Outline

1. Density estimation
2. Kernel regression
3. Splines
4. Additive models
5. Functional data analysis
2. Kernel regression

1. Some examples
2. Kernel averaging
3. Bandwidth selection
4. Local polynomial estimation
5. Derivative estimation
6. Inference
7. Multidimensional smoothing
Outline

1. Some examples
2. Kernel averaging
3. Bandwidth selection
4. Local polynomial estimation
5. Derivative estimation
6. Inference
7. Multidimensional smoothing
Pulp prices

![Graph showing the relationship between world pulp price and pulp shipments. The x-axis represents the world pulp price, ranging from 500 to 850, and the y-axis represents pulp shipments, ranging from 10 to 35.]
Pulp prices

![Graph showing the relationship between world pulp price and pulp shipments. The graph includes a scatter plot with a linear trend line. The x-axis represents world pulp price, ranging from 500 to 850, and the y-axis represents pulp shipments, ranging from 5 to 35.](image-url)
Pulp prices

![Graph showing residuals versus world pulp price.](image-url)
Pulp prices

![Pulp price graph](image-url)
Age-earnings profile

Age–earnings profile: merchant marines (1861–1912)
Age-earnings profile

Age–earnings profile: merchant marines (1861–1912)
Outline

1. Some examples
2. **Kernel averaging**
3. Bandwidth selection
4. Local polynomial estimation
5. Derivative estimation
6. Inference
7. Multidimensional smoothing
The problem

Response variable: \( y_1, \ldots, y_n \)
Explanatory variable: \( x_1, \ldots, x_n \)

\[ y_j = r(x_j) + e_j, \quad \mathbb{E}(e_j) = 0 \]
The problem

Response variable: \( y_1, \ldots, y_n \)
Explanatory variable: \( x_1, \ldots, x_n \)

\[ y_j = r(x_j) + e_j, \quad E(e_j) = 0 \]

• If \( r(x) \) of known form (e.g., \( r(x) = a + bx \)), use LS estimation.
The problem

Response variable: \( y_1, \ldots, y_n \)
Explanatory variable: \( x_1, \ldots, x_n \)

\[ y_j = r(x_j) + e_j, \quad \mathbb{E}(e_j) = 0 \]

- If \( r(x) \) of known form (e.g., \( r(x) = a + bx \)), use LS estimation.
- If \( r(x) \) of unknown form, use regression smoother.
The problem

Response variable: \( y_1, \ldots, y_n \)
Explanatory variable: \( x_1, \ldots, x_n \)

\[ y_j = r(x_j) + e_j, \quad E(e_j) = 0 \]

- If \( r(x) \) of known form (e.g., \( r(x) = a + bx \)), use LS estimation.
- If \( r(x) \) of unknown form, use regression smoother.
The problem

Response variable: \( y_1, \ldots, y_n \)
Explanatory variable: \( x_1, \ldots, x_n \)

\[
y_j = r(x_j) + e_j, \quad \text{E}(e_j) = 0
\]

- If \( r(x) \) of known form (e.g., \( r(x) = a + bx \)), use LS estimation.
- If \( r(x) \) of unknown form, use regression smoother.

Assume:

- \( e_j \) independent and identically distributed;
The problem

Response variable: \( y_1, \ldots, y_n \)
Explanatory variable: \( x_1, \ldots, x_n \)

\[ y_j = r(x_j) + e_j, \quad \text{E}(e_j) = 0 \]

- If \( r(x) \) of known form (e.g., \( r(x) = a + bx \)), use LS estimation.
- If \( r(x) \) of unknown form, use regression smoother.

Assume:
- \( e_j \) independent and identically distributed;
- \( r \) is ‘smooth’ function;
The problem

Response variable: $y_1, \ldots, y_n$
Explanatory variable: $x_1, \ldots, x_n$

$$y_j = r(x_j) + e_j, \quad \mathbb{E}(e_j) = 0$$

- If $r(x)$ of known form (e.g., $r(x) = a + bx$), use LS estimation.
- If $r(x)$ of unknown form, use regression smoother.

Assume:

- $e_j$ independent and identically distributed;
- $r$ is ‘smooth’ function;
- $x_j$ values are non-stochastic.
**Kernel smoothers**

**Idea:** Estimate $r(x_j)$, by averaging nearby values of \( \{y_i\} \).

\[
\hat{r}(x) = \sum_{j=1}^{n} w_j(x) y_j \quad \text{where} \quad \sum_{j=1}^{n} w_j(x) = 1.
\]
Kernel smoothers

**Idea:** Estimate \( r(x_j) \), by averaging nearby values of \( \{y_i\} \).

\[
\hat{r}(x) = \sum_{j=1}^{n} w_j(x) y_j \quad \text{where} \quad \sum_{j=1}^{n} w_j(x) = 1.
\]

- If \( w_j(x) = w \) for all \( j \) such that \( |x - x_j| < h \), then \( \hat{r}(x) \) is called a “moving average”.
Kernel smoothers

**Idea:** Estimate \( r(x_j) \), by averaging nearby values of \( \{y_i\} \).

\[
\hat{r}(x) = \sum_{j=1}^{n} w_j(x) y_j \quad \text{where} \quad \sum_{j=1}^{n} w_j(x) = 1.
\]

- If \( w_j(x) = w \) for all \( j \) such that \(|x - x_j| < h\), then \( \hat{r}(x) \) is called a “moving average”.
- Weighted averages give a smoother smoother.
Kernel smoothers

Idea: Estimate $r(x_j)$, by averaging nearby values of $\{y_i\}$.

$$\hat{r}(x) = \sum_{j=1}^{n} w_j(x) y_j \quad \text{where} \quad \sum_{j=1}^{n} w_j(x) = 1.$$

- If $w_j(x) = w$ for all $j$ such that $|x - x_j| < h$, then $\hat{r}(x)$ is called a “moving average”.
- Weighted averages give a smoother smoother.
- If the weights are defined using a kernel function, then $\hat{r}(x)$ is called a kernel smoother.
Kernel weights

We shall assign weights using a kernel function:

\[ w_j(x) = cK \left( \frac{x_i - x}{h} \right) \]

where \( c \) is a constant to ensure weights sum to 1.
Kernel weights

We shall assign weights using a kernel function:

\[ w_j(x) = c K \left( \frac{x_i - x}{h} \right) \]

where \( c \) is a constant to ensure weights sum to 1. Substituting gives

\[ \sum_{j=1}^{n} c K \left( \frac{x_i - x}{h} \right) = 1 \]

or

\[ c = \left[ \sum_{j=1}^{n} K \left( \frac{x_i - x}{h} \right) \right]^{-1} \]

giving

\[ \hat{r}(x) = \frac{\sum_{j=1}^{n} K \left( \frac{|x_j - x|}{h} \right) y_j}{\sum_{j=1}^{n} K \left( \frac{|x_j - x|}{h} \right)} \]
Kernel smoother

![Graph showing the relationship between world pulp price and pulp shipments. The graph includes a scatter plot with data points and a smooth curve indicating the trend.]
Kernel smoother

The diagram illustrates a scatter plot of world pulp price vs. pulp shipments. Points are plotted on the vertical axis, representing pulp shipments, and the horizontal axis represents world pulp price. A blue line through the data points represents the kernel smoother, which is a method used for nonparametric smoothing in statistics. The kernel smoother helps to identify patterns and trends in the data without assuming a specific functional form.
Kernel smoother

The diagram illustrates the relationship between world pulp price and pulp shipments using a kernel smoother. The kernel smoother is a nonparametric method used to estimate the underlying distribution of data. The data points are represented by circles, and the smoother line represents the estimated trend or density of the data.

The x-axis represents the world pulp price, ranging from 500 to 850 units, while the y-axis represents pulp shipments, ranging from 10 to 35 units. The kernel smoother curve is centered around the mid-range of the pulp price, indicating a peak in shipments at that price level.
Kernel smoother

Graph showing the relationship between world pulp price and pulp shipments.
Kernel smoother

The diagram shows a scatter plot with the world pulp price on the x-axis and pulp shipments on the y-axis. The data points are plotted, and a horizontal line is drawn at a certain level, indicating a smoothing effect of the kernel smoother.
Kernel smoother

World pulp price vs Pulp shipments

500 550 600 650 700 750 800 850
10 15 20 25 30 35

- The graph shows the relationship between world pulp price and pulp shipments.
- The data points are scattered, indicating variability in the relationship.
- A kernel smoother line is drawn to illustrate the trend.
Kernel smoother

![Graph showing the relationship between world pulp price and pulp shipments. The graph includes a scatter plot with marked data points and a trend line. The x-axis represents world pulp price, ranging from 500 to 850, while the y-axis represents pulp shipments, ranging from 10 to 35. The trend line indicates a positive correlation between the two variables.]
Kernel smoother

![Graph showing the relationship between world pulp price and pulp shipments. The graph displays a scatter plot with world pulp price on the x-axis and pulp shipments on the y-axis.]
Kernel smoother

![Graph showing the relationship between world pulp price and pulp shipments. The x-axis represents world pulp price with values from 500 to 850. The y-axis represents pulp shipments with values from 10 to 35. The graph displays a scatter plot with data points.](image-url)
Kernel smoother

![Graph showing the relationship between world pulp price and pulp shipments.]
Kernel smoother

