Recall that the nonparametric regression model is
\[ Y_i = f(x_i) + \varepsilon_i, \]
where \( f \) is the regression function and the \( \varepsilon_i \) are errors such that \( \mathbb{E}\varepsilon_i = 0. \)

**The Nadaraya-Watson Kernel Estimator**

The Nadaraya-Watson kernel estimator offers what is probably the simplest approach to nonparametric regression. The kernel estimator is an example of a linear smoother. The estimator is linear in the sense that it is given by a linear transformation of the response. Specifically, let \( s(x) = (s_1(x), \ldots, s_n(x))^t \), where
\[ s_i(x) = \frac{w_i(x)}{\sum_{j=1}^n w_j(x)}, \]
where \( w_i(x) = K\{(x - x_i)/h\} \). Now, if \( Y = (Y_1, \ldots, Y_n)^t \), the kernel estimator of \( f(x) \) is
\[ \hat{f}_n(x) = s'(x)Y \]
\[ = \sum_{i=1}^n s_i(x)Y_i \]
\[ = \sum_{i=1}^n \frac{w_i(x)}{\sum_{j=1}^n w_j(x)}Y_i \]
\[ = \sum_{i=1}^n \frac{K\{(x - x_i)/h\}}{\sum_{j=1}^n K\{(x - x_j)/h\}}Y_i. \]

This shows that \( \hat{f}_n(x) \) is a weighted average of the observations, where the weights \( s(x) \) are normalized kernel weights.

This formulation can easily be extended to handle a grid of estimation points \( z = (z_1, \ldots, z_{n_g})^t \). Form the \( n_g \times n \) matrix \( S \), the \( k \)th row of which is \( s'(z_k) \). Then
\[ \hat{f}_n(z) = (\hat{f}_n(z_1), \ldots, \hat{f}_n(z_{n_g}))^t = SY. \]

The matrix \( S \) is called the smoothing matrix. It is analogous to the hat matrix from linear regression.
Theorem 1  The risk (assuming the L^2 loss) of the Nadaraya-Watson kernel estimator is

\[ R(\hat{f}_n, f) = \frac{h^4}{4} \left\{ \int x^2 K(x) dx \right\}^2 \int \left\{ \hat{f}(x) + 2\hat{f}(x) \frac{\hat{g}(x)}{g(x)} \right\}^2 dx \]  

as \( h \to 0 \) and \( nh \to \infty \), where \( g \) is the density from which the \( x_i \) are drawn, and \( \sigma^2 = \sum \varepsilon_i^2 \).

If we set the derivative of (1) equal to zero and solve for \( h \), we get the optimal bandwidth

\[ h_{opt} = n^{-1/5} \left[ \frac{\sigma^2 \int K^2(x) dx \int \frac{1}{g(x)} dx}{\left\{ \int x^2 K(x) dx \right\}^2 \int \left\{ \hat{f}(x) + 2\hat{f}(x) \frac{\hat{g}(x)}{g(x)} \right\}^2 dx} \right]^{1/5}, \]

which implies that \( h_{opt} = O(n^{-1/5}) \). If we plug \( h_{opt} \) into (1), we see that the risk decreases at the rate \( O(n^{-4/5}) \). For most parametric models, the risk of the MLE decreases at the rate \( O(n^{-1}) \). The moral of this story is that we pay a price for using a nonparametric approach. We gain flexibility, but we may sacrifice statistical power to get it.

Local Polynomial Regression

A kernel estimator suffers from design bias (a bias that depends on the distribution of the \( x_i \)) and boundary bias (a bias near the endpoints of the \( x_i \)). These biases can be reduced by using local polynomial regression.

Consider choosing an estimator that minimizes \( \sum_{i=1}^n (Y_i - \beta_0)^2 \). Note that this is equivalent to minimizing the squared length of \( Y - \beta_0 1 \), where 'length' is defined as the ordinary Euclidean norm

\[ \|v\| = \sqrt{n} \sum_{i=1}^n v_i^2, \]

which is in turn defined in terms of the usual inner product, the dot product:

\[ \langle u, v \rangle = u' v = \sum_{i=1}^n u_i v_i. \]

That is, \( \|v\|^2 = \langle v, v \rangle = v' v \).

