Lecture 16 — Filtering, Signal Extraction, and Smoothing in Time and Frequency Domains

Chapter 7 with textbook smoothing tools: linear filters, gain and phase, kernels, and HP-style methods

Jiajing Sun

School of Economics and Management, University of Chinese Academy of Sciences

Econometrics and Time Series Methods

Spring 2026

Why Lecture 16 matters

Lecture 15 introduced filtering only as a bridge. Lecture 16 slows down and treats filtering as a full topic in its own right.

1. A filter is a **linear operator** that changes what parts of a series are emphasized.
2. The same filter has both a **time-domain representation** and a **frequency-domain representation**.
3. In applications, filtering is never neutral: it changes trends, cycles, timing, and edge behavior.

Big message

Filtering is where econometrics stops treating dependence as a nuisance and starts treating low-frequency movement, cycles, and noise as objects to be extracted and interpreted.

Empirical use: Blue text marks the application-motivation notes added in this revision, so we can review the empirical purpose of each filter before polishing the lecture script.

Where Lecture 16 fits in the course

- Lecture 14: fixed-b and self-normalization for robust inference.
- Lecture 15: bootstrap, GMM, and the transition toward signal extraction.
- **Lecture 16:** deterministic filters, smoothing, gain, phase, and practical endpoint issues.
- Lecture 17: state-space models and Kalman filtering.

Why the split is useful

If filtering is compressed into one short block, it is easy to memorize names of filters but hard to understand what the formulas are doing. This lecture gives the mathematical detail that makes the later state-space lecture much easier to follow.

Learning goals

By the end of the lecture, you should be able to:

1. define a linear filter in lag-polynomial form;
2. explain the difference between one-sided, two-sided, causal, and symmetric filters;
3. derive the transfer function, gain, and phase of a filter;
4. interpret differencing, moving averages, and band-pass filters as frequency-selective devices;
5. explain kernel smoothing, bandwidth choice, and boundary bias;
6. write the smoothing-spline / HP objective and interpret the penalty parameter;
7. discuss generated-regressor and robustness issues in practical filtering.

Three-hour plan

Hour 1

Filtering in the time domain: linear filters, lag polynomials, moving averages, differencing, one-sided versus two-sided filters, and endpoint problems.

Hour 2

Filtering in the frequency domain: transfer functions, gain, phase, low-pass / high-pass / band-pass logic, and business-cycle filters.

Hour 3

Smoothing in practice: kernel methods, one-sided smoothers, local linear correction, smoothing splines, the HP filter, robust filters, and R workflow.

Why filter at all?

In many empirical settings we do not want the raw series itself. We want a transformed version:

$$\text{raw data} \quad \Longrightarrow \quad \text{trend, cycle, smoothed signal, detrended residual, forecast input.}$$

- Macro: separate trend from business-cycle movement.
- Finance: smooth noisy high-frequency variation or emphasize returns rather than levels.
- Forecasting: remove measurement noise and isolate persistent components.

Textbook intuition

Filtering transforms observed data into a form that is more useful for analysis, forecasting, or signal extraction.

Definition of a linear filter

For an observed series $\{Y_t\}$, a linear filter creates a new series $\{X_t\}$ by

$$X_t = \sum_{j=-\infty}^{\infty} a_j Y_{t-j},$$

whenever the weighted sum is well defined.

- The coefficients $\{a_j\}$ are the **filter weights**.
- Positive and negative weights are both allowed.
- Finite filters have only finitely many nonzero a_j's.

Interpretation

The coefficients answer a very direct question: how much of yesterday, today, and tomorrow do we keep when constructing the transformed series?

Lag-polynomial notation

Using the lag operator $LY_t = Y_{t-1}$, we can write the filter compactly as

$$X_t = a(L)Y_t, \qquad a(L) = \sum_{j=-\infty}^{\infty} a_j L^j.$$

- If $a_j = 0$ for $j < 0$, the filter uses only present and past observations.
- If $a_j = a_{-j}$, the filter is symmetric.
- If $\sum_j a_j = 1$, the filter preserves constants.

Why notation matters

Lag-polynomial notation lets us treat filtering, differencing, smoothing, and detrending in one unified algebraic language.

What a filter does to the mean

If Y_t is weakly stationary with mean μ, then

$$\mathbb{E}(X_t) = \mathbb{E}\left(\sum_j a_j Y_{t-j}\right) = \mu \sum_j a_j.$$

- If $\sum_j a_j = 1$, the filter leaves the mean unchanged.
- If $\sum_j a_j = 0$, the filter removes constant components.
- Differencing has $\sum_j a_j = 1 + (-1) = 0$, so it kills level components.

Low-frequency clue

The sum of the coefficients already tells us whether the filter keeps or removes very slow movement.

