Lecture 17 — State-Space Models, Kalman Filtering, and Likelihood-Based Signal Extraction

Chapter 7: general setup, recursion, and local level inference

Jiajing Sun

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

Econometrics and Time Series Methods
Spring 2026

Why Lecture 17 matters

Lecture 16 treated filters as operators chosen by the analyst. Lecture 17 changes the viewpoint.

$$y_t = H_t \alpha_t + \eta_t, \qquad \alpha_t = F_t \alpha_{t-1} + \omega_t.$$

New question

What if the object of interest is a latent state generated by a probabilistic model, and the observed series is only a noisy measurement of that state?

- Then filtering becomes model-based inference.
- The Kalman filter becomes the central recursive algorithm.
- Likelihood and signal extraction are unified in one framework.

Inferential shift

We are no longer choosing a smoother by eye. We are inferring a hidden process from a fully specified stochastic system.

Where Lecture 17 fits in the course

- Lecture 16 introduced filtering in the time and frequency domains.
- **Lecture 17** turns filtering into a latent-variable problem.
- Later applications include stochastic volatility, missing-data problems, and time-varying parameter models.

Teaching logic

Externally chosen filters are useful, but state-space models let the data determine the optimal recursive filter once a probabilistic structure is specified.

Bridge from Lecture 16

Deterministic filters start from coefficients $\{a_j\}$. State-space methods start from economic or statistical laws for α_t and then derive the implied filter.

Learning goals

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

1. write down state and observation equations;
2. distinguish prediction, filtering, and smoothing;
3. explain the role of Gaussian conditioning in Kalman filtering;
4. derive the recursion for the local level model;
5. interpret the Kalman gain as a signal-to-noise weight;
6. explain the prediction-error decomposition of the likelihood;
7. discuss initialization, missing data, and practical diagnostics.

Three-hour plan

Hour 1

State-space models: general setup and examples.

Hour 2

Kalman filter recursion.

Hour 3

Local level model and likelihood-based inference in state-space systems.

From chosen filters to latent-state models

The key difference from Lecture 15 is:

- a moving average filter is imposed externally;
- a state-space filter is implied by a model for hidden states and noisy observations.

$$\hat{s}_t^{\text{filter}} = \sum_{j=-k}^{k} a_j y_{t-j} \qquad \text{versus} \qquad \hat{\alpha}_{t|t} = \mathbb{E}(\alpha_t \mid \mathcal{F}_t).$$

New perspective

The signal is not just a smoothed series. It is an unobserved stochastic process with its own dynamics.

Why this matters

Once the model is specified, uncertainty about the signal can be quantified formally through conditional variances and likelihoods.

Why latent states arise so often

Many empirical problems naturally involve unobserved components:

- trend and cycle in macroeconomic data;
- latent volatility in financial returns;
- measurement-error contaminated observations;
- time-varying coefficients in predictive regressions;
- missing observations in otherwise regular panels or time series.

Canonical examples

$$y_t = \tau_t + c_t + \varepsilon_t, \qquad \beta_t = \beta_{t-1} + u_t,$$

where τ_t is a latent trend, c_t a latent cycle, and β_t a latent coefficient path.

General nonlinear state-space model

The most general setup is

$$\alpha_t = F(\alpha_{t-1}, \omega_t), \qquad y_t = H(\alpha_t, \eta_t),$$

where:

- α_t is the latent state;
- y_t is the observed variable;
- ω_t is state disturbance;
- η_t is observation noise.

Two structural ideas

- The state is typically first-order Markov once the full latent vector is defined.
- Conditional on the current state, today's observation is independent of earlier observations.

Linear Gaussian state-space form

The classical Kalman filter applies to

$$\alpha_t = F_t\alpha_{t-1} + B_t x_t + \omega_t, \qquad \omega_t \sim N(0, Q_t),$$

$$y_t = H_t\alpha_t + \eta_t, \qquad \eta_t \sim N(0, R_t).$$

- F_t governs state dynamics.
- H_t maps the hidden state into the observed space.
- Q_t and R_t determine the signal-to-noise balance.

