Lecture 11 — Multivariate Volatility Models and an Introduction to Nonparametric Methods

Chapter 4 → Chapter 5: covariance dynamics, kernel density estimation, and nonparametric regression

Jiajing Sun

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

Econometrics and Time Series Methods
Spring 2026

Why Lecture 11 is a transition lecture

Lecture 10 still worked within a mainly *univariate* volatility world. We allowed asymmetry, IGARCH persistence, and likelihood-based inference, but the core object was still a scalar conditional variance σ_t^2.

Lecture 11 makes two transitions:

Transition 1: scalar volatility $\rightarrow$ covariance dynamics

We move from one return series to a *vector* of returns and study the conditional covariance matrix Σ_t.

Transition 2: parametric recursions $\rightarrow$ flexible smoothing

We move from tightly parameterized models, like GARCH and DCC, to nonparametric estimators that learn shapes from the data.

Big econometric idea

The lecture asks two complementary questions: how do second moments move *jointly* over time, and what can we estimate when we do *not* want to impose a rigid functional form?

Learning goals

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

1. write down the conditional covariance matrix for a vector return process and explain why positive definiteness matters;
2. derive and interpret the EWMA covariance recursion and connect it to the IGARCH logic;
3. distinguish DVEC, BEKK, CCC, and DCC type multivariate volatility parameterizations;
4. explain the idea of Cholesky-based orthogonal shocks;
5. define the kernel density estimator and state the role of the kernel and the bandwidth;
6. derive the leading bias and variance orders of kernel density estimation;
7. explain the curse of dimensionality in multivariate smoothing;
8. define and interpret the Nadaraya–Watson and local polynomial estimators.

Practical plan for the three contact hours

Hour 1

Multivariate volatility: notation, EWMA covariance matrices, DVEC, BEKK, CCC, DCC, and orthogonalization.

Hour 2

Nonparametric density estimation: kernels, bandwidth, bias–variance trade-off, and multivariate kernel density estimation.

Hour 3

Nonparametric regression: Nadaraya–Watson, local polynomial regression, boundary bias, and R workflow.

How to read the lecture

The first part is still mainly about *conditional second moments*. The second and third parts are about *smoothing unknown objects* such as densities and regression functions.

From scalar variance to matrix volatility

In the univariate GARCH world, the volatility object is

$$\sigma_t^2 = \mathrm{Var}(\varepsilon_t \mid \mathcal{F}_{t-1}).$$

For a k-dimensional return vector $\boldsymbol{r}_t$, the natural analogue is

$$\boldsymbol{r}_t = \boldsymbol{\mu}_t + \boldsymbol{\varepsilon}_t, \qquad \boldsymbol{\Sigma}_t = \mathrm{Var}(\boldsymbol{\varepsilon}_t \mid \mathcal{F}_{t-1}).$$

- $\boldsymbol{\Sigma}_t$ is a $k \times k$ symmetric positive-definite matrix.
- The diagonal entries are conditional variances.
- The off-diagonal entries are conditional covariances.

Why the multivariate step matters

Portfolio risk, hedge ratios, value-at-risk, contagion, and diversification all depend on how *covariances and correlations* move over time.

Setup, notation, and the dimensionality problem

Write the innovation vector as

$$\boldsymbol{\varepsilon}_t = (\varepsilon_{1t}, \ldots, \varepsilon_{kt})', \qquad \boldsymbol{\Sigma}_t = \mathbb{E}(\boldsymbol{\varepsilon}_t \boldsymbol{\varepsilon}_t' \mid \mathcal{F}_{t-1}).$$

A useful factorization is

$$\boldsymbol{\varepsilon}_t = \boldsymbol{\Sigma}_t^{1/2} \boldsymbol{z}_t, \qquad \mathbb{E}(\boldsymbol{z}_t) = 0, \qquad \mathrm{Var}(\boldsymbol{z}_t) = \boldsymbol{I}_k.$$

The number of distinct entries in $\boldsymbol{\Sigma}_t$ is

$$\frac{k(k+1)}{2}.$$

- If $k = 2$, there are 3 distinct elements.
- If $k = 10$, there are 55 distinct elements.
- If $k = 50$, there are 1275 distinct elements.

Econometric implication

An unrestricted dynamic model for $\boldsymbol{\Sigma}_t$ becomes infeasible very quickly. Structure is not optional; it is essential.

Why conditional covariance matters in applications

Three standard objects depend directly on $\boldsymbol{\Sigma}_t$.

