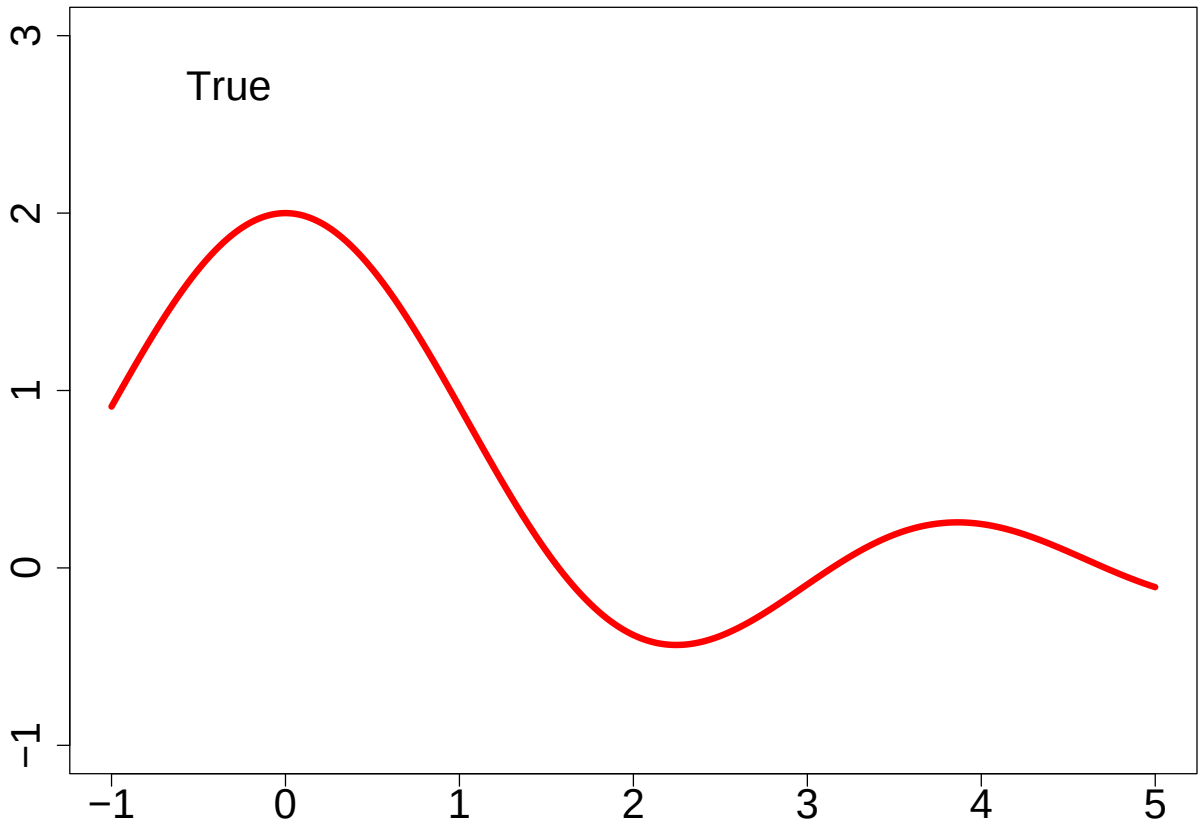
2. Smoothing

In the context of nonparametric regression, a smoothing algorithm is a summary of trend in Y as a function of explanatory variables $X_1, \ldots, X_p$. The smoother takes data and returns a function, called a smooth.