Recall that the solution to this estimation problem is \( \hat{\beta}_0 = \bar{Y} \). The vector \( \bar{Y} 1 \) is the vector in span\{1\} that is closest to \( Y \) with respect to the ordinary norm. You may also recall that \( \bar{Y} 1 \) is the orthogonal
projection of $Y$ onto span\{1\}, where our notion of perpendiculality is
given by the dot product: $u \perp v$ iff $u'v = 0$. To see this, observe that
the orthogonal projection of $Y$ onto span\{1\} is

$$1(1'1)^{-1}1'Y.$$  

(This is just a special case of $X(X'X)^{-1}X'Y$ from ordinary linear regression.) Now, $1'1 = \sum_{i=1}^{n} 1 = n$, and $1'Y = \sum_{i=1}^{n} Y_i$. Thus

$$1(1'1)^{-1}1'Y = \frac{1}{n} \sum_{i=1}^{n} Y_i = \bar{Y}1.$$  

Now change the scenario slightly by changing the inner product
from $u'v$ to

$$u'W_x v,$$

where $W_x = \text{diag}\{w_i(x)\}$, with $w_i(x) = K\{(x - x_i)/h\}$. The anal-
logous estimation problem is to minimize $\sum_{i=1}^{n} w_i(x)(Y_i - \beta_0)^2$, but
now the relevant projection is orthogonal with respect to this new
inner product. Hence,

$$\hat{\beta}_0 = (1'W_x1)^{-1}1'W_xY.$$  

This implies that

$$\hat{f}(x) = \hat{\beta}_0 = \frac{\sum_{i=1}^{n} w_i(x)Y_i}{\sum_{i=1}^{n} w_i(x)},$$

the kernel estimator. And so we see that the kernel estimator results
from introducing kernel weights in an intercept-only linear model.
The weights ‘localize’ the estimator in the sense that more distant
observations are down-weighted. Since the kernel estimator is local
and uses only an intercept, the kernel estimator is sometimes called a
locally constant estimator.

Local polynomial regression is based on the idea that we might
improve the estimator by using a higher-order polynomial as a local
approximation to $f$. Taylor’s theorem tells us this is a sensible idea.
According to Taylor’s theorem,

$$f(x) \approx f(z) + f^{(1)}(z)(z - x) + \frac{f^{(2)}(z)}{2!}(z - x)^2 + \cdots + \frac{f^{(p)}(z)}{p!}(z - x)^p$$

$$= \beta_0 + \beta_1(z - x) + \beta_2(z - x)^2 + \cdots + \beta_p(z - x)^p$$

$$= P_x(z, \beta)$$

for $z$ in a neighborhood of $x$, where $f^{(m)}$ denotes the $m$th derivative
of $f$. The kernel estimator takes $p = 0$. More generally, local poly-
nomial regression of order $p$ minimizes

$$\sum_{i=1}^{n} w_i(x)\{Y_i - P_x(x_i, \beta)\}^2.$$  \hfill (2)
This yields the local estimate

$$\hat{f}_n(x) = P_x(x, \hat{\beta}) = \hat{\beta}_0(x).$$

Note that the minimizer of (2) is

$$\hat{\beta}(x) = (X_x' W_x X_x)^{-1} X_x' W_x Y,$$

where

$$X_x = \begin{pmatrix} 1 & x - x_1 & (x_1 - x)^2 & \cdots & \frac{(x_1 - x)^p}{p!} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n - x & (x_n - x)^2 & \cdots & \frac{(x_n - x)^p}{p!} \end{pmatrix}.$$ 

This implies that \( \hat{f}_n(x) = \hat{\beta}_0(x) \) is the inner product of \( Y \) with the first row of \( (X_x' W_x X_x)^{-1} X_x' W_x Y \), and so \( \hat{f}_n(x) \) is a linear smoother. The estimator has mean and variance

$$\mathbb{E} \hat{f}_n(x) = s(x)f(x),$$

$$\mathbb{V} \hat{f}_n(x) = \sigma^2 \|s(x)\|^2,$$

where \( s(x) \) is the first row of \( (X_x' W_x X_x)^{-1} X_x' W_x Y \) and \( f(x) = (f(x_1), \ldots, f(x_n))' \).

**Why \( p \) Should Be Odd**

The case \( p = 1 \) is called local linear regression. Local linear regression eliminates design bias and alleviates boundary bias.