Standard assumptions

Usually ω_t and η_t are serially independent, mutually independent, and independent of the initial state. Those assumptions keep the recursion transparent.

State noise versus measurement noise

State noise

ω_t changes the latent process itself.

Measurement noise

η_t affects only how the state is observed.

- Large state noise means the hidden component is genuinely volatile.
- Large measurement noise means the observations are unreliable.

Diagnostic intuition

If Q_t is large relative to R_t, the latent state is allowed to move a lot. If R_t dominates, the filter smooths aggressively because it treats the data as contaminated measurements.

Information sets and conditional densities

Let

$$\mathcal{F}_t = \sigma(y_1, \dots, y_t).$$

Then the three central objects are:

$$\begin{aligned} f(\alpha_t \mid \mathcal{F}_{t-1}) &\quad \text{prediction,} \\ f(\alpha_t \mid \mathcal{F}_t) &\quad \text{filtering,} \\ f(\alpha_t \mid \mathcal{F}_T) &\quad \text{smoothing.} \end{aligned}$$

Point estimation versus full distribution

In practice we often report $\alpha_{t|t} = \mathbb{E}(\alpha_t \mid \mathcal{F}_t)$, but the full conditional density also delivers uncertainty bands through $P_{t|t} = \mathrm{Var}(\alpha_t \mid \mathcal{F}_t)$.

Prediction, filtering, and smoothing

- **Prediction:** what do we think the state is before seeing y_t?
- **Filtering:** how does that belief change after seeing y_t?
- **Smoothing:** after the whole sample is observed, how do we revise earlier state estimates?

Key distinction

Filtering is real-time. Smoothing is retrospective.

$$\mathrm{Var}(\alpha_t \mid \mathcal{F}_T) \leq \mathrm{Var}(\alpha_t \mid \mathcal{F}_t) \leq \mathrm{Var}(\alpha_t \mid \mathcal{F}_{t-1}),$$

so using more information weakly reduces uncertainty.

Example: local level model

The simplest stochastic-trend model is

$$\alpha_t = \alpha_{t-1} + \omega_t, \qquad y_t = \alpha_t + \eta_t.$$

- The latent level follows a random walk.
- The data measure that level with noise.
- This one example already captures trend extraction, missing-data interpolation, and recursive updating.

Useful intuition

Taking first differences gives

$$\Delta y_t = \omega_t + \eta_t - \eta_{t-1},$$

so observed growth mixes genuine trend innovations with a moving-average measurement-noise term.

Example: AR model in state-space form

An AR(p) process can be written in companion form by stacking lags into a state vector:

$$\alpha_t = (y_t, y_{t-1}, \ldots, y_{t-p+1})'.$$

$$\alpha_t = \begin{pmatrix} \phi_1 & \phi_2 & \cdots & \phi_p \\ 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix} \alpha_{t-1} + \begin{pmatrix} \varepsilon_t \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \qquad y_t = \begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix} \alpha_t.$$

- The state equation becomes first order in a higher-dimensional state.
- The observation equation simply selects the first element.

Takeaway

State-space form is not exotic. Many standard time-series models already live inside it.

Example: MA model in state-space form

Moving-average models can also be written in state-space form by letting the state contain current and lagged shocks.

$$y_t = \varepsilon_t + \theta_1 \varepsilon_{t-1}, \qquad \alpha_t = \begin{pmatrix} \varepsilon_t \\ \varepsilon_{t-1} \end{pmatrix}.$$

- This is useful because MA models are awkward for direct recursive updating.
- State-space form makes likelihood evaluation and forecasting straightforward.

Practical point

This is one reason ARMA models are commonly estimated by state-space likelihood methods rather than by manipulating long covariance matrices directly.