Portfolio variance

For weights $\boldsymbol{w}$, conditional portfolio risk is

$$\mathrm{Var}(\boldsymbol{w}'\boldsymbol{r}_t \mid \mathcal{F}_{t-1}) = \boldsymbol{w}'\boldsymbol{\Sigma}_t\boldsymbol{w}.$$

Dynamic hedge ratio

For two assets i and j, a natural hedge ratio is

$$\beta_{ij,t} = \frac{\mathrm{Cov}(\varepsilon_{it}, \varepsilon_{jt} \mid \mathcal{F}_{t-1})}{\mathrm{Var}(\varepsilon_{jt} \mid \mathcal{F}_{t-1})}.$$

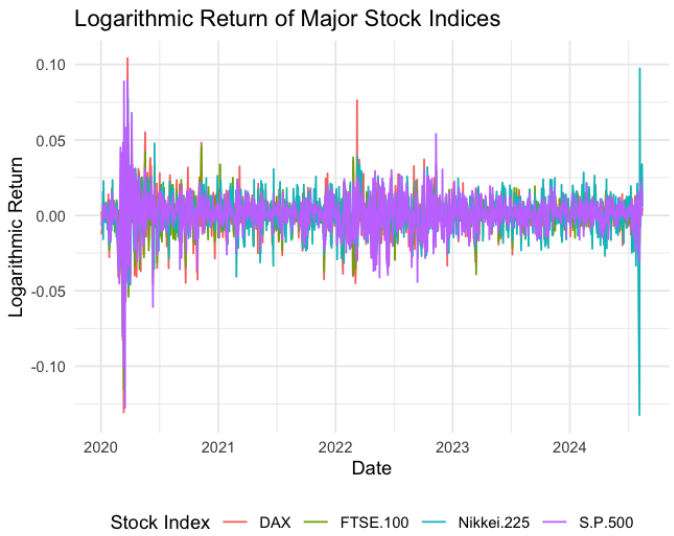
Conditional correlation

Correlation is obtained by standardizing covariance:

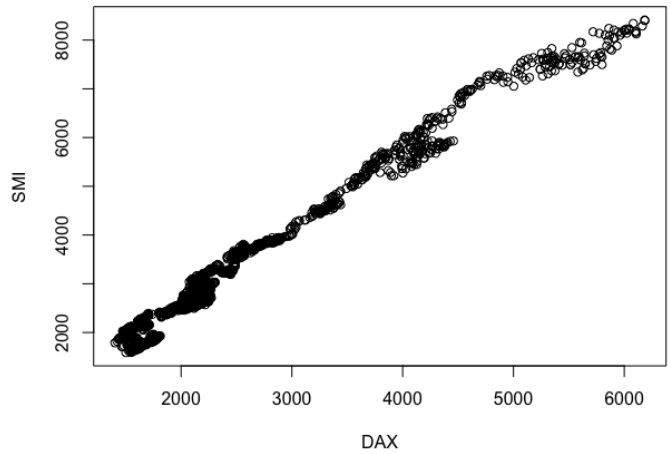
$$\rho_{ij,t} = \frac{\Sigma_{ij,t}}{\sqrt{\Sigma_{ii,t}\Sigma_{jj,t}}}.$$

So multivariate volatility is not just a theoretical extension of GARCH. It is the direct language of risk management and portfolio allocation.

Empirical motivation: comovement and co-volatility



Large equity markets display volatility bursts at similar times.



Return clouds are typically tilted, not spherical: covariance matters.

- Volatility clusters are often synchronized across markets.
- Correlations rise in stressed periods, reducing diversification benefits.
- A univariate volatility model cannot describe these joint movements.

Exponentially weighted covariance matrices: definition

A simple dynamic covariance estimator is the exponentially weighted moving average (EWMA):

$$\widehat{\boldsymbol{\Sigma}}_t = \frac{1-\lambda}{1-\lambda^{t-1}} \sum_{j=1}^{t-1} \lambda^{j-1} \boldsymbol{\varepsilon}_{t-j} \boldsymbol{\varepsilon}_{t-j}', \qquad 0 < \lambda < 1.$$

For large t, this is well approximated by the recursion

$$\widehat{\boldsymbol{\Sigma}}_t = (1-\lambda) \boldsymbol{\varepsilon}_{t-1} \boldsymbol{\varepsilon}_{t-1}' + \lambda \widehat{\boldsymbol{\Sigma}}_{t-1}.$$

- The newest outer product gets weight $(1-\lambda)$.
- Older observations are geometrically down-weighted.
- The same decay factor is imposed on every variance and covariance entry.