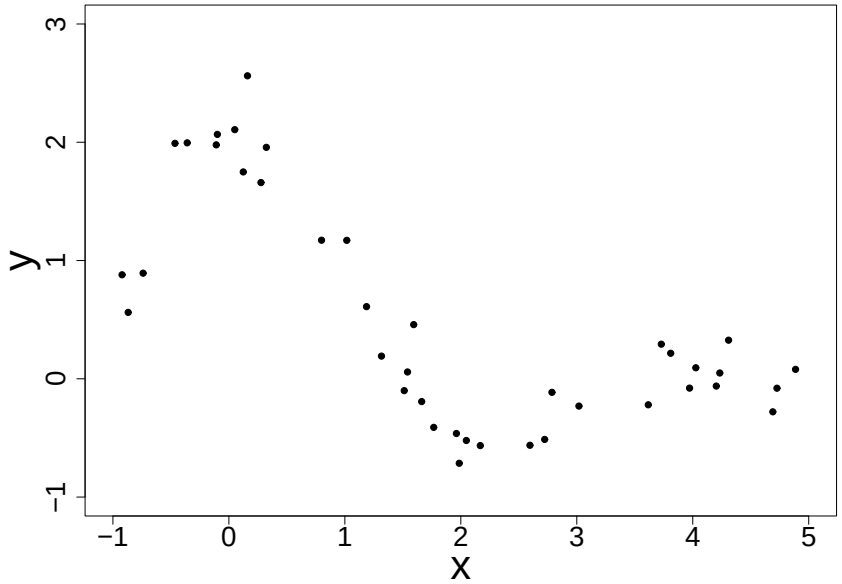
We focus on scatterplot smooths, for which $p = 1$. These usually generalize to $p = 2$ and $p = 3$, but the COD quickly renders them impractical. Even so, they can be used as building blocks for more sophisticated smoothing algorithms that break down less badly with large p, and we discuss several standard smoothers it illustrate the issues and tradeoffs.

Essentially, a smooth just finds an estimate of f in the nonparametric regression function $Y = f(x) + \epsilon$.

As a running example for the next several sections, assume we have data generated from the following function by adding $N(0, .25)$ noise.

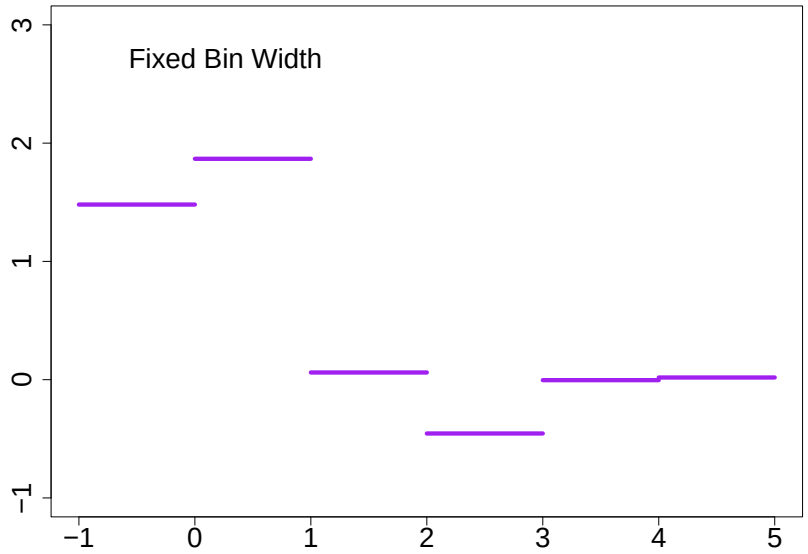


The x values were chosen to be spaced out at the left and right sides of the domain, and the raw data are shown below.



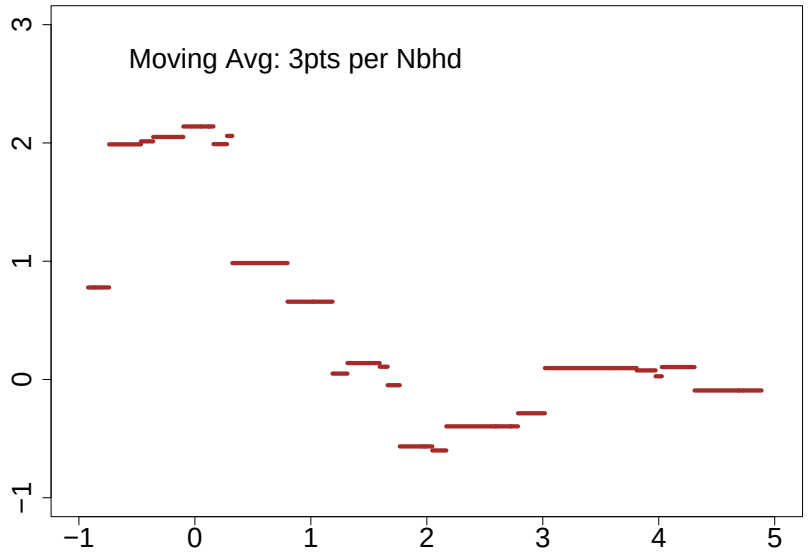
2.1 Bin Smoothing

Here one partitions $\mathbb{R}^p$ into disjoint bins; e.g., for $p = 1$ take $\{[i, i+1), i \in \mathcal{Z}\}$. Within each bin average the Y values to obtain a smooth that is a step function.