World pulp price vs Pulp shipments
Kernel smoother

![Kernel smoother](image)

- **World pulp price**
  - 500
  - 550
  - 600
  - 650
  - 700
  - 750
  - 800
  - 850

- **Pulp shipments**
  - 10
  - 15
  - 20
  - 25
  - 30
  - 35
Kernel smoother

The diagram illustrates the relationship between world pulp price and pulp shipments. The x-axis represents the world pulp price, ranging from 500 to 850. The y-axis represents pulp shipments, ranging from 10 to 35. The data points are scattered across the graph, showing fluctuations in both price and shipments.
Kernel smoother

The diagram shows a scatter plot of world pulp price on the x-axis and pulp shipments on the y-axis. The data points are plotted as circles, with a trend line indicating the relationship between the two variables.
Kernel smoother

World pulp price

Pulp shipments
Kernel smoother

- Plot of World pulp price vs. Pulp shipments
- The data points are scattered across the graph
- A horizontal line indicates a baseline or trend in Pulp shipments
- The graph shows a relationship between World pulp price and Pulp shipments
Kernel smoother

![Graph showing world pulp price vs pulp shipments]
Kernel smoother

The graph shows the relationship between world pulp price and pulp shipments. The data points are scattered, indicating a potential non-linear relationship. A smoothing method, such as a kernel smoother, could be used to identify trends and patterns in the data.
Kernel smoother

![Graph showing the relationship between world pulp price and pulp shipments.](image-url)
Kernel smoother

- A graph showing the relationship between the World pulp price and Pulp shipments.
- The graph illustrates a downward trend as the world pulp price increases.
- The scatter plot displays individual data points, and the smooth line indicates a nonparametric smoothing method, possibly a kernel smoother, to estimate the underlying trend.

**Graph Details:**
- X-axis: World pulp price
- Y-axis: Pulp shipments
- Data points and a smoothing curve indicate the effect of pulp price on shipments.
Bandwidth considerations

- $h$ is sometimes called the \textit{window width} or the \textit{bandwidth};
Bandwidth considerations

- $h$ is sometimes called the *window width* or the *bandwidth*;
- $\hat{r}(x)$ is smoother for larger $h$. 

If the design points are unequally spaced, the number of points in the average will vary with $x$. Let $\Delta_j(x) = |x_j - x|$ be the distance from $x$ to $x_j$. If the kernel has finite support on $[-1, 1]$, then for $x_j$ such that $\Delta_j(x) < h$, the weights are positive and decrease as $\Delta_j(x)$ increases and for $\Delta_j(x) \geq h$, the weights are zero.
Bandwidth considerations

- $h$ is sometimes called the *window width* or the *bandwidth*;
- $\hat{r}(x)$ is smoother for larger $h$.
- If the design points are unequally spaced, the number of points in the average will vary with $x$. 

Bandwidth considerations

- \( h \) is sometimes called the *window width* or the *bandwidth*;
- \( \hat{r}(x) \) is smoother for larger \( h \).
- If the design points are unequally spaced, the number of points in the average will vary with \( x \).
- Let \( \Delta_j(x) = |x_j - x| \) be the distance from \( x \) to \( x_j \). If the kernel has finite support on \([-1, 1]\), then for \( x_j \) such that \( \Delta_j(x) < h \), the weights are positive and decrease as \( \Delta_j(x) \) increases and for \( \Delta_j(x) \geq h \), the weights are zero.
Kernel regression

This is also called the **Nadaraya–Watson** estimator after its two discoverers who proposed the estimator independently in 1964.

- Observations $y_j$ obtain more weight in areas where the corresponding $x_j$ are sparse.
Kernel regression

This is also called the Nadaraya–Watson estimator after its two discoverers who proposed the estimator independently in 1964.

- Observations $y_j$ obtain more weight in areas where the corresponding $x_j$ are sparse.
- If the denominator equals zero, then so does the numerator and the estimate is undefined.
Kernel regression

This is also called the **Nadaraya–Watson** estimator after its two discoverers who proposed the estimator independently in 1964.

- Observations $y_j$ obtain more weight in areas where the corresponding $x_j$ are sparse.
- If the denominator equals zero, then so does the numerator and the estimate is undefined.
- If $h \to 0$, then $w_j(x) \to 1$ if $x = x_i$ and zero elsewhere. Hence $\hat{r}(x_i) \to y_i$ and leads to an interpolation of the data.
Kernel regression

This is also called the Nadaraya–Watson estimator after its two discoverers who proposed the estimator independently in 1964.

- Observations $y_j$ obtain more weight in areas where the corresponding $x_j$ are sparse.
- If the denominator equals zero, then so does the numerator and the estimate is undefined.
- If $h \to 0$, then $w_j(x) \to 1$ if $x = x_i$ and zero elsewhere. Hence $\hat{r}(x_i) \to y_i$ and leads to an interpolation of the data.
- For $h \to \infty$, $w_j(x)$, converges to $1/n$ for all values of $x$. Hence the total estimate converges to the constant function $\bar{y}$. 
Bias

\[ E[\hat{r}(x)] = E \left[ \sum_{j=1}^{n} w_j(x)y_j \right] \]

\[ = \sum_{j=1}^{n} w_j(x)E[r(x_j) + e_j] = \sum_{j=1}^{n} w_j(x)r(x_j). \]

Taking a Taylor’s series expansion of \( r(x_j) \) about \( x \) gives the following expression for the bias of \( \hat{m}(x) \):

\[ E[\hat{r}(x)] - r(x) = r'(x) \sum_{j=1}^{n} w_j(x)(x_j - x) + \frac{r''(x)}{2} \sum_{j=1}^{n} w_j(x)(x_j - x)^2 + R \]

where \( R \) is small under some regularity conditions.
Bias

Substantial bias under the following conditions:

- on the boundary of the predictor space because the asymmetry of the kernel neighbourhood causes the first term to be large when \( r'(x) \) is large.
**Bias**

Substantial bias under the following conditions:

- on the boundary of the predictor space because the asymmetry of the kernel neighbourhood causes the first term to be large when \( r'(x) \) is large.

- in the interior if the true mean function has substantial curvature (if \( |r''(x)| \) is large)
Substantial bias under the following conditions:

- on the boundary of the predictor space because the asymmetry of the kernel neighbourhood causes the first term to be large when $r'(x)$ is large.
- in the interior if the true mean function has substantial curvature (if $|r''(x)|$ is large)
- in the interior if the design points are very irregularly spaced (again giving some asymmetric neighbourhoods).
Bias

Hence, kernel smoothers tend to flatten out trends near the endpoints and flatten peaks and troughs. If the \( \{x_i\} \) are symmetric around \( x \) and \( w_j(x) \) is symmetric around \( x \), then for fixed \( x_i \) we obtain

\[
\mathbb{E}[\hat{r}(x)] - r(x) \approx \frac{r''(x)}{2} \sum_{j=1}^{n} w_j(x)(x - x_j)^2.
\]
Variance

\[
V[\hat{r}(x)] = V\left[ \sum_{j=1}^{n} w_j(x)y_j \right] \\
= V\left[ \sum_{j=1}^{n} w_j(x)r(x_j) + e_j \right] \\
= \sum_{j=1}^{n} w_j^2(x)V(e_j) \\
= \sigma^2 \sum_{j=1}^{n} w_j^2(x). 
\]
Asymptotic results

For constant bandwidth kernel smoothers, as $n \to \infty$ and $\sup |x_{j+1} - x_j| \to 0$,

$$E[\hat{r}(x)] - r(x) \to \frac{r'(x) \int K \left( \frac{|x-u|}{h} \right) (x - u) du}{\int K \left( \frac{|x-u|}{h} \right) du}$$

$$+ \frac{r''(x)}{2} \frac{\int K \left( \frac{|x-u|}{h} \right) (x - u)^2 du}{\int K \left( \frac{|x-u|}{h} \right) du}$$

$$= \frac{r''(x)}{2h} \int K \left( \frac{y}{h} \right) y^2 dy$$

$$= \frac{1}{2} r''(x) h^2 \sigma_K^2 \quad \text{where } \sigma_K^2 = \int K(z) z^2 dz.$$
Asymptotic results

So smoother is **asymptotically unbiased** if
\[ n \to \infty, \sup |x_{i+1} - x_i| \to 0 \text{ and } h \to 0. \]
Asymptotic results

So smoother is asymptotically unbiased if \( n \to \infty, \sup |x_{i+1} - x_i| \to 0 \) and \( h \to 0 \).

Similarly, as \( n \to \infty \) and \( \sup |x_{i+1} - x_i| \to 0 \),

\[
V[\hat{r}(x)] \to \frac{\sigma^2 h R(K)}{n \left( h \int K^2(y)dy \right)^2} = \frac{\sigma^2}{nh} R(K)
\]

where \( R(K) = \int K^2(y)dy \).
Asymptotic results

So smoother is **asymptotically unbiased** if $n \to \infty$, $\sup |x_{i+1} - x_i| \to 0$ and $h \to 0$. Similarly, as $n \to \infty$ and $\sup |x_{i+1} - x_i| \to 0$,

$$V[\hat{r}(x)] \to \frac{\sigma^2 h R(K)}{n \left( h \int K^2(y) dy \right)^2} = \frac{\sigma^2}{nh} R(K)$$

where $R(K) = \int K^2(y) dy$.