Connection to RiskMetrics

This is the matrix analogue of the univariate EWMA volatility estimator widely used in finance.

EWMA recursion, half-life, and the IGARCH connection

Element by element, the recursion says

$$\widehat{\Sigma}_{ij,t} = (1-\lambda)\varepsilon_{i,t-1}\varepsilon_{j,t-1} + \lambda\widehat{\Sigma}_{ij,t-1}.$$

This is formally similar to a scalar IGARCH(1,1) recursion: the newest shock product enters once, and the rest of the weight is carried over from the previous estimate.

Half-life of a shock

The weight on an observation h periods old is proportional to λ^h, so the half-life is

$$h_{1/2} = \frac{\log(1/2)}{\log(\lambda)}.$$

For $\lambda = 0.94$, the half-life is roughly 11 trading days.

- Large λ means slow decay and very persistent covariance estimates.
- Smaller λ means the estimator reacts more quickly to recent shocks.

From covariance matrices to correlation matrices

Given a positive-definite covariance matrix $\boldsymbol{\Sigma}_t$, define the diagonal matrix of conditional standard deviations

$$\boldsymbol{D}_t = \text{diag}\{\Sigma_{11,t}^{1/2}, \ldots, \Sigma_{kk,t}^{1/2}\}.$$

Then the correlation matrix is

$$\boldsymbol{R}_t = \boldsymbol{D}_t^{-1}\boldsymbol{\Sigma}_t\boldsymbol{D}_t^{-1}, \qquad \boldsymbol{\Sigma}_t = \boldsymbol{D}_t\boldsymbol{R}_t\boldsymbol{D}_t.$$

- $\boldsymbol{D}_t$ governs the scale of each individual series.
- $\boldsymbol{R}_t$ governs the contemporaneous dependence pattern.

Why this factorization is useful

It separates the modelling of *marginal volatilities* from the modelling of *correlations*. This is the basic idea behind constant-correlation and dynamic-correlation GARCH models.

EWMA covariance: strengths and limitations

Strengths

- very easy to compute;
- always positive semi-definite if initialized properly;
- responsive to recent market conditions;
- useful as a benchmark or an initialization device.

Limitations

- same decay parameter for all entries;
- no explicit asymmetry or leverage effect;
- no rich cross-market spillovers;
- mostly descriptive rather than fully structural.

That motivates multivariate GARCH structures in which the covariance matrix itself follows a parameterized dynamic recursion.

DVEC model: direct entry-by-entry dynamics

A diagonal VEC or DVEC(m, s) specification writes

$$\boldsymbol{\Sigma}_t = \boldsymbol{C} + \sum_{i=1}^{m} \boldsymbol{A}_i \odot (\boldsymbol{\varepsilon}_{t-i}\boldsymbol{\varepsilon}'_{t-i}) + \sum_{j=1}^{s} \boldsymbol{B}_j \odot \boldsymbol{\Sigma}_{t-j},$$

where $\odot$ denotes the Hadamard (elementwise) product.
For a bivariate DVEC$(1, 1)$ model, the entries behave like

$$\Sigma_{11,t} = c_{11} + a_{11}\varepsilon^2_{1,t-1} + b_{11}\Sigma_{11,t-1},$$

$$\Sigma_{22,t} = c_{22} + a_{22}\varepsilon^2_{2,t-1} + b_{22}\Sigma_{22,t-1},$$

$$\Sigma_{12,t} = c_{12} + a_{12}\varepsilon_{1,t-1}\varepsilon_{2,t-1} + b_{12}\Sigma_{12,t-1}.$$

- Interpretation is straightforward.
- Each variance and covariance gets its own scalar GARCH-type recursion.

Why DVEC is intuitive but problematic

What DVEC gets right

It is easy to see how each shock product affects the matching variance or covariance component.

What DVEC does not guarantee

Positive definiteness of Σ_t is *not automatic*. A matrix can have sensible-looking entries and still fail to be a valid covariance matrix.

- Parameter counting still grows quickly with dimension.
- Spillovers are only partially captured through matching entries.
- The econometric challenge is to keep flexibility *and* preserve matrix admissibility.

This leads naturally to BEKK

BEKK imposes a quadratic matrix structure that preserves positive definiteness by construction.