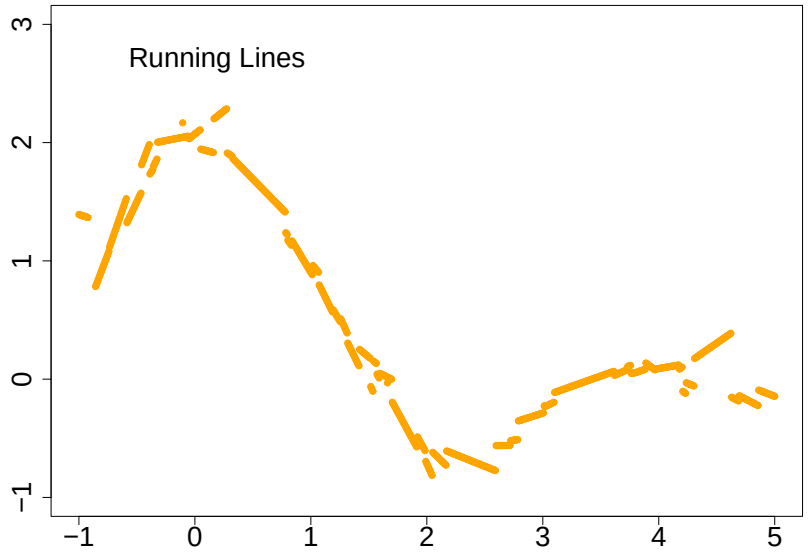
2.2 Moving Averages

Moving averages use variable bins containing a fixed number of observations, rather than fixed-width bins with a variable number of observations. They tend to wiggle near the center of the data, but flatten out near the boundary of the data.



2.3 Running Line

This improves on the moving average by fitting a line rather than an average to the data within a variable-width bin. But it still tends to be rough.



2.4 Loess

Loess was developed by Cleveland (1979; *Journal of the American Statistical Association*, **84**, 829-836).

Loess extends the running line smooth by using weighted linear regression inside the variable-width bins. Loess is more computationally intensive, but is often satisfactorily smooth and flexible.

LOESS fits the model

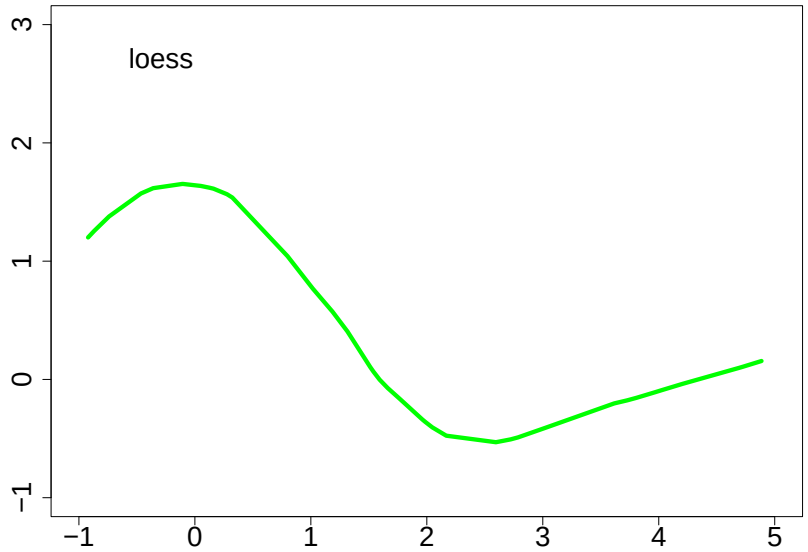
$$\mathbb{E}[Y] = \boldsymbol{\theta}(\boldsymbol{x})'\boldsymbol{x}$$

where

$$\hat{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathbf{argmin}_{\boldsymbol{\theta}\in\mathbb{R}^p} \sum_{i=1}^{n} w(\|\boldsymbol{x} - \boldsymbol{X}_i\|)(Y_i - \boldsymbol{\theta}'\boldsymbol{X}_i)^2$$

and w is a weight function that governs the influence of the ith datum according to the (possibly Mahalanobis) distance of $\boldsymbol{X}_i$ from $\boldsymbol{x}$.

