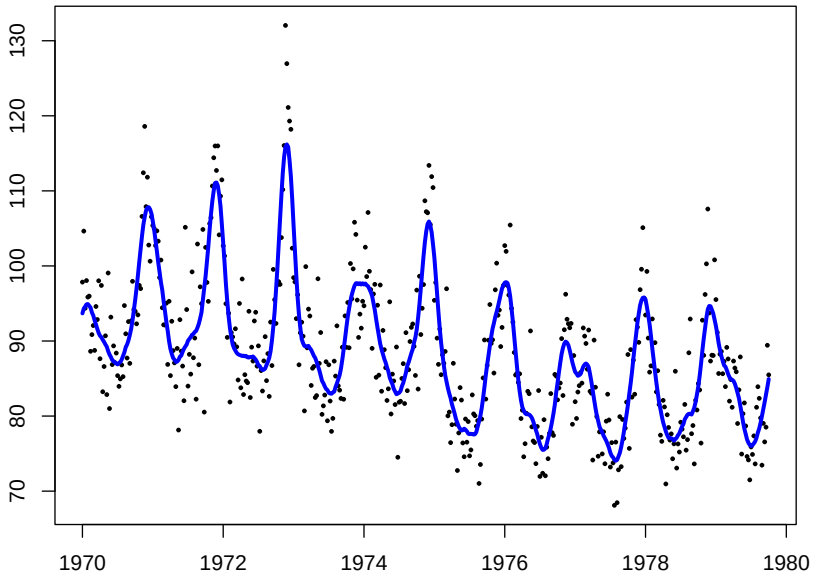


Example: Local quadratic regression of cardiovascular mortality

The results of applying local quadratic regression to the cardiovascular mortality in Los Angeles data set are displayed in the figure below.



The following plot displays fitted values from the calculation with $m = 12$ isolation, colored blue.



This graph does a fairly good job of capturing the overall patterns in the time series, including both long-term trends and cyclic patterns. □

An interesting property of local polynomial regression is that it offers no help beyond moving-average smoothing (weighted or unweighted) when it is configured using *linear* regression, rather than quadratic or higher-order polynomial regression. Within local linear regression, the fitted values are calculated from

$$\hat{x}_t \quad = \quad \hat{\beta}_1 + \hat{\beta}_2 t,$$

where $\hat{\beta}_1$ and $\hat{\beta}_2$ may be calculated from the general formula $\hat{\boldsymbol{\beta}} = (\boldsymbol{Z}^T \boldsymbol{W} \boldsymbol{Z})^{-1} \boldsymbol{Z}^T \boldsymbol{W} \boldsymbol{x}$. It is sufficient to define $\boldsymbol{Z}$ as the $(2m-1) \times 2$ matrix with its first column all ones and its second column consisting of the values $-m, -m+1, \ldots, 0, \ldots, m-1, m$, in which case the

right-hand side of the fitted-value formula $\hat{x}_t = \hat{\beta}_1 + \hat{\beta}_2 t$ shifts from time t to time $t = 0$, and simplifies to $\hat{x}_t = \hat{\beta}_1$.

Alternative formulas for $\hat{\beta}_1$ and $\hat{\beta}_2$, which do not involve vector-matrix notation, are defined from the *weighted mean statistics*

$$\bar{z} = \sum_{j=-m}^{m} w_j z_j \quad \text{and} \quad \bar{x} = \sum_{j=-m}^{m} w_j x_{t-j}$$

having written $z_j = j$, and the *weighted sum-of-squares statistics*

$$S_{zz} = \sum_{j=-m}^{m} w_j (z_j - \bar{z})^2 \quad \text{and} \quad S_{zx} = \sum_{j=-m}^{m} w_j (z_j - \bar{z})(x_j - \bar{x}).$$

The alternative formulas are

$$\hat{\beta}_2 = S_{zx}/S_{zz} \quad \text{and} \quad \hat{\beta}_1 = \bar{x} - \hat{\beta}_2 \bar{z}.$$

In the typical case where the weights are defined from a symmetric kernel, where $K(-j) = K(j)$, the weighted mean of the $z_{-m}, \ldots, 0, \ldots, z_m$ (which is just $-m, \ldots, 0, \ldots, m$) will be zero; that is $\bar{z} = 0$. It follows that the fitted intercept coefficient simplifies to $\hat{\beta}_1 = \bar{x}$, and the fitted-value formula becomes

$$\hat{x}_t = \hat{\beta}_1 = \bar{x} = \sum_{j=-m}^{m} w_j x_{t-j},$$

which exactly matches the fitted-value formula used in weighted moving-average smoothing.

As a final comment on the smoothing techniques discussed in this set of notes (and smoothing methods in general), different techniques may produce similar results, but some techniques capture trends or other systematic patterns a little better than others. When applying a smoothing technique, the selection of tuning parameters, such as smoothing-window width, or configuration of knots in cubic B-spline smoothing, is an influential consideration. As was alluded to in our discussion of smoothing a periodogram, tuning parameters are selected to avoid both under-smoothing and over-smoothing:

- *Under-smoothing* is when the smoothed curve follows the random vacillations in the data without emphasizing an underlying systematic pattern. The curve is "too rough."
- *Over-smoothing* is when the systematic patterns in the data are smoothed out as well as any random patterns. The curve is "too smooth."

For instance, over-smoothing is exhibited in each of the above examples when the tuning parameter is set to emphasize the long-term trend in the mortality data, while "smoothing out" the cyclic seasonal patterns.

In general, smoothing techniques are useful for highlighting patterns in the data, and to some extent for building models. Many were developed outside of the time-series context,

and may not take into account autocorrelations, which is a critical piece of any model for time series data. In that way, they are not always suitable for use in formal inference, but may serve most appropriately in exploratory investigations. Moreover, seemingly optimal settings of the tuning parameter are often chosen to clarify patterns that the analyst identifies as interesting and obvious. In that way, tuning-parameter selection is not automatic, and may be difficult to fully integrate into a framework of formal statistical inference.