What a filter does to covariance

If $\gamma_Y(h) = \mathrm{Cov}(Y_t, Y_{t-h})$, then

$$\gamma_X(h) = \mathrm{Cov}\left(\sum_j a_j Y_{t-j}, \sum_k a_k Y_{t-h-k}\right) = \sum_j \sum_k a_j a_k \gamma_Y(h + j - k).$$

- Filtering changes the serial dependence pattern, not only the scale.
- A smoother can turn a noisy short-memory series into a much more persistent-looking output.
- A difference filter can reduce low-frequency persistence sharply.

Warning

After filtering, the dynamic properties of the series are different. A filtered series should not be interpreted as if it were the original data.

Stationary-spectrum implication

For a weakly stationary input with spectral density $f_Y(\lambda)$, the output has

$$f_X(\lambda) = |A(\lambda)|^2 f_Y(\lambda), \qquad A(\lambda) = \sum_{j=-\infty}^{\infty} a_j e^{-ij\lambda}.$$

- $A(\lambda)$ is the **transfer function**.
- $|A(\lambda)|$ is the **gain**.
- The filter shapes the spectrum by multiplying the input spectrum by $|A(\lambda)|^2$.

Key bridge

The same coefficients $\{a_j\}$ that look like weighted averages in the time domain become a frequency-selective device in the spectral domain.

Causal, one-sided, two-sided, and symmetric filters

Causal / one-sided

$$X_t = \sum_{j=0}^{\infty} a_j Y_{t-j}.$$

- real-time feasible;
- used in forecasting and nowcasting;
- no future data required.

Two-sided / symmetric

$$X_t = \sum_{j=-m}^{m} a_j Y_{t-j}, \qquad a_j = a_{-j}.$$

- often smoother;
- no phase shift if symmetric;
- not feasible in real time.

Practical split

Historical description often prefers two-sided filters. Real-time policy and forecasting require one-sided filters.

Empirical use: In economics, two-sided filters are common for ex post output-gap or business-cycle dating; one-sided filters are needed for policy dashboards, trading signals, and engineering control systems where future observations are unavailable.

Simple moving average filter

The symmetric moving average with half-window r is

$$X_t = \frac{1}{2r+1} \sum_{j=-r}^{r} Y_{t-j}.$$

- Equal positive weights average nearby observations.
- Constants are preserved because the weights sum to one.
- Short-run wiggles are damped by local averaging.

Interpretation

This is the textbook prototype of a low-pass smoother: slow movements pass through almost unchanged, while very rapid oscillations are attenuated.

Empirical use: In empirical work, moving averages are useful for reporting trends in sales, passenger traffic, prices, sensor readings, and epidemiological series. The implication is descriptive rather than structural: the smoother clarifies a path but can blur turning points.

Transfer function of the moving average

For the symmetric moving average,

$$A_r(\lambda) = \frac{1}{2r+1} \sum_{j=-r}^{r} e^{-ij\lambda} = \frac{1}{2r+1} \frac{\sin((2r+1)\lambda/2)}{\sin(\lambda/2)}.$$

$$|A_r(\lambda)| = \frac{1}{2r+1} \left| \frac{\sin((2r+1)\lambda/2)}{\sin(\lambda/2)} \right|.$$

- Near $\lambda = 0$, the gain is close to one.
- At higher frequencies, the gain falls and oscillates.
- Larger windows create stronger smoothing but more delay and endpoint loss.

Differencing as a high-pass filter

The ordinary difference filter is

$$\Delta Y_t = (1 - L) Y_t = Y_t - Y_{t-1}.$$

Its transfer function is

$$A_\Delta(\lambda) = 1 - e^{-i\lambda}, \qquad |A_\Delta(\lambda)|^2 = (1 - e^{-i\lambda})(1 - e^{i\lambda}) = 2(1 - \cos\lambda).$$

- At $\lambda = 0$, the gain is zero: constant and trend-like movement are removed.
- At high frequencies, the gain is comparatively large.

Interpretation

Differencing is not just a stationarity trick. It is a filter that suppresses very low frequencies and emphasizes change.

Empirical use: In macroeconomics, differencing turns levels such as GDP or prices into growth or inflation rates; in finance, log-differencing turns prices into returns. The empirical implication is that we study short-run changes, not long-run levels.

Second and seasonal differencing

More aggressive filters arise from higher-order difference operators:

$$\Delta^2 = (1 - L)^2 = 1 - 2L + L^2,$$

$$\Delta_s = (1 - L^s), \qquad \Delta_s Y_t = Y_t - Y_{t-s}.$$

- Δ^2 removes linear deterministic trend components.
- Δ_s removes seasonal mean components repeating every s periods.
- Combined operators, such as $(1 - L)(1 - L^{12})$, remove both trend-like and seasonal low frequencies.

Cost of stronger filtering

Each extra differencing step removes more low-frequency information and can amplify measurement noise or create moving-average type behavior.