So the smoother is **consistent** if $n \to \infty$, $\sup |x_{j+1} - x_j| \to 0$, $h \to 0$ and $nh \to \infty$. 
Mean square error

Recall: MSE = Variance + Bias².

For a constant bandwidth kernel smoother:

\[
\text{Bias} \approx \frac{r''(x)h^2}{2\sigma_K^2}
\]

\[
\text{Var} \approx \frac{\sigma^2}{nh} R(K)
\]

So

\[
\text{MSE} \approx \frac{\sigma^2}{nh} R(K) + \frac{[r''(x)]^2 h^4}{4} \sigma_K^4.
\]
Mean square error

Recall: \( \text{MSE} = \text{Variance} + \text{Bias}^2 \).
For a constant bandwidth kernel smoother:

\[
\text{Bias} \approx \frac{r''(x)h^2}{\sigma^2} \\
\text{Var} \approx \frac{\sigma^2}{nh} R(K)
\]

So \( \text{MSE} \approx \frac{\sigma^2}{nh} R(K) + \frac{[r''(x)]^2 h^4}{4} \sigma_K^4 \).

So the optimal \( h \) can be obtained as follows.

\[
\frac{d\text{MSE}}{dh} = -\frac{\sigma^2}{nh^2} R(K) + \frac{[r''(x)]^2 h^3}{4} \sigma_K^4 = 0.
\]

\[
\Rightarrow \quad h_{\text{opt}} \approx \left( \frac{\sigma^2 R(K)}{n[r''(x)]^2 \sigma_K^4} \right)^{\frac{1}{5}}
\]
We can plug the optimal value of $h$ into the expression for MSE to obtain the minimum MSE obtainable. This is now a function of $r(x), \sigma^2$ and the kernel, $K$. 

Optimal kernels
Optimal kernels

- We can plug the optimal value of $h$ into the expression for MSE to obtain the minimum MSE obtainable. This is now a function of $r(x)$, $\sigma^2$ and the kernel, $K$.

- We can therefore choose a kernel which minimises the MSE. The optimal kernel is the Epanechnikov kernel, $K(u) = \frac{3}{4}(1 - u^2)$—the same as for kernel density estimation.
Optimal kernels

- We can plug the optimal value of $h$ into the expression for MSE to obtain the minimum MSE obtainable. This is now a function of $r(x), \sigma^2$ and the kernel, $K$.

- We can therefore choose a kernel which minimises the MSE. The optimal kernel is the Epanechnikov kernel, $K(u) = \frac{3}{4}(1 - u^2)$—the same as for kernel density estimation.

- In practice, the choice of kernel makes very little difference to the MSE. By far the most important choice is the bandwidth $h$. 
Kernel regression

**Implementation in R**

```r
require(KernSmooth)
plot(x, y)
fit <- locpoly(x, y, deg=0, bandwidth=.25)
lines(fit, col=4)
```
Outline

1. Some examples
2. Kernel averaging
3. Bandwidth selection
4. Local polynomial estimation
5. Derivative estimation
6. Inference
7. Multidimensional smoothing
Bandwidth selection

Recall: \[ h_{\text{opt}} \approx \left( \frac{\sigma^2 R(K)}{n[r''(x)]^2 \sigma_K^4} \right)^{\frac{1}{5}} \]
Bandwidth selection

Recall: \( h_{\text{opt}} \approx \left( \frac{\sigma^2 R(K)}{n[r''(x)]^2 \sigma_K^4} \right)^{\frac{1}{5}} \)

Using \( h_{\text{opt}} \) is not practicable since we never know \( r''(x) \).
Recall: \( h_{\text{opt}} \approx \left( \frac{\sigma^2 R(K)}{n[r''(x)]^2 \sigma^4_K} \right)^{1/5} \)

Using \( h_{\text{opt}} \) is not practicable since we never know \( r''(x) \).

In practice there are several approaches to overcome this problem.

1. The plug-in approach: estimate \( r''(x) \) under some simplifying assumptions.
Bandwidth selection

Recall: \( h_{\text{opt}} \approx \left( \frac{\sigma^2 R(K)}{n[r''(x)]^2 \sigma_K^4} \right)^{\frac{1}{5}} \)

Using \( h_{\text{opt}} \) is not practicable since we never know \( r''(x) \).

In practice there are several approaches to overcome this problem.

1. The plug-in approach: estimate \( r''(x) \) under some simplifying assumptions.
2. Cross-validation
Recall: \[ h_{opt} \approx \left( \frac{\sigma^2 R(K)}{n[r''(x)]^2 \sigma_K^4} \right)^{\frac{1}{5}} \]

Using \( h_{opt} \) is not practicable since we never know \( r''(x) \).

In practice there are several approaches to overcome this problem.

1. The plug-in approach: estimate \( r''(x) \) under some simplifying assumptions.

2. Cross-validation

3. Penalizing function
Bandwidth selection

Define average squared error:

$$ASE(h) = \frac{1}{n} \sum_{j=1}^{n} [\hat{r}(x_j) - r(x_j)]^2.$$ 

We don’t know \(r(x_j)\) but we can estimate it by \(y_j\). Then we have the average squared prediction error:

$$ASPE(h) = \frac{1}{n} \sum_{j=1}^{n} [\hat{r}(x_j) - y_j]^2.$$
Bandwidth selection

Define average squared error:

$$ASE(h) = \frac{1}{n} \sum_{j=1}^{n} [\hat{r}(x_j) - r(x_j)]^2.$$  

We don’t know $r(x_j)$ but we can estimate it by $y_j$. Then we have the average squared prediction error:

$$ASPE(h) = \frac{1}{n} \sum_{j=1}^{n} [\hat{r}(x_j) - y_j]^2.$$  

**Idea:** Find $h$ which minimises ASPE
Bandwidth selection

Define average squared error:

\[
\text{ASE}(h) = \frac{1}{n} \sum_{j=1}^{n} [\hat{r}(x_j) - r(x_j)]^2.
\]

We don’t know \( r(x_j) \) but we can estimate it by \( y_j \).
Then we have the average squared prediction error:

\[
\text{ASPE}(h) = \frac{1}{n} \sum_{j=1}^{n} [\hat{r}(x_j) - y_j]^2.
\]

**Idea:** Find \( h \) which minimises ASPE

**Problem:** Choosing \( h = 0 \) gives an interpolating function (\( \hat{r}(x_j) = y_j \)).
Cross-validation

Find $h$ which minimises

$$CV(h) = \frac{1}{n} \sum_{j=1}^{n} [\hat{r}_j(x_j) - y_j]^2$$

where $\hat{r}_j(x_j)$ uses all data except $(x_j, y_j)$. This is called a “leave-one-out” estimator.
Cross-validation

Find $h$ which minimises

$$CV(h) = \frac{1}{n} \sum_{j=1}^{n} [\hat{r}_j(x_j) - y_j]^2$$

where $\hat{r}_j(x_j)$ uses all data except $(x_j, y_j)$. This is called a "leave-one-out" estimator.

We can avoid computing $n$ separate smoothers by using a computational trick.
Penalizing function approach

The idea of penalizing the sum of squared errors can be generalized as follows: find $h$ which minimises penalized ASPE:

$$G(h) = \frac{1}{n} \sum_{j=1}^{n} [\hat{r}(x_j) - y_j]^2 p(w_j(x_j))$$

where $p(u)$ is a penalty function.
Penalizing function approach

The idea of penalizing the sum of squared errors can be generalized as follows: find \( h \) which minimises penalized ASPE:

\[
G(h) = \frac{1}{n} \sum_{j=1}^{n} [\hat{r}(x_j) - y_j]^2 p(w_j(x_j))
\]

where \( p(u) \) is a penalty function. CV is obtained with \( p(u) = (1 - u)^{-2} \).
Penalizing function approach

The idea of penalizing the sum of squared errors can be generalized as follows: find $h$ which minimises penalized ASPE:

$$G(h) = \frac{1}{n} \sum_{j=1}^{n} [\hat{r}(x_j) - y_j]^2 p(w_j(x_j))$$

where $p(u)$ is a penalty function. CV is obtained with $p(u) = (1 - u)^{-2}$. If $\hat{h}$ is minimising bandwidth of $G(h)$ and $\hat{h}_0$ is ASE optimal bandwidth, then

$$\frac{\text{ASE}(\hat{h})}{\text{ASE}(\hat{h}_0)} \xrightarrow{p} 1 \quad \text{and} \quad \frac{\hat{h}}{\hat{h}_0} \xrightarrow{p} 1.$$
**Examples:**

- Shibata’s selector: \( p(u) = 1 + 2u \)
- Generalized cross-validation: \( p(u) = (1 - u)^{-2} \)
- Akaike’s Information Criterion: \( p(u) = \exp(2u) \)
- Finite prediction error: \( p(u) = \frac{1 + u}{1 - u} \)
- Rice’s T: \( p(u) = (1 - 2u)^{-1} \)

The goal is to penalize small bandwidths. As \( h \to 0 \), \( w_j(x_j) \to 1 \). As \( h \to \infty \), \( w_j(x_j) \to 0 \).