**Theorem 2** Let \( Y_i = f(X_i) + \sigma(X_i) \varepsilon_i \) for \( i \in \{1, \ldots, n\} \) and \( X_i \in [a, b] \). Assume that the \( X_i \) were drawn from density \( g \). Suppose that \( g \) is positive; \( g, f, \) and \( \sigma \) are continuous in a neighborhood of \( x \); and \( h \to 0 \) and \( nh \to \infty \). Let \( x \in (a, b) \). Then the local constant estimator and the local linear estimator both have variance

$$\frac{\sigma^2(x)}{g(x)nh} \int K^2(u)du + o \left( \frac{1}{nh} \right).$$

The local constant estimator has bias

$$h^2 \left( \frac{1}{2} \hat{f}(x) + \frac{\hat{f}(x) \hat{g}(x)}{g(x)} \right) \int u^2 K(u)du + o(h^2),$$

and the local linear estimator has bias

$$h^2 \frac{1}{2} \hat{f}(x) \int u^2 K(u)du + o(h^2).$$

At the endpoints of \([a, b]\), the local constant estimator has bias of order \( h \), and the local linear estimator has bias of order \( h^2 \).

More generally, let \( p \) be even. Then local polynomial regression of order \( p + 1 \) reduces design bias and boundary bias relative to local polynomial regression of order \( p \), without increasing the variance.
Variance Estimation

Homoscedasticity

Until the previous theorem we had been assuming homoscedasticity, i.e., \( Y_i = f(x_i) + \sigma \varepsilon_i \) for all \( i \), where \( \forall \varepsilon_i = 1 \). In this case, we can estimate \( \sigma^2 \) in a simple and familiar way, namely, as the sum of the squared residuals divided by the residual degrees of freedom. More specifically, the estimator is

\[
\hat{\sigma}^2 = \frac{\sum_{i=1}^{n} (Y_i - \hat{f}_n(x_i))^2}{n - 2\nu + \tilde{\nu}}
\]

where \( \nu = \text{tr}(S) \) and \( \tilde{\nu} = \text{tr}(S'S) = \sum_i \|s(x_i)\|^2 \). Recall that \( S \) is the smoothing matrix.

The estimator \( \hat{\sigma}^2 \) is consistent for \( \sigma^2 \). To see this, first observe that \( e = Y - SY = (I - S)Y \), which implies that

\[
\hat{\sigma}^2 = \frac{Y'AY}{\text{tr}(\Lambda)'}
\]

where \( \Lambda = (I - S)'(I - S) \). A well-known fact about quadratic forms is

\[
\mathbb{E}Y'AY = \text{tr}(\Lambda \Sigma) + \mu'\Lambda \mu,
\]

where \( \Sigma = \mathbb{V}Y \) and \( \mu = \mathbb{E}Y \). Thus

\[
\mathbb{E}\hat{\sigma}^2 = \frac{\mathbb{E}Y'AY}{\text{tr}(\Lambda)'}
\]

\[
= \frac{\text{tr}(\Lambda \sigma^2 I)}{\text{tr}(\Lambda)} + \frac{f'\Lambda f}{n - 2\nu + \tilde{\nu}}
\]

\[
= \sigma^2 + \frac{f'\Lambda f}{n - 2\nu + \tilde{\nu}}.
\]

Under mild conditions, the second term in (3) will go to zero as \( n \to \infty \).

The appearance of \( n - 2\nu + \tilde{\nu} \) may seem mysterious, but this quantity is in fact analogous to the residual degrees of freedom \( n - p \) in ordinary linear regression. In that setting, \( n - p = \text{tr}(I - H) = \text{tr}\{(I - H)'(I - H)\} \), where \( H \) is the hat matrix. In the current setting,
\(I - H\) is replaced by \(I - S\), and we have

\[
\text{tr}\{ (I - S)'(I - S) \} = \text{tr}(I'I - I'S - S'I + S'S)
= \text{tr}(I - S - S' + S'S)
= \text{tr}(I) - \text{tr}(S) - \text{tr}(S') + \text{tr}(S'S)
= n - 2\text{tr}(S) + \text{tr}(S'S)
= n - 2\nu + \tilde{\nu}.
\]

Heteroscedasticity

Now suppose that \(Y_i = f(x_i) + \sigma(x_i)\varepsilon_i\). Since this implies that \(\sigma\) is a (presumably non-constant) function, estimating it requires a second regression. The second regression is for the model

\[
Z_i = \log\{Y_i - f(x_i)\}^2
= \log\sigma^2(x_i)\varepsilon_i^2
= \log\sigma^2(x_i) + \log\varepsilon_i^2
= \log\sigma^2(x_i) + \delta_i.
\]

This model suggests that we could estimate \(\log\sigma^2(x)\) by doing a regression with the log squared residuals from the first regression as the response. Specifically, we do the following.