LOESS is a consistent estimator, but may be inefficient at finding relatively simple structures in the data. Although not originally intended for high-dimensional regression, LOESS is often used.



2.5 Kernel Smoothers

These are much like moving averages except that the average is weighted and the bin-width is fixed. Kernel smoothers work well and are mathematically tractable. The weights in the average depend upon the kernel $K(\boldsymbol{x})$. A kernel is usually symmetric, continuous, nonnegative, and integrates to 1 (e.g., a Gaussian kernel).

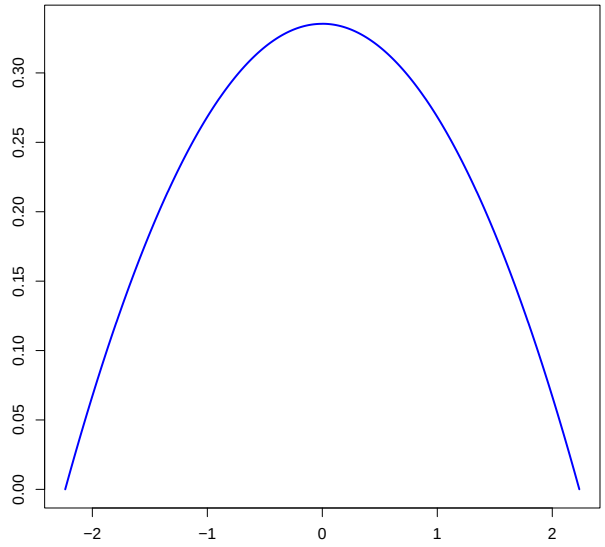
The bin-width is set by h, also called the bandwidth. This can vary, adapting to information in the data on the roughness of the function.

Let $\{(y_i, \boldsymbol{x}_i)\}$ denote the sample. Set $K_h(\boldsymbol{x}) = h^{-1}K(h^{-1}\boldsymbol{x})$. Then the Nadaraya-Watson estimate of f at $\boldsymbol{x}$ is

$$\hat{f}(\boldsymbol{x}) = \frac{\sum_{i=1}^{n} K_h(\boldsymbol{x} - \boldsymbol{x}_i) y_i}{\sum_{i=1}^{n} K_h(\boldsymbol{x} - \boldsymbol{x}_i)}.$$

See Watson (1964; *Theory and Probability Applications*, **10**, 186-190) and Nadaraya (1964; *Sankha A*, **26**, 359-372).

Dor kernel estimation, the Epanechnikov function has good properties (see Silverman; 1986, *Density Estimation for Statistics and Data Analysis*, Chapman & Hall).



The function is

$$K(x) = \frac{3}{4}(1 - x^2) \quad \textbf{on} \quad -1 \leq x \leq 1.$$

2.6 Splines

If one estimates f by minimizing the equation that balances least squares fit with a roughness roughness penalty, e.g.,

$$\min_{f \in \mathcal{F}} \sum_{I=1}^{n} [y_i - f(\boldsymbol{x}_i)]^2 + \lambda \int [f^{(k)}(\boldsymbol{x})]^2 \, d\boldsymbol{x} \tag{1}$$