Different \( p(u) \) are almost equal for large bandwidths but penalize small bandwidths differently.
### Examples:

<table>
<thead>
<tr>
<th>Method</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shibata’s selector</td>
<td>$p(u) = 1 + 2u$</td>
</tr>
<tr>
<td>Generalized cross-validation</td>
<td>$p(u) = (1 - u)^{-2}$</td>
</tr>
<tr>
<td>Akaike’s Information Criterion</td>
<td>$p(u) = \exp(2u)$</td>
</tr>
<tr>
<td>Finite prediction error</td>
<td>$p(u) = (1 + u)/(1 - u)$</td>
</tr>
<tr>
<td>Rice’s T</td>
<td>$p(u) = (1 - 2u)^{-1}$</td>
</tr>
</tbody>
</table>

- Goal is to penalize small bandwidths.
### Examples:

<table>
<thead>
<tr>
<th>Method</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shibata’s selector</td>
<td>( p(u) = 1 + 2u )</td>
</tr>
<tr>
<td>Generalized cross-validation</td>
<td>( p(u) = (1 - u)^{-2} )</td>
</tr>
<tr>
<td>Akaike’s Information Criterion</td>
<td>( p(u) = \exp(2u) )</td>
</tr>
<tr>
<td>Finite prediction error</td>
<td>( p(u) = (1 + u)/(1 - u) )</td>
</tr>
<tr>
<td>Rice’s T</td>
<td>( p(u) = (1 - 2u)^{-1} )</td>
</tr>
</tbody>
</table>

- Goal is to penalize small bandwidths.
- \( w_j(x_j) \to 1 \) as \( h \to 0 \) and \( w_j(x_j) \to 0 \) as \( h \to \infty \).
## Examples:

<table>
<thead>
<tr>
<th>Method</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shibata’s selector</td>
<td>$p(u) = 1 + 2u$</td>
</tr>
<tr>
<td>Generalized cross-validation</td>
<td>$p(u) = (1 - u)^{-2}$</td>
</tr>
<tr>
<td>Akaike’s Information Criterion</td>
<td>$p(u) = \exp(2u)$</td>
</tr>
<tr>
<td>Finite prediction error</td>
<td>$p(u) = (1 + u)/(1 - u)$</td>
</tr>
<tr>
<td>Rice’s $T$</td>
<td>$p(u) = (1 - 2u)^{-1}$</td>
</tr>
</tbody>
</table>

- Goal is to penalize small bandwidths.
- $w_j(x_j) \rightarrow 1$ as $h \rightarrow 0$ and $w_j(x_j) \rightarrow 0$ as $h \rightarrow \infty$.
- Different $p(u)$ are almost equal for large bandwidths but penalize small bandwidths differently.
Outline

1. Some examples
2. Kernel averaging
3. Bandwidth selection
4. Local polynomial estimation
5. Derivative estimation
6. Inference
7. Multidimensional smoothing
Local constant estimator

Define the weighted least squares function

\[
WLS(x) = \sum_{j=1}^{n} w_j(x)(y_j - a_0)^2.
\]

Then \( WLS(x) \) is minimised by \( \hat{a}_0 = \sum_{j=1}^{n} w_j(x)y_j. \)
Local constant estimator

Define the weighted least squares function

\[ \text{WLS}(x) = \sum_{j=1}^{n} w_j(x)(y_j - a_0)^2. \]

Then \( \text{WLS}(x) \) is minimised by \( \hat{a}_0 = \sum_{j=1}^{n} w_j(x)y_j. \)

- A kernel smoother is equivalent to a locally weighted constant fit, \( r(u) \approx a_0 \) in the neighbourhood of \( u \).
Local constant estimator

Define the weighted least squares function

\[ WLS(x) = \sum_{j=1}^{n} w_j(x)(y_j - a_0)^2. \]

Then \( WLS(x) \) is minimised by \( \hat{a}_0 = \sum_{j=1}^{n} w_j(x)y_j \).

- A kernel smoother is equivalent to a locally weighted constant fit, \( r(u) \approx a_0 \) in the neighbourhood of \( u \).
- \( \hat{a}_0 \) is a function of \( x \).
Local linear estimator

Assume

\[ r(u) \approx a_0 + a_1(u - x) \]

in the neighbourhood of \( x \) (i.e., when \( |u - x| < h \)).
Local linear estimator

Assume

\[ r(u) \approx a_0 + a_1(u - x) \]

in the neighbourhood of \( x \) (i.e., when \( |u - x| < h \)). Let \( \hat{a}_0 \) and \( \hat{a}_1 \) be the values of \( a_0 \) and \( a_1 \) which minimise

\[
\text{WLS}(x) = \sum_{j=1}^{n} w_j(x)(y_j - a_0 - a_1(x_j - x))^2.
\]
Assume

\[ r(u) \approx a_0 + a_1(u - x) \]

in the neighbourhood of \( x \) (i.e., when \( |u - x| < h \)).

Let \( \hat{a}_0 \) and \( \hat{a}_1 \) be the values of \( a_0 \) and \( a_1 \) which minimise

\[
WLS(x) = \sum_{j=1}^{n} w_j(x)(y_j - a_0 - a_1(x_j - x))^2.
\]

Then \( \hat{r}(x) = \hat{a}_0 \)
Local linear estimator

![Graph showing the relationship between world pulp price and pulp shipments. The x-axis represents the world pulp price (ranging from 500 to 850), and the y-axis represents pulp shipments (ranging from 10 to 35). The data points are scattered across the graph, indicating a potential trend between the two variables.]
Local linear estimator
Local linear estimator

World pulp price vs Pulp shipments

- X-axis: World pulp price (500 to 850)
- Y-axis: Pulp shipments (10 to 35)

The plot shows a local linear estimator with a continuous line and data points representing pulp shipments at different world pulp prices.
Local linear estimator

The diagram shows a scatter plot with data points representing world pulp price (x-axis) and pulp shipments (y-axis). The plot includes a local linear estimator (blue curve) and a local polynomial estimator (pink line) to illustrate the smoothing methods.

The x-axis ranges from 500 to 850, and the y-axis ranges from 10 to 35.
Local linear estimator
Local linear estimator

![Graph showing the relationship between world pulp price and pulp shipments](image)

- **World pulp price** (X-axis)
- **Pulp shipments** (Y-axis)

The graph illustrates a negative correlation between world pulp price and pulp shipments. A local linear estimator is fitted to the data points, indicating a trend that decreases as the world pulp price increases.
Local linear estimator

![Graph showing the relationship between world pulp price and pulp shipments. The graph has a scatter plot with a line of best fit. The x-axis represents world pulp price, ranging from 500 to 850. The y-axis represents pulp shipments, ranging from 10 to 35. The line of best fit indicates a negative correlation between the two variables.](image-url)
Local linear estimator

World pulp price

Pulp shipments
Local linear estimator

![Graph showing the relationship between world pulp price and pulp shipments. The graph includes a line indicating a negative correlation.]
Local linear estimator

![Graph showing the relationship between world pulp price and pulp shipments. The graph includes a scatter plot with data points and a linear trend line. The x-axis represents world pulp price, ranging from 500 to 850, and the y-axis represents pulp shipments, ranging from 10 to 35. The trend line suggests a negative correlation between the two variables.]
Local linear estimator

![Graph showing local linear estimator relationship between World pulp price and Pulp shipments. The graph includes a scatter plot with a pink line of best fit. The x-axis represents World pulp price ranging from 500 to 850, and the y-axis represents Pulp shipments ranging from 10 to 35.](image-url)
Local linear estimator

![Graph showing a scatter plot of world pulp price against pulp shipments with a local linear estimator line.](image-url)
Local linear estimator

![Graph showing the relationship between world pulp price and pulp shipments. The graph includes a scatter plot with a trend line indicating a negative correlation.](image-url)
Local linear estimator
**Local linear estimator**

![Graph showing the relationship between World pulp price and Pulp shipments](image)

- The graph illustrates the scatter plot of World pulp price against Pulp shipments.
- A local linear estimator is applied to the data points, indicated by the pink line.
- The data points are represented by circles, and the trend line shows a negative correlation, suggesting that as the world pulp price increases, pulp shipments decrease.

This visual representation helps in understanding the nonparametric smoothing methods, specifically the local linear estimator, in the context of economic data analysis.
Local linear estimator

![Graph showing the relationship between world pulp price and pulp shipments. The graph displays a scatter plot with a negative trend line. The x-axis represents world pulp price, ranging from 500 to 850, and the y-axis represents pulp shipments, ranging from 10 to 35. The data points are scattered along a downward-sloping line, indicating a negative correlation.]
Local linear estimator
Local linear estimator

![Graph showing the relationship between world pulp price and pulp shipments. The graph includes a scatter plot with data points and a fitted line representing the local linear estimator.](image)
Local linear estimator

World pulp price vs Pulp shipments graph.
Local linear estimator

Pulp shipments vs. World pulp price

- The graph shows a scatter plot with each data point representing a pair of pulp shipments and world pulp price.
- A line is fitted to the data points, indicating a trend in the relationship between pulp shipments and the world pulp price.

Key:
- Each dot represents a data point.
- The line is a local linear estimator.
Local linear estimator

The diagram shows a scatter plot with circles representing data points. The x-axis represents the World pulp price, ranging from 500 to 850, while the y-axis represents Pulp shipments, ranging from 10 to 35. A linear trend line is also included in the plot, indicating a negative correlation between the World pulp price and Pulp shipments.
Local linear estimator
Local linear estimator

![Graph showing the relationship between world pulp price and pulp shipments. The graph includes a scatter plot with data points and a pink line representing the local linear estimator. The x-axis represents the world pulp price, ranging from 500 to 850, while the y-axis represents pulp shipments, ranging from 10 to 35. The data points are spread across the graph, indicating a potential trend in the relationship.]
Local linear estimator

![Graph showing the relationship between world pulp price and pulp shipments. The graph includes a scatter plot with data points and a fitted line.](image-url)
Local linear estimator

World pulp price vs. Pulp shipments graph.