Empirical use: Seasonal differencing is natural for monthly retail sales, electricity demand, tourism flows, or airline traffic. It removes recurring calendar patterns, but it can also hide economically meaningful seasonal adjustment failures.

Exponentially weighted moving average

A standard one-sided smoother is the EWMA:

$$X_t = (1-\alpha)\sum_{j=0}^{\infty} \alpha^j Y_{t-j}, \qquad 0 < \alpha < 1.$$

Equivalently,

$$X_t = (1-\alpha) Y_t + \alpha X_{t-1}.$$

- Recent observations receive more weight.
- The recursion is real-time feasible and computationally cheap.
- Larger α means a longer memory and smoother output.

Textbook point

EWMA is often attractive in prediction and monitoring because future observations are unavailable, even though one-sided smoothing generally has more boundary bias than two-sided smoothing.

Empirical use: Finance uses EWMA ideas for volatility monitoring and risk systems such as RiskMetrics-style variance forecasts. Engineering and quality control use EWMA charts to detect small persistent shifts in a process mean.

One-sided equal-weight filter and Bollinger-style logic

Another practical smoother is the one-sided equal-weight filter:

$$X_t = \frac{1}{n+1} \sum_{j=0}^{n} Y_{t-j}.$$

- It uses only current and past observations.
- It is easy to explain to non-technical audiences.
- It appears in reporting conventions such as 7-day moving averages and in practitioner tools such as Bollinger-style bands.

But there is a cost

Because the weights are one-sided, the smoother reacts more slowly at turning points and typically suffers more from boundary bias than a two-sided smoother.

Empirical use: Rolling means are common in dashboards because they are transparent: seven-day averages, rolling revenue, rolling volatility, and Bollinger-style bands are easy to explain to decision makers. The trade-off is delayed reaction.

Real-time versus retrospective filtering

Retrospective analysis

- Can use future observations.
- Symmetric filters often look cleaner.
- Appropriate for historical decomposition.

Real-time analysis

- Must use only information available at date t.
- One-sided filters are feasible.
- Revision risk is unavoidable.

$$\hat{s}_{t|T} \neq \hat{s}_{t|t}$$

in general, even before we move to formal state-space notation.

Lesson

When a filtered series looks impressive, always ask whether it is a two-sided historical reconstruction or a real-time feasible estimate.

Empirical use: This distinction matters for recession monitoring, central-bank nowcasting, and trading systems: a filter that identifies the peak only after future data arrive is useful historically but not as a live decision rule.

Endpoint loss and boundary bias

Symmetric filters with window $2m+1$ lose observations at the sample edges:

$$X_t = \sum_{j=-m}^{m} a_j Y_{t-j} \quad \text{is unavailable for } t = 1, \ldots, m \text{ and } t = T-m+1, \ldots, T.$$

- Two-sided business-cycle filters often discard endpoints.
- If we shorten the window near the boundary, the effective filter changes with t.
- One-sided fixes are feasible but may alter timing and smoothness substantially.

Interpretive warning

Many revisions in trend or cycle estimates are revisions of the filter, not only revisions of the economy.

Time-domain summary before frequency analysis

1. A filter is a weighted transformation of the data.
2. The sign, magnitude, and support of the weights determine smoothing, differencing, and timing behavior.
3. One-sided filters are feasible in real time; symmetric filters are often cleaner but retrospective.
4. Endpoint behavior is part of the method, not a minor afterthought.

Next step

To understand exactly *which periodicities* are being emphasized or suppressed, we now translate the same filters into the frequency domain.

Why move to the frequency domain?

The time-domain coefficients $\{a_j\}$ tell us how observations are weighted. The frequency-domain representation tells us how cycles of different lengths are treated.

- Slow trend-like movement corresponds to low frequencies.
- Rapid oscillation corresponds to high frequencies.
- Business-cycle extraction is naturally expressed as keeping some frequencies and removing others.

Same filter, new language

Frequency-domain analysis does not replace time-domain analysis. It tells the same story in a form better suited for understanding trend, cycle, and timing distortion.

Transfer function

For a linear filter $a(L) = \sum_j a_j L^j$, define

$$A(\lambda) = \sum_{j=-\infty}^{\infty} a_j e^{-ij\lambda}, \qquad -\pi \leq \lambda \leq \pi.$$

- $A(\lambda)$ is generally complex valued.
- Its modulus controls amplitude attenuation.
- Its argument controls timing shift.

$$A(\lambda) = G(\lambda) e^{-i\phi(\lambda)},$$

where $G(\lambda) = |A(\lambda)|$ is gain and $\phi(\lambda)$ is phase.