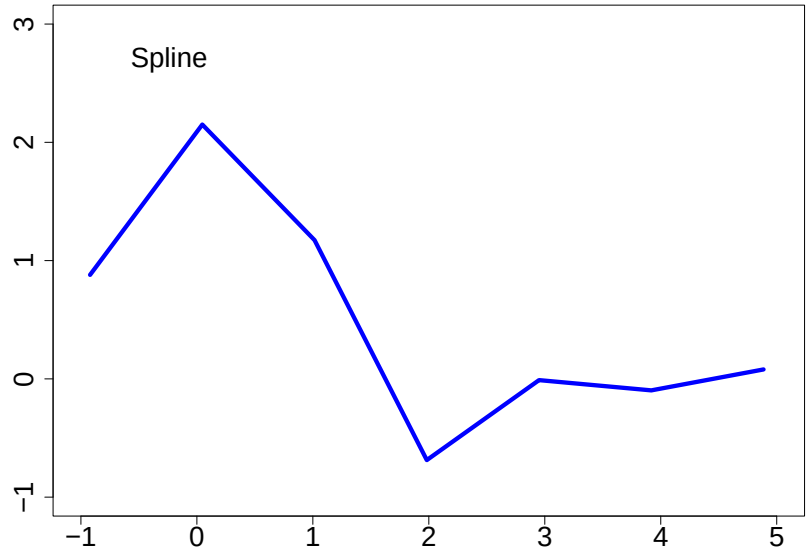
over an appropriate set of functions (e.g., the usual Hilbert space of square-integrable functions), then the solution one obtains are smoothing splines.

Smoothing splines are piecewise polynomials, and the pieces are divided at the sample values $\boldsymbol{x}_i$.

The $\boldsymbol{x}$ values that divide the fit into polynomial portions are called knots. Usually splines are constrained to be smooth across the knots.

Regression splines have fixed knots that need not depend upon the data. But knot selection techniques enable one to find good knots automatically.

Splines are computationally fast, enjoy strong theory, work well, and are widely used.



2.7 Comparing Smoothers

Most smoothing methods are approximately kernel smoothers, with parameters that correspond to the kernel $K(\boldsymbol{x})$ and the bandwidth h.

In practice, one can:

- fix h by judgment,
- find the optimal fixed h,
- fit h adaptively from the data,
- fit the kernel $K(\boldsymbol{x})$ adaptively from the data.

There is a point of diminishing returns, and this is usually hit when one fits the h adaptively.

Silverman (1986; *Density Estimation for Statistics and Data Analysis*, Chapman-Hall) provides a nice discussion of smoothing issues in the context of density estimation.

Breiman and Peters (1992; *International Statistics Review*, **60**, 271-290) give results on a simulation experiment to compare smoothers. Broadly, they found that:

- adaptive kernel smoothing is good but slow,
- smoothing splines are accurate but too rough in large samples,
- everything else is not really competitive in hard problems.

Theoretical understanding of the properties of smoothing depends upon the eigenstructure of the smoothing matrix. Hastie, Tibshirani, and Freidman (2001; *The Elements of Statistical Learning*, Springer) provide an introduction and summary of this area in Chapter 5.

A key issue is the tension between how "wiggly" (or flexible) the smooth can be, and how many observations one has. You need a lot of data to fit local wiggles. This tradeoff is reflected in the degrees of freedom associated with the smooth.

In linear regression one starts with a quantity of information equal to n degrees of freedom where n is the number of independent observations in the sample. Each estimated parameter reduces the information by 1 degree of freedom. This occurs because each estimate corresponds to a linear constraint in $\mathbb{R}^n$, the space in which the n observations lie.

Smoothing is a nonlinear constraint and costs more information. But most smoothers can be expressed as a linear operator (matrix) $\boldsymbol{S}$ acting on the response vector $\boldsymbol{y} \in \mathbb{R}^n$. It turns out that the degrees of freedom lost to smoothing is then $\text{tr}(\boldsymbol{S})$.

In linear regression, the "smoother" is the linear operator that acts on the data to produce the estimate:

$$\hat{\boldsymbol{y}} = \boldsymbol{S}\boldsymbol{Y} = \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

The matrix $\boldsymbol{S}$ is sometimes called the **hat matrix**.

Note that:

$$\begin{aligned} \mathbf{tr}[\boldsymbol{S}] &= \mathbf{tr}[\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T] \\ &= \mathbf{tr}[\boldsymbol{X}^T\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}] \\ &= \mathbf{tr}[\boldsymbol{I}] \\ &= p \end{aligned}$$

since $\boldsymbol{X}$ is $n \times p$ and from matrix algebra we know that $\mathrm{tr}[\boldsymbol{ABC}] = \mathrm{tr}[\boldsymbol{CAB}]$ (see Graybill, *Applied Matrix Algebra in the Statistical Sciences*, 1983, for a proof).