The local linear estimator is depicted as a smooth line fitting the scattered data points represented by circles. The x-axis shows the world pulp price, while the y-axis represents the pulp shipments. The graph highlights the trend between the two variables.
Local linear estimator

\[ \text{WLS}(x) = \sum_{j=1}^{n} w_j(x)(y_j - a_0 - a_1(x_j - x))^2. \]

Then \( \hat{r}(x) = \hat{a}_0 \)
Local linear estimator

$$WLS(x) = \sum_{j=1}^{n} w_j(x)(y_j - a_0 - a_1(x_j - x))^2.$$ 

Then $\hat{r}(x) = \hat{a}_0$

- If all weights, $w_j(x)$, are equal, this is equivalent to a least squares regression.
Local linear estimator

\[ WLS(x) = \sum_{j=1}^{n} w_j(x)(y_j - a_0 - a_1(x_j - x))^2. \]

Then \( \hat{r}(x) = \hat{a}_0 \)

- If all weights, \( w_j(x) \), are equal, this is equivalent to a least squares regression.
- Size of bandwidth controls smoothness. Larger bandwidths produce smoother curves.
Local linear estimator

\[ WLS(x) = \sum_{j=1}^{n} w_j(x)(y_j - a_0 - a_1(x_j - x))^2. \]

Then \( \hat{r}(x) = \hat{a}_0 \)

- If all weights, \( w_j(x) \), are equal, this is equivalent to a least squares regression.
- Size of bandwidth controls smoothness. Larger bandwidths produce smoother curves.
- Typically, in the interior of the data, the target point \( x \) is close to the neighbourhood mean, so a locally linear smoother behaves like a locally constant smoother.
Big advantage of locally linear smoothers is that they are much less biased at the boundaries of the predictor space than a locally constant smoother.
Local linear estimator

- Big advantage of locally linear smoothers is that they are much less biased at the boundaries of the predictor space than a locally constant smoother.

- Local linear smoothing was introduced by Cleveland (1979) and others. Fan and Gijbels (1992) and Fan (1993) showed it has excellent theoretical properties.
Local polynomial estimator

\[ \text{WLS}(x) = \sum_{j=1}^{n} w_j(x) \left[ y_j - a_0 - a_1(x_j - x) - \cdots - a_p(x_j - x)^p \right]^2 \]

and \( \hat{r}(x) = \hat{a}_0. \)
Local polynomial estimator

\[ WLS(x) = \sum_{j=1}^{n} w_j(x) [y_j - a_0 - a_1(x_j - x) - \cdots - a_p(x_j - x)^p]^2 \]

and \( \hat{r}(x) = \hat{a}_0 \).

In matrix notation we can write

\[ WLS(x) = (Y - Xa)^T W(x) (Y - Xa) \]

where \( [X]_{ji} = (x_j - x)^i \) and \( W(x) \) is the diagonal matrix with elements \( w_j(x) \).
Local quadratic estimator
Local quadratic estimator
Local quadratic estimator

![Graph showing the relationship between world pulp price and pulp shipments. The x-axis represents world pulp price with values ranging from 500 to 850. The y-axis represents pulp shipments with values ranging from 10 to 35. The graph includes a curve that models the quadratic relationship between the two variables.](image-url)
Local quadratic estimator

![Graph showing the relationship between world pulp price and pulp shipments. The graph includes a curve fitted to the data points, indicating a decreasing trend.]
Local quadratic estimator

![Graph showing the relationship between world pulp price and pulp shipments. The graph includes a line of best fit indicated by a magenta line, suggesting a negative correlation between the two variables.](image-url)
Local quadratic estimator
Local quadratic estimator

![Graph showing the relationship between world pulp price and pulp shipments. The graph includes a scatter plot with data points and a fitted line representing the local quadratic estimator.](image-url)
Local quadratic estimator

![Graph showing the relationship between world pulp price and pulp shipments. The graph displays a scatter plot with a local quadratic estimator line. The x-axis represents the world pulp price, ranging from 500 to 850, while the y-axis represents pulp shipments, ranging from 10 to 35.]
Local quadratic estimator

![Graph showing relationship between World pulp price and Pulp shipments]

- The graph plots World pulp price on the x-axis and Pulp shipments on the y-axis.
- The scatter plot with a trend line indicates a negative correlation between the two variables.
- The trend line is fitted using a local quadratic estimator, which is a nonparametric smoothing method commonly used in regression analysis.

This visualization helps in understanding the pattern or trend in the data without assuming a specific parametric form for the relationship.
Local quadratic estimator
Local quadratic estimator

![Graph showing the relationship between World pulp price and Pulp shipments](image)

- **Pulp shipments**
- **World pulp price**

- The graph illustrates the local quadratic estimator for the relationship between World pulp price and Pulp shipments.
Local quadratic estimator

![Graph showing the relationship between world pulp price and pulp shipments. The graph includes data points and a curve estimated using a local quadratic estimator.](image-url)
Local quadratic estimator

![Graph showing the relationship between World pulp price and Pulp shipments. The graph includes a fitted curve that represents the local quadratic estimator.](image)
Local quadratic estimator

![Graph showing the relationship between World pulp price and Pulp shipments](image)
Local quadratic estimator

World pulp price

Pulp shipments

500 550 600 650 700 750 800 850

10 15 20 25 30 35
Local quadratic estimator
Local quadratic estimator

The diagram illustrates the relationship between world pulp price and pulp shipments. The data points are shown as circles, and the line represents a local quadratic estimator smoothing the relationship. The x-axis represents world pulp price, while the y-axis represents pulp shipments.
Local quadratic estimator

![Graph showing World pulp price vs Pulp shipments. The graph displays a downward trend as the World pulp price increases. The data points are scattered, and a smooth curve is fitted to the data.]
Local quadratic estimator

![Graph showing the relationship between world pulp price and pulp shipments. The graph includes a smooth curve fitted to the data points, indicating a negative correlation.]
Local polynomial estimator

\[ WLS(x) = (Y - Xa)^T W(x)(Y - Xa) \]
Local polynomial estimator

\[ WLS(x) = (\mathbf{Y} - X\mathbf{a})^T W(x)(\mathbf{Y} - X\mathbf{a}) \]

The minimizer of this function is

\[ \hat{\mathbf{a}} = (X'W(x)X)^{-1}X'W(x)\mathbf{Y}. \]
**Local polynomial estimator**

\[ WLS(x) = (Y - Xa)^T W(x) (Y - Xa) \]

The minimizer of this function is

\[ \hat{a} = (X'W(x)X)^{-1} X'W(x)Y. \]

Therefore,

\[ \hat{r}(x) = [1, 0, \ldots, 0](X'W(x)X)^{-1} X'W(x)Y = \sum_{j=1}^{n} l_j(x)y_j \]

where

\[ l_j(x) = [1, 0, \ldots, 0](X'W(x)X)^{-1}[1, (x_j-x), \ldots, (x_j-x)^p]w_j(x) \]
Local polynomial estimator

\[
WLS(x) = (Y - Xa)^T W(x)(Y - Xa)
\]

The minimizer of this function is

\[
\hat{a} = (X'W(x)X)^{-1}X'W(x)Y.
\]

Therefore,

\[
\hat{r}(x) = [1, 0, \ldots, 0] (X'W(x)X)^{-1} X'W(x)Y = \sum_{j=1}^{n} l_j(x) y_j
\]

where

\[
l_j(x) = [1, 0, \ldots, 0] (X'W(x)X)^{-1} [1, (x_j-x), \ldots, (x_j-x)^p] w_j(x)
\]

So a local polynomial is equivalent to a kernel smoother but with an unusual weight function. We call the weights \(l_j(x)\) the **effective kernel** at \(x\). If \(p = 0\), then \(l_j(x) = w_j(x)\).
Bias and variance of local polynomials

Under suitable conditions on the $x_j$'s,

$$E[\hat{r}(x)] - r(x) = r'(x) \sum_{j=1}^{n} l_j(x)(x - x_j) + \frac{r''(x)}{2} \sum_{j=1}^{n} l_j(x)(x - x_j)^2 + \cdots.$$
Bias and variance of local polynomials

Under suitable conditions on the $x_j$'s,

$$E[\hat{r}(x)] - r(x) = r'(x) \sum_{j=1}^{n} l_j(x)(x - x_j)$$

$$+ \frac{r''(x)}{2} \sum_{j=1}^{n} l_j(x)(x - x_j)^2 + \cdots.$$ 

Let $q(x)$ be a polynomial of degree $\leq p$. and suppose we use local polynomial regression to fit a curve to $\{x_j, q(x_j)\}$. 