Example: time-varying parameter regression

Suppose

$$y_t = x_t'\beta_t + \eta_t, \qquad \beta_t = \beta_{t-1} + \omega_t.$$

- The coefficients are now states.
- The regression becomes a state-space model.
- Kalman filtering gives recursive estimates of the evolving coefficients.

Economic interpretation

Small Q means coefficients evolve slowly and the model is close to a constant-parameter regression. Large Q allows rapid structural change.

Example: stochastic volatility idea

For financial returns, one often wants the latent volatility rather than a filtered mean.

- The volatility process itself is hidden.
- The observation equation is nonlinear or non-Gaussian.
- Kalman filtering still motivates approximations and extensions.

$$r_t = \exp(h_t/2)\varepsilon_t, \qquad h_t = \mu + \phi(h_{t-1} - \mu) + u_t.$$

Forward look

The state-space language is the natural gateway to stochastic-volatility modeling.

Why Gaussian conditioning matters

The Kalman filter works because linear transformations of Gaussian vectors remain Gaussian.

Core fact

If the joint distribution of the state and the new observation is Gaussian, then the conditional mean and conditional variance are available in closed form.

$$\mathrm{Var}(\alpha \mid y) = \Sigma_{\alpha\alpha} - \Sigma_{\alpha y}\Sigma_{yy}^{-1}\Sigma_{y\alpha}.$$

- That is why the filter is recursive.
- That is why the filter is computationally efficient.

Conditional moments of a joint normal vector

If

$$\begin{pmatrix} \alpha \\ y \end{pmatrix} \sim N\left(\begin{pmatrix} \mu_\alpha \\ \mu_y \end{pmatrix}, \begin{pmatrix} \Sigma_{\alpha\alpha} & \Sigma_{\alpha y} \\ \Sigma_{y\alpha} & \Sigma_{yy} \end{pmatrix} \right),$$

then

$$\mathbb{E}(\alpha \mid y) = \mu_\alpha + \Sigma_{\alpha y}\Sigma_{yy}^{-1}(y - \mu_y).$$

$$\mathrm{Var}(\alpha \mid y) = \Sigma_{\alpha\alpha} - \Sigma_{\alpha y}\Sigma_{yy}^{-1}\Sigma_{y\alpha}.$$

Why this is the whole Kalman filter in miniature

The posterior mean equals the prior mean plus a correction term, and the posterior variance equals prior variance minus the uncertainty resolved by observing y.

Notation for the Kalman filter

Let

$$\alpha_{t|t-1} = \mathbb{E}(\alpha_t \mid \mathcal{F}_{t-1}), \qquad P_{t|t-1} = \mathrm{Var}(\alpha_t \mid \mathcal{F}_{t-1}).$$

After observing y_t, define

$$\alpha_{t|t} = \mathbb{E}(\alpha_t \mid \mathcal{F}_t), \qquad P_{t|t} = \mathrm{Var}(\alpha_t \mid \mathcal{F}_t).$$

$$\hat{y}_{t|t-1} = H_t \alpha_{t|t-1}, \qquad \nu_t = y_t - \hat{y}_{t|t-1}.$$

Interpretation

$\alpha_{t|t-1}$ is the prior. $\alpha_{t|t}$ is the posterior.

Prediction step: mean

In the linear Gaussian model,

$$\alpha_{t|t-1} = F_t \alpha_{t-1|t-1} + B_t x_t.$$

$$\hat{y}_{t|t-1} = H_t F_t \alpha_{t-1|t-1} + H_t B_t x_t.$$

- This is just the model's dynamic forecast of the hidden state.
- No new observation has yet been used.

Prediction step: variance

The prediction variance evolves as

$$P_{t|t-1} = F_t P_{t-1|t-1} F_t' + Q_t.$$

- The previous posterior uncertainty is propagated through the transition matrix.
- State noise Q_t is added because the system itself is evolving randomly.

Law-of-total-variance intuition

Even if yesterday's state were known perfectly, fresh state noise would still create uncertainty today. That is exactly what the $+Q_t$ term records.