Spectrum of the filtered output

If the input has spectral density $f_Y(\lambda)$, then

$$f_X(\lambda) = |A(\lambda)|^2 f_Y(\lambda).$$

- The filter acts multiplicatively on the spectrum.
- Frequencies with gain near one are preserved.
- Frequencies with gain near zero are strongly attenuated.

Practical payoff

Once $G(\lambda)$ is known, we know immediately whether the filter behaves like a low-pass, high-pass, or band-pass device.

Gain, phase, and zero-phase filters

$$A(\lambda) = G(\lambda)e^{-i\phi(\lambda)}.$$

- **Gain** $G(\lambda)$**:** how much of frequency λ survives.
- **Phase** $\phi(\lambda)$**:** how much the filter shifts timing at frequency λ.
- **Symmetric filters:** $a_j = a_{-j}$ imply $A(\lambda)$ is real and phase distortion can be zero or only sign changes.

Why symmetry is attractive

Symmetric filters often avoid timing distortion, but they need future observations. That is why beautiful historical filters can be unusable in real time.

Low-pass, high-pass, and band-pass logic

Low-pass

$$G(\lambda) \approx 1$$

for small $|\lambda|$.

High-pass

$$G(\lambda) \approx 0$$

for small $|\lambda|$.

Band-pass

$$G(\lambda) \approx 1$$

only on a chosen band.

- Trend extraction is usually low-pass.
- Differencing is a high-pass idea.
- Business-cycle extraction is usually band-pass.

Empirical use: A useful empirical translation is: low-pass estimates potential output or long-run productivity; high-pass emphasizes growth, returns, or innovations; band-pass targets cycles such as 2–8 year business-cycle movements.

Ideal low-pass filter

An ideal low-pass filter with cutoff λ_c has gain

$$G_{\mathrm{LP}}(\lambda) = \begin{cases} 1, & |\lambda| \le \lambda_c, \\ 0, & |\lambda| > \lambda_c. \end{cases}$$

The corresponding time-domain weights are infinite:

$$a_j = \frac{1}{2\pi} \int_{-\lambda_c}^{\lambda_c} e^{ij\lambda}\, d\lambda = \begin{cases} \lambda_c/\pi, & j = 0, \\ \dfrac{\sin(j\lambda_c)}{\pi j}, & j \neq 0. \end{cases}$$

Why this matters

The ideal filter is mathematically clean but operationally impossible in finite real-time samples because it is two-sided and infinite.

Empirical use: Low-pass thinking appears in macro trend estimation, engineering denoising, and signal extraction from noisy sensors. Its implication is that slow movement is treated as signal and high-frequency variation as noise.

Ideal band-pass filter

To isolate a cycle band $[\underline{\lambda}, \bar{\lambda}]$, the ideal gain is

$$G_{\mathrm{BP}}(\lambda) = \begin{cases} 1, & \underline{\lambda} \le |\lambda| \le \bar{\lambda}, \\ 0, & \text{otherwise.} \end{cases}$$

The ideal weights are

$$a_0 = \frac{\bar{\lambda} - \underline{\lambda}}{\pi}, \qquad a_j = \frac{\sin(j\bar{\lambda}) - \sin(j\underline{\lambda})}{\pi j}, \quad j \neq 0.$$

- The target band is usually chosen through periodicities P_u, P_l:

$$\underline{\lambda} = \frac{2\pi}{P_u}, \qquad \bar{\lambda} = \frac{2\pi}{P_l}.$$

- Business-cycle filters use exactly this logic.

Empirical use: Band-pass filters are useful when the object is not the trend itself but a range of fluctuations, such as business cycles, inventory cycles, seasonal sub-bands, or vibration bands in engineering data.

Finite approximation and truncation

Because the ideal band-pass filter has infinitely many coefficients, practice replaces it by a truncated finite filter:

$$X_t = \sum_{j=-K}^{K} a_j Y_{t-j}.$$

- Larger K gives a closer approximation to the ideal gain.
- Larger K also increases endpoint loss and revision sensitivity.
- In finite samples, a filter is always a compromise between theoretical sharpness and practical feasibility.

Same trade-off in a new form

Sharper frequency selection usually requires longer support in the time domain.

Baxter–King filter coefficients

The Baxter–King filter is a truncated symmetric approximation to the ideal band-pass filter:

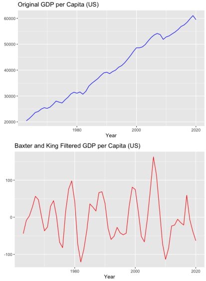
$$X_t = \sum_{j=-K}^{K} \phi_j Y_{t-j}, \qquad \phi_0 = \frac{\bar{\lambda} - \underline{\lambda}}{\pi},$$

$$\phi_j = \frac{\sin(j\bar{\lambda}) - \sin(j\underline{\lambda})}{\pi j}, \qquad j \neq 0.$$

- Usually the coefficients are adjusted to sum to zero so very low frequencies are removed.
- For quarterly business-cycle work, typical retained periodicities are between 6 and 32 quarters.
- The filter is symmetric, so it is inherently retrospective.