BEKK(1, 1) model

The basic BEKK recursion is

$$\Sigma_t = \boldsymbol{C}\boldsymbol{C}' + \boldsymbol{A}'\varepsilon_{t-1}\varepsilon_{t-1}'\boldsymbol{A} + \boldsymbol{B}'\Sigma_{t-1}\boldsymbol{B},$$

where $\boldsymbol{C}$ is typically lower triangular and $\boldsymbol{A}, \boldsymbol{B}$ are $k \times k$ parameter matrices.

Why positive definiteness is automatic

For any nonzero vector $\boldsymbol{x}$,

$$\boldsymbol{x}'\Sigma_t\boldsymbol{x} = \|\boldsymbol{C}'\boldsymbol{x}\|^2 + (\varepsilon_{t-1}'\boldsymbol{A}\boldsymbol{x})^2 + \|\Sigma_{t-1}^{1/2}\boldsymbol{B}\boldsymbol{x}\|^2 > 0,$$

provided $\boldsymbol{C}$ has full rank.

- The BEKK form is more restrictive algebraically,
- but that restriction buys us admissibility of the covariance matrix.

Interpreting the BEKK matrices

The BEKK recursion separates three roles:

$$\boldsymbol{CC}' \quad \text{baseline covariance level,}$$

$$\boldsymbol{A}'\boldsymbol{\varepsilon}_{t-1}\boldsymbol{\varepsilon}_{t-1}'\boldsymbol{A} \quad \text{reaction to new shocks,}$$

$$\boldsymbol{B}'\boldsymbol{\Sigma}_{t-1}\boldsymbol{B} \quad \text{volatility persistence.}$$

- Diagonal entries of $\boldsymbol{A}$ and $\boldsymbol{B}$ govern own-series volatility response and persistence.
- Off-diagonal entries allow *cross-market spillovers*: a shock in asset j can change asset i's conditional variance.

Trade-off

Compared with DVEC, BEKK is safer from a covariance-matrix standpoint, but harder to estimate and interpret as dimension grows.

Correlation-based parameterizations: CCC

A popular alternative is to factor the covariance matrix as

$$\boldsymbol{\Sigma}_t = \boldsymbol{D}_t \boldsymbol{R}_t \boldsymbol{D}_t, \qquad \boldsymbol{D}_t = \operatorname{diag}\{\sigma_{1t}, \ldots, \sigma_{kt}\}.$$

The **constant conditional correlation** (CCC) model assumes

$$\boldsymbol{R}_t = \boldsymbol{R} \qquad \text{for all } t.$$

Then each variance σ_{it}^2 can follow its own univariate GARCH-type recursion, while

$$\operatorname{Cov}(\varepsilon_{it}, \varepsilon_{jt} \mid \mathcal{F}_{t-1}) = R_{ij}\sigma_{it}\sigma_{jt}.$$

- The model is parsimonious.
- It allows time-varying variances.
- But it forces the correlation pattern itself to stay fixed.

CCC is often a good baseline, but in stressed financial periods correlations clearly change, sometimes dramatically.

Dynamic conditional correlation (DCC)

DCC keeps the factorization

$$\Sigma_t = D_t R_t D_t,$$

but now allows R_t to vary over time. First standardize the shocks:

$$z_t = D_t^{-1} \varepsilon_t.$$

Then define the auxiliary matrix

$$Q_t = (1 - a - b)\bar{Q} + a z_{t-1} z_{t-1}' + b Q_{t-1}, \qquad a, b \geq 0, \qquad a + b < 1.$$

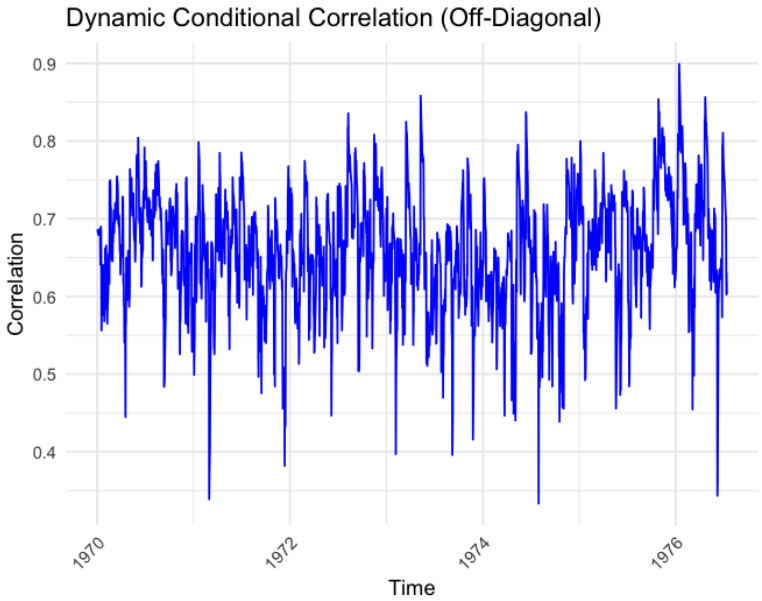
Normalize it to obtain a correlation matrix:

$$R_t = J_t^{-1} Q_t J_t^{-1}, \qquad J_t = \text{diag}\{q_{11,t}^{1/2}, \ldots, q_{kk,t}^{1/2}\}.$$

Interpretation

a measures the immediate impact of recent standardized shocks on correlations, while b measures persistence in the correlation dynamics.

Illustration: dynamic conditional correlation



- DCC is attractive because it keeps the correlation dynamics low-dimensional.
- The plot shows that correlations are not constant: they rise and fall with market conditions.
- In crisis periods, correlations often jump upward, weakening diversification.

Econometric lesson

Time-varying covariance is not only about individual variances. The entire dependence structure can change through time.

Cholesky decomposition and orthogonal shocks

Because Σ_t is positive-definite, it admits a Cholesky-type factorization

$$\Sigma_t = L_t G_t L_t',$$

where L_t is lower triangular with ones on the diagonal and G_t is diagonal with positive entries.
Define

$$b_t = L_t^{-1} \varepsilon_t.$$

Then

$$\mathrm{Var}(b_t \mid \mathcal{F}_{t-1}) = G_t,$$

so the components of b_t are conditionally uncorrelated.

- This is useful for building multivariate models from simpler univariate pieces.
- It is also useful for structural interpretation and orthogonalization.
- The ordering of variables matters because Cholesky decompositions are triangular.

R block: a compact multivariate-volatility workflow

```
# 1. Obtain returns and fit a mean model
library(quantmod)
library(BEKKs)
# library(rmgarch) # for DCC-type models

# eps: T x k matrix of demeaned returns or VAR residuals
lambda <- 0.94
Sigma <- vector("list", nrow(eps))
Sigma[[1]] <- cov(eps)

# 2. EWMA recursion
for (t in 2:nrow(eps)) {
  eprev <- matrix(eps[t-1,], ncol = 1)
  Sigma[[t]] <- (1-lambda) * (eprev %*% t(eprev)) + lambda * Sigma[[t-1]]
}

# 3. BEKK example (conceptual)
# bekk_spec <- bekk_spec()
# bekk_fit <- bekk_fit(bekk_spec, eps)

# 4. DCC example (conceptual)
# uspec <- ugarchmultispec(replicate(k, ugarchspec(), simplify = FALSE))
# dspec <- dccspec(uspec = uspec, dccOrder = c(1,1), distribution = "mvnorm")
# dfit <- dccfit(dspec, data = eps)
```

The empirical workflow is: estimate or filter the conditional mean first, obtain innovation vectors, then model their covariance matrix recursively.

Takeaways from multivariate volatility and transition to nonparametrics

The multivariate-volatility part gives us a set of structured recursions for the second moment matrix:

- EWMA for a simple benchmark,
- DVEC for entrywise dynamics,
- BEKK for positive-definite matrix recursions,
- CCC/DCC for volatility–correlation decompositions,
- Cholesky-based approaches for orthogonal shocks.

Why move on?

These models are powerful, but they are still highly *parametric*. Next we ask a different question: what if the object of interest is an unknown density or regression function and we do not want to impose a rigid global shape?

Why nonparametric methods enter econometrics

Parametric models are powerful when the assumed functional form is close to reality. But they can mislead us when the shape of the object of interest is unknown.

Examples of unknown objects

- the unconditional density of returns;
- the conditional mean $r(x) = \mathbb{E}(Y \mid X = x)$;
- the conditional variance as a function of state variables;
- nonlinear predictive relationships.

Nonparametric idea

Instead of specifying a small finite-dimensional parameter vector, we estimate the object locally from nearby observations.

Nonparametric methods reduce functional-form risk, but they pay for that flexibility with slower convergence rates and bandwidth-selection problems.