Innovation and forecast-error variance

The innovation is

$$\nu_t = y_t - H_t \alpha_{t|t-1}.$$

Its variance is

$$S_t = H_t P_{t|t-1} H_t' + R_t.$$

Why ν_t matters

It is the discrepancy between what the model predicted and what the data actually reported.

$$\mathbb{E}(\nu_t \mid \mathcal{F}_{t-1}) = 0, \qquad \mathrm{Var}(\nu_t \mid \mathcal{F}_{t-1}) = S_t.$$

Update step: mean

The filtered mean is

$$\alpha_{t|t} = \alpha_{t|t-1} + K_t v_t.$$

- Start from the prior.
- Add a correction proportional to the innovation.
- The proportionality factor is the Kalman gain.

Conditional-mean interpretation

This is the multivariate analog of linear projection: the state estimate changes only to the extent that forecast errors contain information about the hidden state.

Update step: variance

The filtered variance is

$$P_{t|t} = P_{t|t-1} - K_t S_t K_t'.$$

$$P_{t|t} = (I - K_t H_t) P_{t|t-1} (I - K_t H_t)' + K_t R_t K_t'.$$

- New data reduce uncertainty.
- The reduction is larger when the innovation is informative.

Useful numerical view

The second formula is the Joseph form. It is algebraically equivalent in exact arithmetic and often more stable in computation.

Kalman gain

The Kalman gain is

$$K_t = P_{t|t-1} H_t' S_t^{-1}.$$

$$\text{scalar case:} \qquad K_t = \frac{P_{t|t-1}}{P_{t|t-1} + R_t} \qquad \text{when } H_t = 1.$$

Interpretation

The gain tells us how strongly the filter reacts to the new observation relative to the model-based prior.

Signal-to-noise interpretation

- If the predicted state is very uncertain, the data deserve more weight.
- If the measurement is very noisy, the model prior deserves more weight.

Intuition

The Kalman gain is the dynamic weight balancing signal uncertainty and measurement noise.

Scalar benchmark

With $H_t = 1$, K_t behaves like a time-varying fraction

$$\frac{\text{signal uncertainty}}{\text{signal uncertainty} + \text{measurement noise}}.$$

When observation noise is large

If R_t is large:

- S_t is large;
- K_t is small;
- the filter trusts the prior more than the new observation.

Economic interpretation

Noisy data should not cause violent swings in the estimated latent state.

$$R_t \to \infty \qquad \Longrightarrow \qquad K_t \to 0.$$

When state noise is large

If Q_t is large:

- $P_{t|t-1}$ rises;
- K_t rises;
- the filter becomes more responsive to incoming data.

Interpretation

When the hidden state itself moves around a lot, the prior gets stale quickly.

$$Q_t \uparrow \quad \Longrightarrow \quad P_{t|t-1} \uparrow \quad \Longrightarrow \quad K_t \uparrow .$$

Recursive algorithm summary

Each date t repeats the same logic:

1. predict the state mean and variance;
2. compute the innovation and its variance;
3. update the mean and variance;
4. move to the next date.

Computational payoff

The whole sample need not be handled as one giant Gaussian vector. The recursion processes information one period at a time.

Algorithmic template

Prior → forecast error → posterior → next prior. That cycle is what makes the Kalman filter suitable for online updating.

Forecast errors and the Cholesky view

The innovation sequence ν_t does more than update the state.

- It orthogonalizes the information in the sample.
- It yields a lower-triangular decomposition of the covariance structure of y.
- It is therefore the foundation of likelihood evaluation.

Likelihood preview

For multivariate observations, the same innovation sequence produces

$$\ell(\theta) = -\frac{1}{2}\sum_{t=1}^{T}\left\{\log\det S_t + \nu_t' S_t^{-1}\nu_t\right\}$$

up to the constant $-\frac{nT}{2}\log(2\pi)$.