Empirical use: Baxter–King is designed for macro business-cycle measurement: it keeps medium-run fluctuations and removes both trend and high-frequency noise. The practical cost is endpoint loss and a retrospective interpretation.

Baxter–King style business-cycle extraction

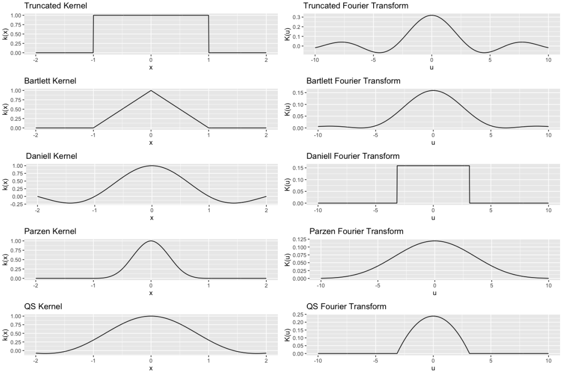


How to read the figure

The band-pass filtered series keeps medium-frequency business-cycle movement while removing very slow trend and very rapid noise. But it also sacrifices observations near the sample ends because the symmetric weights need future data.

Empirical use: In applied macro papers, this type of output is often read as a cyclical component of GDP, employment, investment, or credit. It should not be interpreted as an observed variable; it is a filter-defined object.

Kernels and Fourier transforms



Why this figure belongs here

The figure reminds us that smoothing kernels in the time domain and gain functions in the frequency domain are Fourier pairs. The same mathematical object can be described either as weights on observations or as a shape applied to frequencies.

Differencing seen in the frequency domain

For the difference filter,

$$A_\Delta(\lambda) = 1 - e^{-i\lambda}, \qquad |A_\Delta(\lambda)|^2 = 2(1 - \cos\lambda) = 4\sin^2(\lambda/2).$$

- At $\lambda = 0$, the gain is zero.
- Near $\lambda = 0$, the gain behaves like λ^2, so very low frequencies are strongly suppressed.
- Around π, the gain is large, so rapid alternation is emphasized.

Interpretation

This is why differencing can remove trend-like persistence but also make a series look much noisier.

Phase distortion and turning-point timing

If the phase $\phi(\lambda)$ is not zero, the filter shifts the timing of oscillations:

$$A(\lambda) = G(\lambda)e^{-i\phi(\lambda)}.$$

- A causal smoother often introduces delay at turning points.
- A symmetric smoother can avoid delay but uses future data.
- Two filters with similar gain can still tell different economic stories because of phase distortion.

Economic interpretation

If a filter systematically delays peaks and troughs, it may be poor for real-time monitoring even when it looks excellent in ex post charts.

Empirical use: Phase matters for lead-lag analysis: a smoother that delays peaks can make a financial indicator look less useful for timing recessions, risk build-ups, or engineering faults.

Why symmetric filters are usually retrospective

Suppose

$$X_t = \sum_{j=-m}^{m} a_j Y_{t-j}, \qquad a_j = a_{-j}.$$

Then

- the filter uses future values $Y_{t+1}, \ldots, Y_{t+m}$;
- zero-phase or near-zero-phase behavior is feasible;
- the price is that X_t is unavailable in real time.

Practical moral

Never confuse a good *historical decomposition* with a good *real-time indicator*. Those are different filtering problems.

There is no universally best deterministic filter

- Smoother filters reduce short-run noise but blur timing.
- Sharper band-pass filters isolate chosen periodicities but lose endpoints.
- One-sided filters are feasible in real time but suffer more from delay and boundary problems.
- Frequency-selective filters can distort relationships across variables if applied mechanically.

Filter choice depends on the question

Are we estimating a trend? Isolating business-cycle frequencies? Producing a real-time monitoring index? Removing a stochastic trend before modeling? The correct filter depends on the target.

Design checklist before filtering

Before choosing a deterministic filter, ask:

1. What is the object of interest: trend, cycle, short-run change, or real-time signal?
2. Is the output meant for ex post description or for real-time use?
3. How much endpoint loss or revision is acceptable?
4. Is it acceptable to distort phase or should turning points be preserved?
5. Will the filtered output be used later in another model?

Bridge to Hour 3

These questions lead naturally to smoothing methods that can be interpreted as filters but are often introduced as nonparametric estimators of a trend function.

From linear filters to smooth trend estimation

The textbook treats many smoothers as estimates of an unknown function $g(\cdot)$ in

$$Y_t = g(t/T) + \sigma(t/T)\varepsilon_t.$$

- Here $g(\cdot)$ is a smooth deterministic trend.
- The disturbance ε_t captures short-run stationary variation.
- Filtering becomes a way to estimate the low-frequency component g.