Kernel density estimator: basic definition

Suppose $X_1, \ldots, X_T$ have marginal density $g(x)$. The univariate kernel density estimator is

$$\widehat{g}(x) = \frac{1}{T}\sum_{t=1}^{T} K_h(x - X_t), \qquad K_h(u) = \frac{1}{h}K\left(\frac{u}{h}\right).$$

Equivalently,

$$\widehat{g}(x) = \frac{1}{Th}\sum_{t=1}^{T} K\left(\frac{x - X_t}{h}\right).$$

- $K(\cdot)$ is the kernel: it tells us how to weight observations by distance.
- $h > 0$ is the bandwidth: it tells us what counts as *nearby*.

Interpretation

Every observation contributes a small bump centered at its own location. The estimator adds those bumps and normalizes them into a smooth density curve.

What makes a valid second-order kernel?

A standard second-order kernel satisfies

$$\int K(u)\,du = 1, \qquad \int uK(u)\,du = 0, \qquad \int u^2 K(u)\,du = C_K < \infty,$$

and also

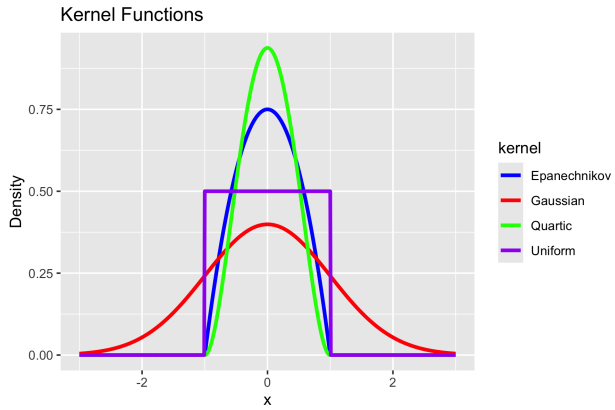
$$\int K(u)^2\,du = D_K < \infty.$$

- The first condition normalizes the estimator.
- The zero first moment removes first-order bias terms in the interior.
- The second moment and squared-integrability conditions control bias and variance constants.

Important practical point

For density and regression estimation, the exact kernel shape matters much less than the bandwidth. Kernel choice is usually second-order; bandwidth choice is first-order.

Common kernel functions



- Uniform kernel: equal weight inside a neighborhood.
- Gaussian kernel: unbounded support with smooth tails.
- Epanechnikov and quartic kernels: compact support and bell-shaped weighting.

Key intuition

A kernel is a *local weighting rule*: smoother, more peaked kernels down-weight distant observations more aggressively than flatter ones.

Histogram as a special kernel estimator

If we use the uniform kernel

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

then

$$\widehat{g}(x) = \frac{1}{2Th}\sum_{t=1}^{T} 1(|X_t - x| \leq h).$$

This counts the proportion of observations in the window $[x-h, x+h]$ and rescales by the window width.

Interpretation

A histogram is therefore a very simple kernel density estimator: it uses local counting with equal weights inside each bin.

- Histograms are intuitive but rough.
- Smooth kernels replace abrupt bin edges by gradually decaying weights.

Bandwidth intuition: under- vs over-smoothing

The bandwidth h determines the bias–variance trade-off.

Small h

- very local fit;
- low bias;
- high variance;
- wiggly estimate.