Local level model equations

The scalar local level model is

$$\alpha_t = \alpha_{t-1} + \omega_t, \qquad \omega_t \sim N(0, Q),$$

$$y_t = \alpha_t + \eta_t, \qquad \eta_t \sim N(0, R).$$

- The hidden level follows a random walk.
- The observed series measures that level with noise.

Key tuning ratio

The relative magnitude $q = Q/R$ determines how rough the latent path is allowed to be. Small q implies a very smooth signal; large q lets the level respond quickly.

Deriving the forecast error

For the local level model,

$$\alpha_{t|t-1} = \alpha_{t-1|t-1}, \qquad P_{t|t-1} = P_{t-1|t-1} + Q.$$

Therefore

$$\nu_t = y_t - \alpha_{t|t-1}, \qquad V_t = P_{t|t-1} + R.$$

Key observation

The innovation variance is just predicted state variance plus measurement noise variance.

$$\hat{y}_{t|t-1} = \alpha_{t|t-1}, \qquad \nu_t = y_t - \hat{y}_{t|t-1}.$$

Deriving the update

The Kalman gain becomes

$$K_t = \frac{P_{t|t-1}}{V_t}.$$

So the update is

$$\alpha_{t|t} = \alpha_{t|t-1} + K_t v_t, \qquad P_{t|t} = (1 - K_t) P_{t|t-1}.$$

Scalar intuition

Because $0 < K_t < 1$, the filtered level is a convex combination of the prior level and the current observation. The exact weight changes over time with uncertainty.

Deriving the next-step prediction

Because the state follows a random walk,

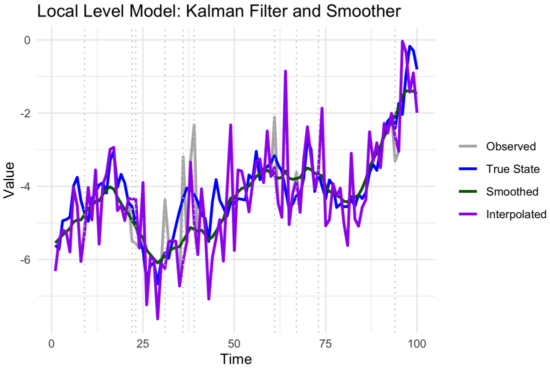
$$\alpha_{t+1|t} = \alpha_{t|t}, \qquad P_{t+1|t} = P_{t|t} + Q.$$

- Filtering uses y_t.
- Prediction for $t+1$ propagates uncertainty one period ahead.

Steady-state idea

If the model is time invariant and the sample is long enough, $P_{t|t-1}$ and K_t may converge to constants. Then the filter behaves like an exponentially weighted moving average with model-based weights.

Local level model: main illustration



Filtered versus smoothed states

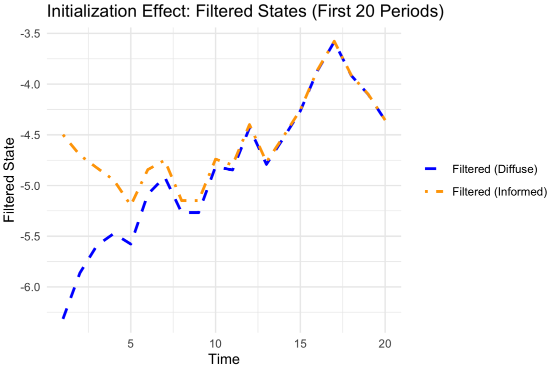
- Filtered states use data only up to date t.
- Smoothed states use future observations as well.
- Smoothed estimates are not feasible in real time, but they are often more accurate for retrospective analysis.

$$\alpha_{t|T} = \mathbb{E}(\alpha_t \mid \mathcal{F}_T), \qquad \alpha_{t|t} = \mathbb{E}(\alpha_t \mid \mathcal{F}_t).$$