Connection

A simple moving average is one smoother. Kernel regression and spline smoothing are richer ways to estimate the same low-frequency object.

Nonparametric local trend model

The textbook's local-trend formulation is

$$Y_t = g\left(\frac{t}{T^\kappa}\right) + \sigma\left(\frac{t}{T^\kappa}\right)\varepsilon_t, \qquad t = 1, \ldots, T,$$

with ε_t weakly dependent and stationary.

- $\kappa = 1$ is the standard compact-domain case.
- $g(\cdot)$ is smooth but otherwise unspecified.
- The task is to estimate a function, not just a finite-dimensional parameter.

Why rates slow down

Because we estimate an entire curve locally, convergence is slower than the usual $\sqrt{T}$ parametric rate.

Nadaraya–Watson kernel estimator

For $u \in (0,1)$, the kernel estimator is

$$\widehat{g}(u) = \frac{\sum_{t=1}^{T} K\left(\frac{u - t/T}{h}\right) Y_t}{\sum_{t=1}^{T} K\left(\frac{u - t/T}{h}\right)} \approx \frac{1}{Th} \sum_{t=1}^{T} K\left(\frac{u - t/T}{h}\right) Y_t.$$

- K is the kernel.
- h is the bandwidth.
- Nearby observations receive more weight than distant ones.

$$\widehat{g}(u) = \sum_{t=1}^{T} w_{Tt}(u) Y_t, \qquad \sum_{t=1}^{T} w_{Tt}(u) = 1.$$

Empirical use: Kernel trend estimation is useful when the trend is smooth but not well described by a straight line or a small parametric model, for example long-run productivity, climate/economic indicators, or smooth volatility paths.

Common kernel choices

$$K_{\mathrm{tri}}(x) = 1 - |x|, \quad |x| \leq 1; \qquad K_{\mathrm{Epa}}(x) = 0.75(1 - x^2), \quad |x| \leq 1;$$

$$K_{\mathrm{Gauss}}(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}; \qquad K_{\mathrm{DE}}(x) = \frac{1}{2} e^{-|x|}.$$

- Triangular and Epanechnikov kernels have compact support.
- Gaussian and double-exponential kernels have unbounded support.
- Kernel choice matters less than bandwidth, but support and smoothness still affect finite-sample appearance.

Textbook remark

The Epanechnikov kernel is optimal in a standard MSE sense, while Gaussian-type kernels often look visually smoother because the implied estimator is infinitely differentiable.

Asymptotic distribution of the kernel trend estimator

For interior u, under smoothness and mixing conditions,

$$\sqrt{Th}\big(\widehat{g}(u) - g(u)\big) \Longrightarrow N\big(b(u), v(u)\big),$$

with

$$b(u) = c^{1/2} g''(u)\mu_2(K), \qquad v(u) = \sigma^2(u)\,\mathrm{lrvar}(\varepsilon)\|K\|_2^2,$$

when $h \to 0$ and $Th^5 \to c > 0$.

- Bias depends on curvature $g''(u)$.
- Variance depends on local scale and long-run variance.
- The convergence rate is $T^{2/5}$ when $h \asymp T^{-1/5}$.

Bandwidth choice and the bias–variance trade-off

$$\mathrm{MSE}(\widehat{g}(u)) \approx \underbrace{h^4\{g''(u)\}^2\mu_2(K)^2}_{\text{squared bias}} + \underbrace{\frac{\sigma^2(u)\,\mathrm{lrvar}(\varepsilon)\|K\|_2^2}{Th}}_{\text{variance}}.$$

- Larger h: smoother curve, lower variance, more bias.
- Smaller h: more local detail, lower bias, higher variance.
- The textbook emphasizes that bandwidth matters more than the exact kernel family.

Same lesson as before

Filtering never eliminates tuning choices. It changes which tuning parameter drives the approximation.

Empirical use: The bandwidth is an empirical judgment about scale: a larger bandwidth asks for broad movements, while a smaller bandwidth treats local wiggles as potentially meaningful signal.

Two-sided equal weights as a uniform kernel

The textbook explicitly connects moving averages and kernel smoothing.

$$K(s) = \frac{1}{2}1\{|s| \leq 1\}$$

corresponds to a two-sided equal-weight smoother.

$$\widehat{T}_t = \sum_{j=-n}^{n} w_j Y_{t-j}, \qquad w_j = \frac{1}{2n+1}.$$

- This minimizes variance for a fixed window.
- It does not minimize asymptotic MSE because it does not control bias as well as smoother kernels.
- It is still widely used because it is transparent and easy to explain.

Empirical use: Equal-weight smoothers are often chosen in public reporting because the rule is auditable. That is useful for communication, even when more efficient smoothers exist statistically.