1. Estimate \(f(x)\) to arrive at \(\hat{f}_n(x)\).
2. Let \(Z_i = \log\{Y_i - \hat{f}_n(x_i)\}^2\).
3. Regress the \(Z_i\) on the \(x_i\) to get an estimate \(\hat{g}(x)\) of \(\log\sigma^2(x)\).
4. Let \(\hat{\sigma}^2(x) = \exp\hat{g}(x)\).

Confidence Bands

We would of course like to construct confidence bands for \(f\). A confidence interval for \(f(x)\) usually has the form

\[
\hat{f}_n(x) \pm c\text{se}(x),
\]

where \(c > 0\) is a constant and \(\text{se}(x)\) is an estimate of the standard deviation of \(\hat{f}_n(x)\). Perhaps counterintuitively, such a confidence interval is not truly an interval for \(f(x)\), but is instead an interval for

\[
\hat{f}_n(x) = \mathbb{E}\hat{f}_n(x) = s(x)f(x).
\]

This is because there is a bias that does not disappear as the sample size becomes large.
Let $s_n(x)$ be the standard deviation of $\hat{f}_n(x)$. Then

$$\frac{\hat{f}_n(x) - f(x)}{s_n(x)} = \frac{\hat{f}_n(x) - \bar{f}_n(x)}{s_n(x)} + \frac{\bar{f}_n(x) - f(x)}{s_n(x)} = Z_n(x) + \frac{\text{bias}\{\hat{f}_n(x)\}}{\sqrt{V\hat{f}_n(x)}}.$$

Typically, $Z_n(x) \Rightarrow \mathcal{N}(0, 1)$. In a nonparametric setting, the second term does not go to zero as the sample size increases. This means the bias is present in the limit, which implies that the resulting confidence interval is not centered around $f(x)$. We might respond to this by

1. accepting that our confidence interval is for $\bar{f}_n(x)$ rather than $f(x)$;
2. attempting to correct the bias by estimating the bias function $f_n(x) - f(x)$; or
3. minimizing the bias by undersmoothing.

The second option is perhaps the most tempting but is considerably more difficult than estimating $f(x)$ since the bias involves $\hat{f}(x)$. This fact makes the first and third options more appealing. Most people go with the first option because it is difficult to choose the right amount of undersmoothing.

**Pointwise Bands**

We can construct a pointwise band by invoking asymptotic normality or by using the bootstrap. In the former case, the interval is

$$\hat{f}_n(x) \pm \Phi^{-1}(1 - \alpha/2)se(x).$$

As for the bootstrap, how we should resample depends on whether we assume homoscedasticity. If we do assume constant variance, i.e., $\sigma(x) = \sigma$, the $k$th bootstrap dataset is

$$Y_i^{(k)} = \hat{f}_n(x_i) + e_i^{(k)} \quad (i = 1, \ldots, n),$$

where $e^{(k)} = (e_1^{(k)}, \ldots, e_n^{(k)})'$ is a sample (with replacement) of size $n$ from the vector of residuals $e = (Y_1 - \hat{f}_n(x_1), \ldots, Y_n - \hat{f}_n(x_n))'$. The endpoints of the resulting interval at $x_i$ are the $\alpha/2$ and $1 - \alpha/2$ quantiles of the bootstrap sample $\hat{f}_n^{(1)}(x_i), \ldots, \hat{f}_n^{(b)}(x_i)$.

If we assume that $\sigma(x)$ is a non-constant function, we can still do a bootstrap, but we must modify the resampling procedure. Here is the algorithm in detail.

1. Estimate $\sigma(x_i)$ to arrive at $\hat{\sigma}(x_i)$ for $i \in \{1, \ldots, n\}$. 
2. Studentize the vector of residuals \((Y_1 - \hat{f}_n(x_1), \ldots, Y_n - \hat{f}_n(x_n))'\) by dividing the \(i\)th element by \(\hat{\sigma}(x_i)\):

\[ e_i = \frac{Y_i - \hat{f}_n(x_i)}{\hat{\sigma}(x_i)}. \]

3. Compute the \(k\)th bootstrap dataset as

\[ Y_i^{(k)} = f_n^{(k)}(x_i) + \hat{\sigma}(x_i) e_i^{(k)} \quad (i = 1, \ldots, n), \]

where \(e^{(k)} = (e_1^{(k)}, \ldots, e_n^{(k)})'\) is a sample (with replacement) of size \(n\) from the vector of Studentized residuals.