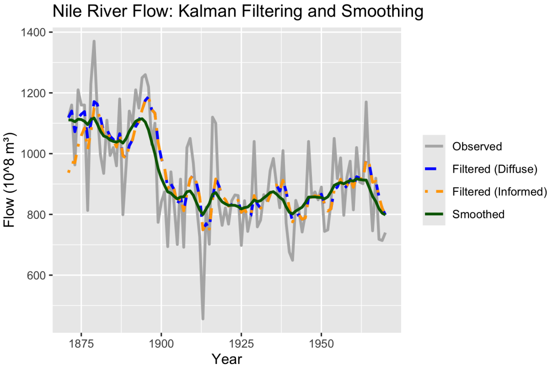
Accuracy ranking

Smoothing is an ex post device: it can improve historical state estimates, but it cannot be used for real-time decision making at date t.

Filtered and smoothed trajectories



Nile River example



Likelihood from prediction errors

Under Gaussianity,

$$\nu_t \mid \mathcal{F}_{t-1} \sim N(0, V_t).$$

Therefore the log-likelihood is

$$\ell(\theta) = -\frac{1}{2} \sum_{t=1}^{T} \left\{ \log(2\pi) + \log V_t + \frac{\nu_t^2}{V_t} \right\}.$$

Importance

The filter itself produces ν_t and V_t, so likelihood evaluation comes almost for free.

Estimation logic

Maximum likelihood chooses parameters that make forecast errors small after scaling by their model-implied variance. In other words, we fit the model by asking whether the innovations look like white noise.

Estimating Q and R

- Q controls state evolution.
- R controls measurement noise.
- Maximum likelihood chooses the pair that best explains the innovation sequence.

Interpretation

Estimating Q/R is effectively estimating how smooth the latent path should be relative to the noisiness of the data.

Identification intuition

If Q is pushed too low, almost all short-run variation is attributed to measurement error. If R is pushed too low, the latent state chases every wiggle in the data.

Informative initialization

If prior information exists, initialize with

$$\alpha_{1|0} \sim N(m_0, P_0),$$

using meaningful m_0 and P_0.

- This is helpful when the initial state has a clear economic interpretation.
- Early filtered states can depend noticeably on this choice.

When this matters most

Initialization is especially important in short samples or when the signal-to-noise ratio is low, because the data take longer to overwhelm the prior.

Diffuse initialization

When prior information is weak, a diffuse prior uses a very large P_0.

- The first few observations dominate the early state estimate.
- This is conceptually attractive but can be numerically delicate.

Practical compromise

Analysts often use a very large but finite P_0, or estimate the initial state jointly by maximum likelihood.

Technical note

Textbook implementations often rely on exact diffuse initialization so that the first few likelihood contributions are handled correctly rather than approximated crudely.

Missing observations

State-space models handle missing y_t elegantly.

- If y_t is missing, skip the update step.
- Keep propagating the state by the prediction step.
- When observations resume, the filter continues normally.

Why this matters

Irregularly spaced or partially missing data are common in practice.

$$y_t \text{ missing} \qquad \Longrightarrow \qquad \alpha_{t|t} = \alpha_{t|t-1}, \quad P_{t|t} = P_{t|t-1}.$$

Multi-step forecasting

Once the filtered state at time t is available, future forecasts are straightforward:

- propagate the state equation forward repeatedly;
- map the predicted states into predicted observations;
- accumulate uncertainty across horizons.

Difference from simple AR forecasting

Forecasting is now performed through latent-state dynamics rather than directly through the observed series alone.

Local-level formulas

For the local level model,

$$\alpha_{t+h|t} = \alpha_{t|t}, \qquad \mathrm{Var}(\alpha_{t+h} \mid \mathcal{F}_t) = P_{t|t} + hQ.$$

Observation-forecast uncertainty adds R on top of that state uncertainty.

Smoothing recursion

Filtering is forward-looking. Smoothing adds a backward pass.

- Future innovations revise earlier state estimates.
- Smoothed states are especially useful for structural decomposition and historical narrative analysis.