This means that the degrees of freedom associated fitting a linear regression (without an intercept) is p. Thus one has used up an amount of information equivalent to a random sample of size p in fitting the line, leaving an amount of information equivalent to a sample of size $n - p$ with which to estimate dispersion or assess lack of fit.

If one uses an intercept, then this is equivalent to adding a column of ones to $\boldsymbol{X}$, so its dimension is $n \times (p + 1)$ and everything sorts out as usual.

Similarly, we can find the smoothing operator for bin smoothing in $\mathbb{R}^1$. Here one divides a line segment (a, b) into m equally spaced bins, say $B_1, B_2, \ldots, B_m$. Assume there are k_j observations whose x-values line in B_j. Then the bin estimate is:

$$\hat{\boldsymbol{y}} = \boldsymbol{S}\boldsymbol{Y} = \frac{1}{k_j} \sum_{i=1}^{n} Y_i I_{B_j}(X_i).$$

In this case the matrix $\boldsymbol{S}$ is a block matrix whose diagonal blocks are $k_j \times k_j$ submatrices whose every entry is $1/k_j$, and the off-diagonal blocks are matrices of zeroes.

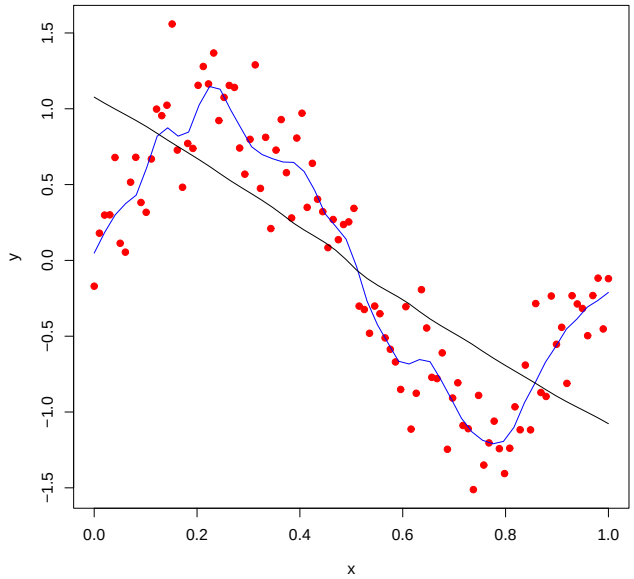
Clearly, the trace of this matrix is m, since the diagonal of each block adds to 1.

Most smoothers are shrinkage estimators. Mathematically, they pull the weights on the coefficients in the basis expansion for $\boldsymbol{y}$ towards zero.

Shrinkage is why smoothing has an effective degrees of freedom between p (as in regression, which does not shrink) and n (which is what one would expect from a naive count of the number of parameters needed to make the function perfectly fit each data point).

For bin smoothing we can oversmooth or undersmooth. If $m << n$, there are few bins and the fitted function has very few jumps (and so is oversmoothed). But if m is large (say equal to n with the bins chosen so that each contains exactly one observation), then the degrees of freedom is n and one has undersmoothed.

Smoothing entails a tradeoff between the bias and variance in $\hat{f}$. If one undersmooths, $\hat{f}$ is wiggly (high variance) but has low bias. If one smooths too much, $\hat{f}$ has small variance but high bias.



Mean squared error is a criterion that captures both aspects. At $\boldsymbol{x}$,

$$\mathbf{MSE}[\hat{f}] = \mathbb{E}[(\hat{f}(\boldsymbol{x}) - f(\boldsymbol{x}))^2] = \mathrm{Var}\,[\hat{f}(\boldsymbol{x})] + \mathbf{bias}^2[\hat{f}(\boldsymbol{x})].$$

One wants a smooth that minimizes $\mathrm{MSE}[\hat{f}(\boldsymbol{x})]$ over all $\boldsymbol{x}$.