Bias and variance of local polynomials

Under suitable conditions on the $x_j$'s,

$$E[\hat{r}(x)] - r(x) = r'(x) \sum_{j=1}^{n} l_j(x)(x - x_j) + \frac{r''(x)}{2} \sum_{j=1}^{n} l_j(x)(x - x_j)^2 + \cdots.$$ 

Let $q(x)$ be a polynomial of degree $\leq p$. and suppose we use local polynomial regression to fit a curve to $\{x_j, q(x_j)\}$. Then the first $p$ terms in the bias expression are removed.
Bias and variance of local polynomials

For $p = 1$:

$$E[\hat{r}(x)] - r(x) \approx \frac{r''(x)}{2} \sum_{j=1}^{n} l_j(x)(x - x_j)^2$$
Bias and variance of local polynomials

For $p = 1$:

\[
E[\hat{r}(x)] - r(x) \approx \frac{r''(x)}{2} \sum_{j=1}^{n} l_j(x)(x - x_j)^2
\]

For $p = 2$:

\[
E[\hat{r}(x)] - r(x) \approx \frac{r'''(x)}{6} \sum_{j=1}^{n} l_j(x)(x - x_j)^3
\]
Bias and variance of local polynomials

For $p = 1$:

$$E[\hat{r}(x)] - r(x) \approx \frac{r''(x)}{2} \sum_{j=1}^{n} l_j(x)(x - x_j)^2$$

For $p = 2$:

$$E[\hat{r}(x)] - r(x) \approx \frac{r'''(x)}{6} \sum_{j=1}^{n} l_j(x)(x - x_j)^3$$

- It is not usually necessary to use $p > 2$. 
Bias and variance of local polynomials

**For** $p = 1$:

$$E[\hat{r}(x)] - r(x) \approx \frac{r''(x)}{2} \sum_{j=1}^{n} l_j(x)(x - x_j)^2$$

**For** $p = 2$:

$$E[\hat{r}(x)] - r(x) \approx \frac{r'''(x)}{6} \sum_{j=1}^{n} l_j(x)(x - x_j)^3$$

- It is not usually necessary to use $p > 2$.
- The variance of the local linear smoother is simply $\sum l_j^2(x)y_j$. 
Local polynomial estimation

Implementation in R

```r
h <- dpill(x, y)
fit <- locpoly(x, y, bandwidth=h)
plot(x, y)
lines(fit, col=2)
```
Loess

- implementation of local linear and local quadratic smoothing.
Loess

- implementation of local linear and local quadratic smoothing.
- developed by Bill Cleveland of AT&T Bell Laboratories.
Loess

- implementation of local linear and local quadratic smoothing.
- developed by Bill Cleveland of AT&T Bell Laboratories.
- available in many packages and is the most popular smoothing method currently available.
Loess

- implementation of local linear and local quadratic smoothing.
- developed by Bill Cleveland of AT&T Bell Laboratories.
- available in many packages and is the most popular smoothing method currently available.
- uses nearest neighbours and a tri-cube weight function.
Loess

- implementation of local linear and local quadratic smoothing.
- developed by Bill Cleveland of AT&T Bell Laboratories.
- available in many packages and is the most popular smoothing method currently available.
- uses nearest neighbours and a tri-cube weight function.
- Also uses some ideas of robustness to protect against outliers. This effectively works by repeatedly smoothing the data, and at each iteration down-weighting points with large residuals.
Loess

Implementation in R

loess(y ~ x, span)
Computes a loess smoother.
To plot the results:

fit <- loess(y ~ x, span=0.75)
xx <- 500:900
plot(x, y)
lines(xx, predict(fit, data.frame(x=xx)))
Outline

1. Some examples
2. Kernel averaging
3. Bandwidth selection
4. Local polynomial estimation
5. **Derivative estimation**
6. Inference
7. Multidimensional smoothing
Derivative estimation

\[ \hat{r}(x) = \sum_{j=1}^{n} l_j(x)y_j \quad \Rightarrow \quad \hat{r}^{(k)}(x) = \sum_{j=1}^{n} l_j^{(k)}(x)y_j. \]
Derivative estimation

\[ \hat{r}(x) = \sum_{j=1}^{n} l_j(x) y_j \quad \Rightarrow \quad \hat{r}^{(k)}(x) = \sum_{j=1}^{n} l_j^{(k)}(x) y_j. \]

- kth derivative of \( \hat{r}(x) \) found using kernel smooth with kth derivative of the weighting function as weights.
Derivative estimation

\[ \hat{r}(x) = \sum_{j=1}^{n} l_j(x) y_j \Rightarrow \hat{r}^{(k)}(x) = \sum_{j=1}^{n} l_j^{(k)}(x) y_j. \]

- $k$th derivative of $\hat{r}(x)$ found using kernel smooth with $k$th derivative of the weighting function as weights.
- If $l_j(x)$ is not smooth, then $l_j^{(k)}(x)$ will have some discontinuities.
Derivative estimation

\[ \hat{r}(x) = \sum_{j=1}^{n} l_j(x)y_j \quad \Rightarrow \quad \hat{r}^{(k)}(x) = \sum_{j=1}^{n} l_j^{(k)}(x)y_j. \]

-\( k \)th derivative of \( \hat{r}(x) \) found using kernel smooth with \( k \)th derivative of the weighting function as weights.

- If \( l_j(x) \) is not smooth, then \( l_j^{(k)}(x) \) will have some discontinuities.

- To obtain smooth estimate of \( \hat{r}^{(k)}(x) \), we need \( l_j(x) \) to have continuous derivatives up to order \( k \). This rules out many of the standard kernel weighting functions.
Derivative estimation

Using the standard bias expression derived earlier, we see that

\[
E[\hat{r}^{(k)}(x)] = r(x) \sum_{j=1}^{n} l_j^{(k)}(x) + r'(x) \sum_{j=1}^{n} l_j^{(k)}(x)(x - x_j) \\
+ \frac{r''(x)}{2} \sum_{j=1}^{n} l_j^{(k)}(x)(x - x_j)^2 + \ldots
\]
Using the standard bias expression derived earlier, we see that

$$E[\hat{r}^{(k)}(x)] = r(x) \sum_{j=1}^{n} l_j^{(k)}(x) + r'(x) \sum_{j=1}^{n} l_j^{(k)}(x)(x - x_j)$$

$$+ \frac{r''(x)}{2} \sum_{j=1}^{n} l_j^{(k)}(x)(x - x_j)^2 + \ldots$$

Therefore, for an asymptotically unbiased estimator of $r'(x)$,

$$\sum_{j=1}^{n} l_j^{(1)}(x) = 0$$

and

$$\sum_{j=1}^{n} l_j^{(1)}(x)(x - x_j) = 1.$$
Using the standard bias expression derived earlier, we see that

\[
E[\hat{r}^{(k)}(x)] = r(x) \sum_{j=1}^{n} l_j^{(k)}(x) + r'(x) \sum_{j=1}^{n} l_j^{(k)}(x)(x - x_j) + \frac{r''(x)}{2} \sum_{j=1}^{n} l_j^{(k)}(x)(x - x_j)^2 + \ldots
\]

Therefore, for an asymptotically unbiased estimator of \( r'(x) \),

\[
\sum_{j=1}^{n} l_j^{(1)}(x) = 0
\]

and

\[
\sum_{j=1}^{n} l_j^{(1)}(x)(x - x_j) = 1.
\]

Local polynomials of degree \( p \geq 1 \) will satisfy these constraints.
Derivative estimation

For an asymptotically unbiased estimator of \( r''(x) \)

\[
\sum_{j=1}^{n} l_j^{(2)}(x) = 0
\]

\[
\sum_{j=1}^{n} l_j^{(2)}(x)(x - x_j) = 0
\]

and

\[
\sum_{j=1}^{n} l_j^{(2)}(x)(x - x_j)^2 = 2.
\]
Derivative estimation

For an asymptotically unbiased estimator of $r''(x)$

$$\sum_{j=1}^{n} l_j^{(2)}(x) = 0$$

$$\sum_{j=1}^{n} l_j^{(2)}(x)(x - x_j) = 0$$

and

$$\sum_{j=1}^{n} l_j^{(2)}(x)(x - x_j)^2 = 2.$$

Local polynomials of degree $p \geq 2$ will satisfy these constraints.
Derivative estimation

For an asymptotically unbiased estimator of $r''(x)$

$$\sum_{j=1}^{n} l_j^{(2)}(x) = 0$$

$$\sum_{j=1}^{n} l_j^{(2)}(x)(x - x_j) = 0$$

and

$$\sum_{j=1}^{n} l_j^{(2)}(x)(x - x_j)^2 = 2.$$ 

Local polynomials of degree $p \geq 2$ will satisfy these constraints.

Implementation in R

```r
fit <- locpoly(x, y, bandwidth=h, drv=1)
plot(fit, type="l")
```
Derivative estimation

[Graph showing the relationship between world pulp price and pulp shipments, with a smooth curve fitting the data points.]

Change in pulp shipments

[Graph showing the change in pulp shipments against world pulp price, with a smooth curve.]
Outline

1. Some examples
2. Kernel averaging
3. Bandwidth selection
4. Local polynomial estimation
5. Derivative estimation
6. Inference
7. Multidimensional smoothing
Inference for linear smoothers

All local polynomial methods can be written in the form

$$\hat{r}(x) = \sum_{j=1}^{n} w_j(x)y_j.$$ 

Thus they are linear in the observations. The set of weights, $w_j(x)$, is known as the equivalent kernel at $x$. 
Inference for linear smoothers

All local polynomial methods can be written in the form

\[ \hat{r}(x) = \sum_{j=1}^{n} w_j(x)y_j. \]

Thus they are linear in the observations. The set of weights, \( w_j(x) \), is known as the equivalent kernel at \( x \).