One-sided smoothers, prediction, and boundary bias

For forecasting or monitoring, future observations are unavailable, so one-sided smoothing is unavoidable. Examples include:

$$w(s) = 1\{-1 \le s \le 0\}, \qquad w(s) = e^{-s} 1\{s > 0\}$$

after suitable normalization.

- One-sided equal weights correspond to rolling means.
- Exponential weights give EWMA-type smoothers.
- The leading bias is typically of order h, not h^2, because symmetry is broken.

Meaning

The real-time feasibility of one-sided filters comes with a mathematical price: larger boundary bias and more timing distortion.

Empirical use: This is the default situation in nowcasting, online monitoring, and trading: the analyst must trade smoothness against delay using only data already observed.

Local linear correction and negative weights

The textbook points out that better one-sided smoothers can be constructed by imposing

$$\int w(s)\,ds = 1, \qquad \int s\,w(s)\,ds = 0.$$

- These moment restrictions reduce first-order bias.
- Some improved one-sided filters therefore use **negative weights**.
- Local linear regression is another standard boundary-bias correction.

$$w(s) = 4 + 6s \quad \text{or} \quad w(s) = 3 - 6s^2$$

on $[-1, 0]$ satisfy the textbook's moment conditions after normalization.

Important intuition

Negative weights are not a bug. They are often the price of removing first-order boundary bias in real-time smoothers.

Empirical use: Local linear correction is useful near sample endpoints and policy-relevant turning points because it reduces the tendency of one-sided smoothers to lag behind a changing trend.

Smoothing spline objective

A cubic smoothing spline chooses $\widehat{g}_\lambda$ to minimize

$$Q_\lambda(g) = \sum_{t=1}^{T} \big(Y_t - g(t/T)\big)^2 + \lambda \int \{g''(u)\}^2 \, du.$$

- The first term rewards fit to the data.
- The second term penalizes roughness.
- Larger λ means a smoother estimated trend.

Interpretation

Spline smoothing is a filtering problem written as penalized least squares rather than as an explicit weighted average.

Empirical use: Spline smoothers are common when analysts want a smooth trend estimate but prefer to state the tuning choice as a roughness penalty. This is useful for yield curves, potential output, and engineering calibration curves.

Discrete second-difference form and the HP filter

In discrete form, the smoothing-spline problem becomes

$$Q_\lambda(g) = (Y - g)'(Y - g) + \lambda g' D g,$$

where D is the second-difference penalty matrix. Equivalently,

$$\widehat{g}_\lambda = (I + \lambda D)^{-1} Y.$$

- The penalty is based on $\Delta^2 = (1 - L)^2$.
- Large second differences mean a rough trend path.
- This is the logic behind the Hodrick–Prescott filter.

Empirical use: The HP filter is widely used for trend-cycle decomposition in macroeconomics, especially output gaps and postwar business-cycle facts. Its empirical meaning is controlled by the smoothness penalty.

Interpreting the smoothing parameter λ

$$\lambda \downarrow 0 \Longrightarrow \widehat{g}_\lambda \approx Y \qquad \text{and} \qquad \lambda \uparrow \infty \Longrightarrow \widehat{g}_\lambda \text{ approaches a straight line.}$$

- Small λ: fit the data closely, little smoothing.
- Large λ: smooth aggressively, treat curvature as costly.
- Macro practice often uses conventional values of λ, but those values embody strong filtering choices.

Why this is not innocuous

Changing λ changes the trend-cycle decomposition, the timing of turning points, and sometimes the apparent persistence of the "cycle" itself.

Empirical use: In policy work, a larger λ can make potential output look smoother and the output gap look more cyclical; a smaller λ can absorb more short-run movement into the trend.

Generated-regressor caveat after detrending

Suppose

$$Y_t = g(t/T) + \varepsilon_t, \qquad \widehat{\varepsilon}_t = Y_t - \widehat{g}(t/T).$$

Then the empirical cdf of the residuals,

$$\widehat{F}_T(e) = \frac{1}{T}\sum_{t=1}^{T} 1\{\widehat{\varepsilon}_t \le e\},$$

satisfies

$$\sqrt{T}\big(\widehat{F}_T(e) - F(e)\big) = \frac{1}{\sqrt{T}}\sum_{t=1}^{T}\big(1\{\varepsilon_t \le e\} - F(e)\big) + f(e)\frac{1}{\sqrt{T}}\sum_{t=1}^{T}\varepsilon_t + o_p(1).$$

Meaning

Preliminary detrending changes later inference. Filtered or detrended data are generated regressors, not raw observations.

Why HP-style detrending can distort dynamics

The textbook is deliberately cautious:

- Detrended series can distort dynamic relationships across variables.
- A cycle extracted mechanically is not automatically an economically meaningful cycle.
- Subsequent parametric modeling must account for first-stage smoothing uncertainty.