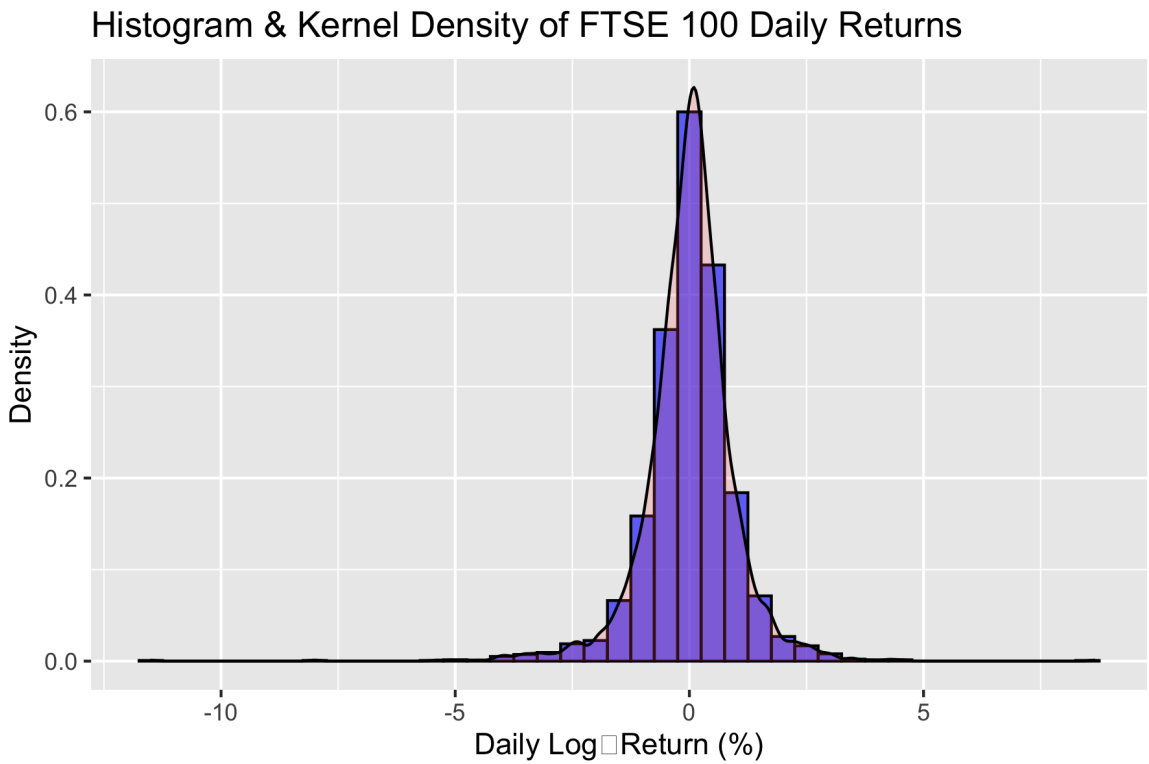
Large h

- broad averaging;
- low variance;
- high bias;
- oversmoothed estimate.

This is the central tuning problem

Kernel choice is often relatively unimportant, but the bandwidth can completely change the estimator.

Example: histogram and kernel density for FTSE returns



- Financial returns are often peaked and heavy-tailed.
- They may also be skewed, especially in stressed samples.
- A kernel density estimate lets the data reveal these features without forcing Gaussian symmetry.

Bias–variance decomposition of the density estimator

For a fixed interior point x,

$$\widehat{g}(x) - g(x) = \left[\mathbb{E}\widehat{g}(x) - g(x)\right] + \left[\widehat{g}(x) - \mathbb{E}\widehat{g}(x)\right].$$

Therefore

$$\mathrm{MSE}\big(\widehat{g}(x)\big) = \mathrm{Bias}^2\big(\widehat{g}(x)\big) + \mathrm{Var}\big(\widehat{g}(x)\big).$$

- The bias is deterministic and reflects smoothing distortion.
- The variance is stochastic and reflects sampling noise.

Goal of asymptotic analysis

Derive the leading orders of these two terms and then choose h to balance them.

Leading bias expansion of kernel density estimation

For an interior point x, a Taylor expansion gives

$$\mathbb{E}\widehat{g}(x) = \int K(u)g(x+hu)\,du = g(x) + \frac{1}{2}h^2 C_K g''(x) + o(h^2),$$

where

$$C_K = \int u^2 K(u)\,du.$$

Hence

$$\mathrm{Bias}\big(\widehat{g}(x)\big) = \frac{1}{2}h^2 C_K g''(x) + o(h^2).$$

- Interior bias is of order h^2 for a second-order kernel.
- The curvature term $g''(x)$ tells us where smoothing distorts the density most.

Leading variance expansion and effective sample size

Under independence, or under weak enough dependence,

$$\mathrm{Var}\big(\widehat{g}(x)\big) = \frac{1}{Th} g(x) D_K + o\left(\frac{1}{Th}\right), \qquad D_K = \int K(u)^2 \, du.$$

Interpretation

The term Th behaves like the *effective number of observations* inside a neighborhood of width $2h$ around x.

- If h becomes smaller, the local sample size falls and variance rises.
- If T increases, local sample size rises and variance falls.