Let \( \hat{r} = [\hat{r}(x_1), \hat{r}(x_2), \ldots, \hat{r}(x_n)]^T \). Then

\[ \hat{r} = S y \]

where \( S = [w_j(x_i)] \) is an \( n \times n \) matrix that we call a smoother matrix.
The rows of $S$ are the equivalent kernels for producing fits at each of the observed values $x_1, \ldots, x_n$. 
Inference for linear smoothers

- The rows of $S$ are the equivalent kernels for producing fits at each of the observed values $x_1, \ldots, x_n$.
- Any reasonable smoother should preserve a constant function so that $S1 = 1$ where $1$ is a vector of ones. This implies that the sum of the weights in each row is one.
The rows of $S$ are the equivalent kernels for producing fits at each of the observed values $x_1, \ldots, x_n$.

Any reasonable smoother should preserve a constant function so that $S \mathbf{1} = \mathbf{1}$ where $\mathbf{1}$ is a vector of ones. This implies that the sum of the weights in each row is one.

The matrix $S$ is analogous to the hat matrix $H = X (X^T X)^{-1} X^T$ in a standard linear model.
Inference for linear smoothers

The bias vector is
\[ \mathbf{b} = \mathbf{r} - \text{E}(\mathbf{Sy}) = \mathbf{r} - \mathbf{Sr} = (I - S)\mathbf{r}. \]
The bias vector is
\[ \mathbf{b} = \mathbf{r} - \mathbb{E}(\mathbf{Sy}) = \mathbf{r} - \mathbf{Sr} = (I - S)\mathbf{r}. \]

Then we can compute the mean square error as
\[
\text{MSE} = \frac{1}{n} \sum_{j=1}^{n} V(\hat{r}_j) + \frac{1}{n} \sum_{j=1}^{n} b_i^2 = \frac{\text{tr} (\mathbf{SS}^T)}{n} \sigma^2 + \frac{\mathbf{b}^T \mathbf{b}}{n}
\]
Inference for linear smoothers

The bias vector is
\[ \mathbf{b} = \mathbf{r} - \mathbb{E}(\mathbf{Sy}) = \mathbf{r} - \mathbf{Sr} = (I - S)\mathbf{r}. \]

Then we can compute the mean square error as
\[
\text{MSE} = \frac{1}{n} \sum_{j=1}^{n} V(\hat{r}_j) + \frac{1}{n} \sum_{j=1}^{n} b_i^2 \\
= \frac{\text{tr}(SS^T)}{n} \sigma^2 + \underbrace{\frac{\mathbf{b}^T \mathbf{b}}{n}}_{\text{bias squared}}
\]

The first term measures variance while the second measures squared bias.
Degrees of freedom

Want: Approximate degrees of freedom for our linear smoothers.
Degrees of freedom

**Want:** Approximate degrees of freedom for our linear smoothers.
- high df for very wiggly smoothers

Least squares regression:

\[ S = X (X^T X)^{-1} X^T. \]

\[ \gamma = \text{df} = \text{rank}(S) = \text{tr}(S) = \text{tr}(SS^T) = \text{tr}(2S - SS^T). \]

Any of these could be used for df of general linear smoother.
Degrees of freedom

**Want:** Approximate degrees of freedom for our linear smoothers.
- high df for very wiggly smoothers
- low df for very smooth smoothers.
Degrees of freedom

**Want:** Approximate degrees of freedom for our linear smoothers.
- high df for very wiggly smoothers
- low df for very smooth smoothers.
- **Least squares regression:** \( S = X(X^T X)^{-1} X^T \).

\[
\gamma = \text{df} = \# \text{ linearly independent predictors} \\
= \text{rank}(S) \\
= \text{tr}(S) \\
= \text{tr}(SS^T) \\
= \text{tr}(2S - SS^T).
\]
Degrees of freedom

**Want:** Approximate degrees of freedom for our linear smoothers.

- high df for very wiggly smoothers
- low df for very smooth smoothers.

Least squares regression: \( S = X(X^T X)^{-1} X^T. \)

\[ \gamma = \text{df} = \# \text{ linearly independent predictors} \]
\[ = \text{rank}(S) \]
\[ = \text{tr}(S) \]
\[ = \text{tr}(SS^T) \]
\[ = \text{tr}(2S - SS^T). \]
**Degrees of freedom**

**Want:** Approximate degrees of freedom for our linear smoothers.

- high df for very wiggly smoothers
- low df for very smooth smoothers.

Least squares regression: $S = X(X^TX)^{-1}X^T$.

\[ \gamma = \text{df} = \# \text{ linearly independent predictors} \]
\[ = \text{rank}(S) \]
\[ = \text{tr}(S) \]
\[ = \text{tr}(SS^T) \]
\[ = \text{tr}(2S - SS^T). \]

Any of these could be used for df of general linear smoother.
Degrees of freedom

Implementation in R

```r
price.model <- loess(formula = shipments ~ price)
price.model
Number of Observations: 25
Equivalent Number of Parameters: 4.69
Residual Standard Error: 2.383
xx <- 500:900
pred <- predict(price.model, data.frame(price=xx))
plot(price, shipments)
lines(xx, pred, col=2)
```
Degrees of freedom

The graph shows a scatter plot with price on the x-axis and shipments on the y-axis. There is a smooth red curve that approximates the trend in the data points, indicating a nonparametric smoothing method. The data points are spread out, and the curve helps to infer the underlying relationship between price and shipments.
Linear regression: error has $n - \gamma$ df.
Estimating the variance

Linear regression: error has \( n - \gamma \) df.

Hence define df of error for a linear smoother as \( n - \gamma \) where \( \gamma = \text{tr}(S) \).
Estimating the variance

Linear regression: error has $n - \gamma$ df.

Hence define df of error for a linear smoother as $n - \gamma$ where $\gamma = \text{tr}(S)$.

Assuming zero bias for smoother, an unbiased estimator of $\sigma^2$ is given by

$$\hat{\sigma}^2 = \frac{1}{n-\gamma} \sum_{j=1}^{n} (y_j - \hat{r}(x_j))^2 / (n - \gamma).$$

Here we have assumed that $\sigma^2$ is constant for all $x$. 
Estimating conditional variance

If $\sigma^2$ is not constant for all $x$, use

$$\hat{\sigma}^2(x) = \sum_{j=1}^{n} w_j(x)(y_i - \hat{r}(x_i))^2.$$ 

Note that this is a fairly crude estimator and no attempt has been made to adjust it for the degrees of freedom of the smoother.
Confidence intervals

\[ \text{Cov}(\hat{r}) = SS^T \sigma^2 \]
Confidence intervals

\[ \text{Cov}(\hat{r}) = \mathbf{S}\mathbf{S}^T \sigma^2 \]

Assuming negligible bias, approximate 95% CI for \( r \) are:

\[ \hat{r} \pm 1.96 \sqrt{\text{diag}(\mathbf{S}\mathbf{S}^T \sigma^2)}. \]
Confidence intervals

\[ \text{Cov}(\hat{r}) = SS^T \sigma^2 \]

Assuming negligible bias, approximate 95% CI for \( r \) are:

\[ \hat{r} \pm 1.96 \sqrt{\text{diag}(SS^T \sigma^2)}. \]

- Pointwise intervals. (i.e., 95% CI for each value of \( x \).)
Confidence intervals

\[ \text{Cov}(\hat{r}) = \mathbf{SS}^T \sigma^2 \]

Assuming negligible bias, approximate 95% CI for \( r \) are:

\[ \hat{r} \pm 1.96 \sqrt{\text{diag}(\mathbf{SS}^T \sigma^2)}. \]

- Pointwise intervals. (i.e., 95% CI for each value of \( x \).)
- On average, true value of \( r(x) \) lies outside these intervals 5% of the time.
Confidence intervals

Implementation in R

```r
price.model <- loess(shipments ~ price)
xx <- 500:900
pred <- predict(price.model,
               data.frame(price=xx), se=TRUE)
plot(price, shipments)
lines(xx, pred$fit, col=2)
lines(xx, pred$fit + 2*pred$se, col=4)
lines(xx, pred$fit - 2*pred$se, col=4)
```
Confidence intervals

The graph shows the relationship between price and shipments. The data points are represented by circles, and the curves indicate confidence intervals for the relationship. The x-axis represents the price, and the y-axis represents the number of shipments.
Approximate F tests

Approximate F tests used in LS regression, by using the approximate df.

\[ \hat{r}_1 = S_1 y (\text{df} = \gamma_1) \]
\[ \hat{r}_2 = S_2 y (\text{df} = \gamma_2) \]

\( \gamma_i = \text{df} = \text{tr}(2S_i - S_i S_i^T) \) for each of the models \( i = 1, 2 \).

Example: \( \hat{r}_2 \) may be rougher than \( \hat{r}_1 \) and we wish to test if it picks up significant bias.
Approximate F tests

Approximate F tests used in LS regression, by using the approximate df.