Connection to Sims' warning

If we over-filter, we may create apparent cycles or remove economically meaningful low-frequency comovement. Filtering is a modeling choice, not merely data cleaning.

Robust filtering: local median smoothing

Mean smoothers can be sensitive to outliers. A robust alternative is the local median:

$$\widehat{\alpha}(u) = \arg\min_{\alpha} \sum_{t=1}^{T} K\left(\frac{u - t/T}{h}\right) |Y_t - \alpha|.$$

- The local median is less sensitive to large isolated shocks.
- It is useful when spikes would distort mean-based smoothers badly.
- Robustness matters especially when filtering is used for descriptive signal extraction.

Empirical use: Robust smoothers are useful for financial transaction data, sensor series, survey outliers, and data-release errors, where a single spike should not redefine the underlying signal.

Hampel filter

Let $\widehat{g}_k(t/T)$ be a moving local median and let

$$s_k(t) = 1.4826 \times \text{median}\{|Y_s - \widehat{g}_k(t/T)| : s \in [t-k, t+k]\}.$$

The Hampel filter replaces Y_t by

$$X_t = \begin{cases} Y_t, & |Y_t - \widehat{g}_k(t/T)| < c\, s_k(t), \\ \widehat{g}_k(t/T), & |Y_t - \widehat{g}_k(t/T)| \geq c\, s_k(t). \end{cases}$$

- $s_k(t)$ is a local MAD scale.
- The constant 1.4826 calibrates MAD to Gaussian standard deviation.
- The filter keeps ordinary observations but pulls extreme local outliers back toward the local median.

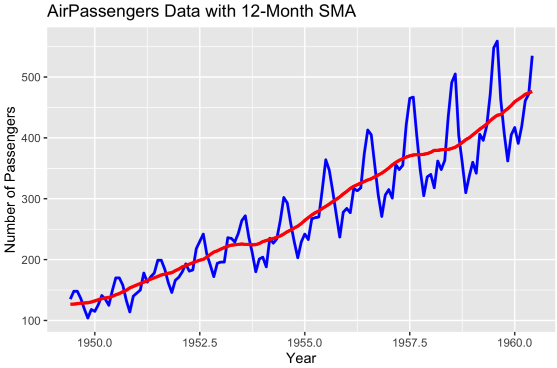
Empirical use: Hampel filters are especially natural for engineering and data-cleaning tasks with impulse noise: they identify local anomalies while preserving the ordinary path of the series.

Practical R workflow for deterministic filters

1. Plot the raw series and ask whether the target is trend, cycle, or real-time smoothing.
2. Try a transparent baseline first: rolling mean, differencing, or seasonal differencing.
3. Inspect the implied loss of endpoints and the sensitivity to window length or λ.
4. If using kernel or spline smoothing, report the bandwidth or penalty parameter explicitly.
5. If filtered data enter a later model, treat them as generated regressors and interpret later standard errors with care.

```
library(zoo)
ma7 <- rollmean(x, k = 7, align = "right", fill = NA)
dx <- diff(x)
hp <- mFilter::hpfilter(x, freq = 1600)
bk <- mFilter::bkfilter(x, pl = 6, pu = 32, nfix = 12)
```

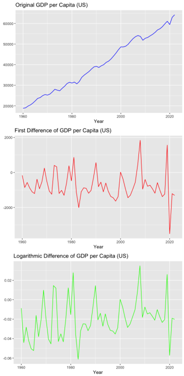
Moving-average smoothing in practice



What the picture shows

The moving average suppresses high-frequency noise and reveals a smoother underlying path. But it also blurs turning points and uses future data when the filter is symmetric.

Differencing and filter-based trend removal



Practical lesson

High-pass transformations such as differencing remove low-frequency movement quickly, but the output is not a neutral "cleaned" version of the original series. It is a different series with a different economic interpretation.

Lecture 16 takeaways

1. A linear filter is a weighted transformation of a series, and lag-polynomial notation makes the algebra transparent.
2. Time-domain weights and frequency-domain gain / phase describe the same operation from two viewpoints.
3. Moving averages are low-pass smoothers; differencing is a high-pass filter; Baxter–King is a band-pass approximation.
4. One-sided filters are real-time feasible but suffer more from bias and timing distortion.
5. Kernel, spline, HP, and robust smoothers can all be understood as filtering devices with tuning parameters and inferential costs.

Preview of Lecture 17

Lecture 17 takes the next step:

1. instead of choosing deterministic weights directly, we will specify a latent state and an observation equation;
2. filtering will become conditional inference about a hidden process;
3. the Kalman filter will emerge as the recursive Gaussian updating rule implied by that model.

Conceptual handoff

Lecture 16 asks, "what does this chosen filter do?" Lecture 17 asks, "what model implies the optimal recursive filter?"