4. Compute \(f_n^{(k)}(x) = SY^{(k)}\) for \(k \in \{1, \ldots, b\}\).

5. The endpoints of the confidence interval at \(x_i\) are again the \(\alpha/2\) and \(1 - \alpha/2\) quantiles of the bootstrap sample \(f_n^{(1)}(x_i), \ldots, f_n^{(b)}(x_i)\).

**Simultaneous Bands**

To construct a simultaneous band we use the so-called tube formula. Suppose that \(\sigma\) is known, and let \(I(x)\) be an interval. Then

\[ \mathbb{P}\{\hat{f}_n(x) \in I(x) \text{ for some } x \in [a, b]\} = \mathbb{P}\left( \max_{x \in [a, b]} \frac{|\hat{f}_n(x) - \hat{f}_n(x)|}{\sigma\|s(x)\|} > c \right) \]

\[ = \mathbb{P}\left( \max_{x \in [a, b]} \frac{\sum Z_i T_i(x)}{\sigma\|s(x)\|} > c \right) \]

\[ = \mathbb{P}\left( \max_{x \in [a, b]} |W(x)| > c \right), \]

where

\[ W(x) = \sum_{i=1}^n Z_i T_i(x), \]

\[ Z_i = \frac{e_i}{\sigma} \sim N(0, 1), \]

\[ T_i(x) = s_i(x)/\|s(x)\|. \]

It turns out that

\[ \mathbb{P}\left( \max_{x} |W(x)| > c \right) \approx 2\{1 - \Phi(c)\} + \frac{\kappa}{\pi} \exp(-c^2/2) \]

for large \(c\), where

\[ \kappa = \int_a^b \|T(x)\|dx, \]

where \(T(x) = (T_1(x), \ldots, T_n(x))'\). Choosing \(c\) to solve

\[ 2\{1 - \Phi(c)\} + \frac{\kappa}{\pi} \exp(-c^2/2) = \alpha \]

yields the desired band \(\hat{f}_n(x) \pm c se(x)\).
Choosing the Right Bandwidth

We want to choose $h$ to minimize the risk

$$R(h) = \mathbb{E} \left( \frac{1}{n} \sum_{i=1}^{n} \left( f_n(x_i) - f(x_i) \right)^2 \right).$$

Since $R(h)$ depends on the unknown function $f$, we will instead minimize an estimate $\hat{R}(h)$ of $R(h)$.

It might seem sensible to estimate $R(h)$ using

$$\hat{R}(h) = \frac{1}{n} \sum_{i=1}^{n} \left( Y_i - \hat{f}_n(x_i) \right)^2,$$

the so-called training error. But this estimator is biased downward and usually leads to undersmoothing. A better risk estimator is the leave-one-out cross-validation score:

$$\text{CV}(h) = \hat{R}(h) = \frac{1}{n} \sum_{i=1}^{n} \left( Y_i - \hat{f}_{-i}(x_i) \right)^2,$$

where $\hat{f}_{-i}(x_i)$ is the estimate obtained by leaving out the $i$th observation. Intuitively, we are asking, "How well can we predict $Y_i$ if we do not use $Y_i$ in the estimation procedure?" For linear smoothers, computing this score is not as burdensome as it may seem because we do not have to recompute the estimate with each observation left out. Instead, we have

$$\text{CV}(h) = \hat{R}(h) = \frac{1}{n} \sum_{i=1}^{n} \left( Y_i - \hat{f}_n(x_i) \right)^2 \frac{1}{1 - S_{ii}},$$

where $S_{ii}$ is the $i$th diagonal element of $S$.

An alternative is the generalized cross-validation score:

$$\text{GCV}(h) = \frac{1}{n} \sum_{i=1}^{n} \left( Y_i - \hat{f}_n(x_i) \right)^2 \frac{1}{1 - \frac{1}{n} \text{tr}(S)},$$

which replaces the $S_{ii}$ with their average. Usually CV and GCV lead to bandwidths that are close to one another.