Practical meaning

After the entire sample is observed, we can often recover past latent states more accurately than was possible in real time.

$$\alpha_{t|T} = \alpha_{t|t} + J_t\left(\alpha_{t+1|T} - \alpha_{t+1|t}\right), \qquad J_t = P_{t|t}F'P_{t+1|t}^{-1}.$$

What smoothing adds beyond filtering

- Better historical estimates of hidden trends.
- Cleaner decomposition of observed movements into signal and noise.
- Improved retrospective interpretation of turning points.

But remember

Smoothing is not a forecasting device. It uses future information.

$$\mathrm{Var}(\alpha_t \mid \mathcal{F}_T) \leq \mathrm{Var}(\alpha_t \mid \mathcal{F}_t),$$

so the gain from smoothing is primarily a gain in historical precision.

Time-varying parameter models revisited

State-space methods are not only for trends.

- Dynamic betas in finance.
- Time-varying Phillips-curve coefficients.
- Evolving policy reaction functions.

General lesson

Whenever coefficients evolve over time, state-space form provides a natural inferential framework.

$$y_t = x_t'\beta_t + \eta_t, \qquad \beta_t = \beta_{t-1} + u_t,$$

so recursive coefficient tracking is just Kalman filtering with a vector state.

Stochastic volatility and quasi-linearization

True stochastic-volatility models are typically non-Gaussian.

- Log-squared transformations can produce approximate linear state-space forms.
- Gaussian-mixture approximations can recover conditional Gaussian structure.
- This is why Kalman ideas remain central even beyond the textbook linear-Gaussian case.

$$\log r_t^2 = h_t + \xi_t, \qquad h_t = \mu + \phi(h_{t-1} - \mu) + u_t,$$

where ξ_t is non-Gaussian. The Kalman filter can still be used approximately after suitable Gaussianization.

R workflow for state-space estimation

A typical workflow is:

1. choose a package such as `KFAS`, `dlm`, or `bsts`;
2. write down the state and observation equations;
3. estimate unknown variances by maximum likelihood;
4. inspect filtered states, smoothed states, and innovations;
5. compare the latent-state interpretation with raw filtered data from simpler deterministic filters.

Empirical discipline

The crucial step is not only obtaining a smooth state path. It is checking whether the implied innovation sequence and parameter estimates make substantive sense.

Diagnostics based on innovations

The innovation sequence should be scrutinized.

- Are innovations centered around zero?
- Is remaining autocorrelation small?
- Are innovation variances stable relative to model assumptions?

Why diagnostics matter

A badly specified state-space model can still produce smooth-looking states that are economically misleading.

$$z_t = \frac{v_t}{\sqrt{V_t}} \quad \text{should look approximately like white noise}$$

in the scalar Gaussian case.

Practical pitfalls

- Confusing measurement noise with genuine state variation.
- Over-smoothing by forcing Q too close to zero.
- Ignoring sensitivity to initialization in short samples.
- Treating filtered states as directly observed data.

Applied warning

A visually plausible latent path is not enough. Always ask whether the estimated variances, innovation diagnostics, and economic interpretation all point in the same direction.

Lecture 17 takeaways

- State-space models separate hidden dynamics from noisy observation.
- The Kalman filter is recursive Gaussian conditioning.
- The local level model is the simplest stochastic-trend example.
- The innovation sequence delivers both state updates and the likelihood.
- Initialization, missing data, and diagnostics are practical essentials.

One-line summary

Specify the hidden process, derive the recursive filter, estimate unknown variances from prediction errors, and then interpret the filtered or smoothed state as the model-based signal.

Where this leads next

The state-space framework prepares us for:

- stochastic-volatility applications;
- nonlinear and non-Gaussian extensions;
- simulation-based methods when Kalman filtering is not exact.

Bigger methodological picture

Lecture 17 is the point where time-series econometrics moves from deterministic signal extraction to explicitly modeling latent economic structure.