To compare two smooths:

\[ \hat{r}_1 = S_1 y \quad (df = \gamma_1) \]
\[ \hat{r}_2 = S_2 y \quad (df = \gamma_2). \]

\[ \gamma_i = df = \text{tr}(2S_i - S_i S_i^T) \] for each of the models \( i = 1, 2. \)
Approximate F tests

Approximate F tests used in LS regression, by using the approximate df.

To compare two smooths:

\[ \hat{r}_1 = S_1 y \quad (df = \gamma_1) \]
\[ \hat{r}_2 = S_2 y \quad (df = \gamma_2). \]

\[ \gamma_i = df = \text{tr}(2S_i - S_i S_i^T) \] for each of the models \( i = 1, 2. \)

**Example:** \( \hat{r}_2 \) may be rougher than \( \hat{r}_1 \) and we wish to test if it picks up significant bias.
Approximate F tests

Let $RSS_1$ and $RSS_2$ be residual sum of squares for each smoother.
Approximate F tests

Let $\text{RSS}_1$ and $\text{RSS}_2$ be residual sum of squares for each smoother.

We assume that $\hat{r}_2$ is unbiased, and $\hat{r}_1$ is unbiased under the null hypothesis. Thus, the null hypothesis implies that there is no significant difference between the two smoothers.
Approximate F tests

Let $\text{RSS}_1$ and $\text{RSS}_2$ be residual sum of squares for each smoother.

We assume that $\hat{r}_2$ is unbiased, and $\hat{r}_1$ is unbiased under the null hypothesis. Thus, the null hypothesis implies that there is no significant difference between the two smoothers.

Then

$$\frac{(\text{RSS}_1 - \text{RSS}_2) / (\gamma_2 - \gamma_1)}{\text{RSS}_2 / (n - \gamma_2)} \sim F_{\gamma_2 - \gamma_1, n - \gamma_2}.$$
Application: test for linearity

Let $\hat{r}_1$ represent a linear regression and we wish to test if the linearity is real by fitting a nonparametric nonlinear smooth curve $\hat{r}_2$.

**Implementation in R**

```r
fit.1 <- loess(shipments ~ price, span=100, deg=1)
fit.2 <- loess(shipments ~ price, span=1/2, deg=1)
anova(fit.1, fit.2)
```

<table>
<thead>
<tr>
<th></th>
<th>ENP</th>
<th>RSS</th>
<th>F-value</th>
<th>Pr(&gt;F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.000</td>
<td>208.945</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4.220</td>
<td>106.264</td>
<td>4.957</td>
<td>0.007762 **</td>
</tr>
</tbody>
</table>

Analysis of Variance: denominator df 19.51
Outline

1. Some examples
2. Kernel averaging
3. Bandwidth selection
4. Local polynomial estimation
5. Derivative estimation
6. Inference
7. Multidimensional smoothing
Multidimensional smoothers

If \( m \geq 2 \) explanatory variables, need to fit surface rather than line.

\[
\hat{r}(z) = \sum_{j=1}^{n} w_j(z) y_j
\]

where \( w_j(z) = K_m(z - x_j) \)

\[
\sum_{j=1}^{n} K_m(z - x_j)
\]

Note that \( z \) and \( x_j \) are \( m \)-dimensional vectors and \( K_m(u) \) is an \( m \)-dimensional function.

Product kernel

\[
K_m(u) = \prod_{i=1}^{m} \frac{1}{h_i} K\left(\frac{u_i}{h_i}\right)
\]

where \( K(u) \) is univariate kernel and \( h_i \) is smoothing parameter in \( i \)th dimension.

Multidimensional distance

\[
K_m(u) = \frac{1}{h} K\left(\frac{\|u\|}{h}\right)
\]

where \( \|u\| \) is distance metric (e.g. Euclidean distance). Only one smoothing parameter, \( h \), used.
Multidimensional smoothers

If $m \geq 2$ explanatory variables, need to fit surface rather than line.

\[
\hat{r}(z) = \sum_{j=1}^{n} w_j(z) y_j \quad \text{where} \quad w_j(z) = \frac{K_m(z - x_j)}{\sum_{j=1}^{n} K_m(z - x_j)}.
\]
Multidimensional smoothers

If $m \geq 2$ explanatory variables, need to fit surface rather than line.

\[
\hat{r}(z) = \sum_{j=1}^{n} w_j(z)y_j \quad \text{where} \quad w_j(z) = \frac{K_m(z - x_j)}{\sum_{j=1}^{n} K_m(z - x_j)}.
\]

Note that $z$ and $x_j$ are $m$-dimensional vectors and $K_m(u)$ is an $m$-dimensional function.
Multidimensional smoothers

If \( m \geq 2 \) explanatory variables, need to fit surface rather than line.

\[
\hat{r}(z) = \sum_{j=1}^{n} w_j(z)y_j \quad \text{where} \quad w_j(z) = \frac{K_m(z - x_j)}{\sum_{j=1}^{n} K_m(z - x_j)}.
\]

Note that \( z \) and \( x_j \) are \( m \)-dimensional vectors and \( K_m(u) \) is an \( m \)-dimensional function.

**Product kernel**

\[
K_m(u) = \prod_{i=1}^{m} \frac{1}{h_i} K(u_i/h_i)
\]

where \( K(u) \) is univariate kernel and \( h_i \) is smoothing parameter in \( i \)th dimension.
Multidimensional smoothers

If $m \geq 2$ explanatory variables, need to fit surface rather than line.

$$\hat{r}(z) = \sum_{j=1}^{n} w_j(z) y_j \quad \text{where} \quad w_j(z) = \frac{K_m(z - x_j)}{\sum_{j=1}^{n} K_m(z - x_j)}.$$

Note that $z$ and $x_j$ are $m$-dimensional vectors and $K_m(u)$ is an $m$-dimensional function.

**Product kernel**

$$K_m(u) = \prod_{i=1}^{m} \frac{1}{h_i} K(u_i/h_i)$$

where $K(u)$ is univariate kernel and $h_i$ is smoothing parameter in $i$th dimension.

**Multidimensional distance**

$$K_m(u) = \frac{1}{h} K(\|u\|/h)$$

where $\|u\|$ is distance metric (e.g. Euclidean distance). Only one smoothing parameter, $h$, used.
Multidimensional smoothers

- If multidimensional distance used, it is usually necessary to standardise each predictor by dividing by its standard deviation or some other measure of spread.
If multidimensional distance used, it is usually necessary to standardise each predictor by dividing by its standard deviation or some other measure of spread.

Multidimensional loess normalizes each predictor by dividing by their 10% trimmed sample standard deviation.
Multidimensional smoothers

- If multidimensional distance used, it is usually necessary to standardise each predictor by dividing by its standard deviation or some other measure of spread.
- Multidimensional loess normalizes each predictor by dividing by their 10\% trimmed sample standard deviation.
- If $m = 1$, both methods give the standard univariate results.
Local polynomial surfaces

Locally weighted lines generalise easily to higher dimensions.

- Instead of computing local lines, compute a local plane.
Local polynomial surfaces

Locally weighted lines generalise easily to higher dimensions.

- Instead of computing local lines, compute a local plane.
- If explanatory variables are $w$ and $v$, local plane is computed using multiple regression on $w$ and $v$. 

Implementation in R

fit <- loess(y ~ x*z, span=0.8)
Local polynomial surfaces

Locally weighted lines generalise easily to higher dimensions.

- Instead of computing local lines, compute a local plane.
- If explanatory variables are $w$ and $v$, local plane is computed using multiple regression on $w$ and $v$.
- Local quadratic surfaces computed using multiple regression on $w$, $v$, $wv$, $w^2$ and $v^2$.

Implementation in R

```r
fit <- loess(y ~ x*z, span=0.8)
```
Local polynomial surfaces

Locally weighted lines generalise easily to higher dimensions.

- Instead of computing local lines, compute a local plane.
- If explanatory variables are $w$ and $v$, local plane is computed using multiple regression on $w$ and $v$.
- Local quadratic surfaces computed using multiple regression on $w$, $v$, $wv$, $w^2$ and $v^2$.

Implementation in R
\[
\text{fit} \leftarrow \text{loess}(y \sim x*z, \text{span}=0.8)
\]
Local polynomial surfaces

Locally weighted lines generalise easily to higher dimensions.

- Instead of computing local lines, compute a local plane.
- If explanatory variables are $w$ and $v$, local plane is computed using multiple regression on $w$ and $v$.
- Local quadratic surfaces computed using multiple regression on $w$, $v$, $wv$, $w^2$ and $v^2$.

Implementation in R

```r
fit <- loess(y ~ x*z, span=0.8)
```
Savings rate: personal saving divided by disposable income.

pop15: percent population under age of 15

ddpi: percent growth rate of per-capita disposable income in $
Example
Example
Example
Example
Example
Example
Example
Example
Example
Example
Example

![Diagram showing multidimensional smoothing with variables pop15, ddpi, and savings.](image-url)
Example
Example
Example
Example
Example
Example
Example
Example
Example
Example
Example
Example
Example
Example
Example

![Diagram showing multidimensional smoothing with variables pop15, ddpi, and savings.](image)
Example

The image shows a 3D plot with axes labeled 'ddpi', 'savings', and 'pop15'. The plot illustrates a multidimensional smoothing method, possibly for the variables 'pop15', 'ddpi', and 'savings'. The smooth surface indicates a nonparametric approach to understanding the relationship between these variables.
Example
Example
Example
Example
Example
Example
Example
Example
Example
Example
Example
Example