Combining the bias and variance expansions yields

$$\mathrm{MSE}(\widehat{g}(x)) \approx \frac{1}{Th} g(x) D_K + \frac{1}{4} h^4 C_K^2 [g''(x)]^2.$$

Optimal bandwidth and convergence rate

Balancing the leading variance term $(Th)^{-1}$ and the leading squared-bias term h^4 gives

$$h_{\text{opt}} \asymp T^{-1/5}.$$

Substituting this order back into the MSE yields

$$\text{MSE}\big(\widehat{g}(x)\big) \asymp T^{-4/5}.$$

So the pointwise estimation error is typically of order

$$\widehat{g}(x) - g(x) = O_p(T^{-2/5}).$$

Key comparison

The nonparametric rate $T^{-2/5}$ is slower than the parametric root-T rate $T^{-1/2}$. Flexibility is bought at the price of slower convergence.

Boundary problems and dependence

Near the boundary of the support, the kernel cannot spread symmetrically across both sides of x. As a result,

- interior bias is of order $O(h^2)$,
- boundary bias is typically only of order $O(h)$.

Why this happens

The kernel puts weight on points outside the support where no data exist, so the local averaging becomes one-sided.

For time-series data, dependence complicates the variance analysis, but under suitable stationarity and mixing assumptions the same leading orders often survive, up to modified constants.

Practical message

Boundary correction and bandwidth choice are usually more important empirically than choosing among reasonable second-order kernels.

Multivariate kernel density estimation and the curse of dimensionality

For a d-dimensional vector $\boldsymbol{X}_t = (X_{1t}, \ldots, X_{dt})'$, the product-kernel estimator is

$$\widehat{f}(\boldsymbol{x}) = \frac{1}{T} \sum_{t=1}^{T} \prod_{i=1}^{d} K_h(x_i - X_{it}).$$

Its leading orders become

$$\text{Bias}(\widehat{f}(\boldsymbol{x})) = O(h^2), \qquad \text{Var}(\widehat{f}(\boldsymbol{x})) = O\left(\frac{1}{Th^d}\right).$$

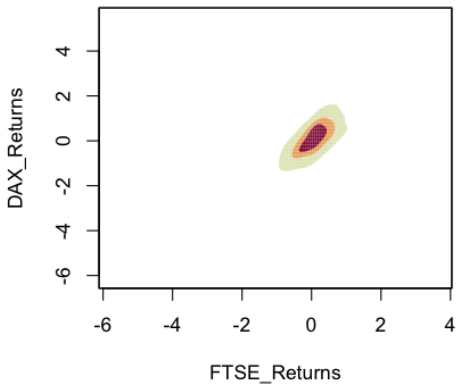
Hence

$$\text{MSE}(\widehat{f}(\boldsymbol{x})) = O\left(\frac{1}{Th^d} + h^4\right), \qquad h_{\text{opt}} \asymp T^{-1/(4+d)}.$$

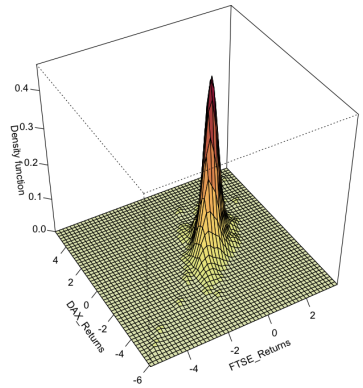
Curse of dimensionality

The convergence rate deteriorates to $T^{-4/(4+d)}$. As d rises, local neighborhoods become sparse and nonparametric estimation becomes data-hungry.

Example: multivariate KDE for joint returns



Filled contour view of the joint density.



3D view of the same smoothed joint density.

- Multivariate smoothing is useful for visualizing dependence and tail concentration.
- But it becomes difficult quickly as dimension rises.

Transition

Density estimation studies the whole distribution. Regression estimation studies the conditional mean function.

Why nonparametric regression?

In regression, we often want to estimate

$$r(x) = \mathbb{E}(Y \mid X = x)$$

without imposing a global linear or polynomial form. This is attractive when the true relationship is nonlinear, state-dependent, or otherwise unknown.

Parametric versus nonparametric logic

A linear model says the shape is known up to a few coefficients. A nonparametric model says the shape itself must be learned from local data patterns.

- less model-specification risk,
- more flexibility,
- slower convergence and more tuning choices.

Regression setup and the target object

Consider the model

$$Y_t = r(X_t) + \varepsilon_t, \qquad \mathbb{E}(\varepsilon_t \mid X_t) = 0, \qquad \mathrm{Var}(\varepsilon_t \mid X_t = x) = \sigma^2(x).$$

Then the conditional mean function is

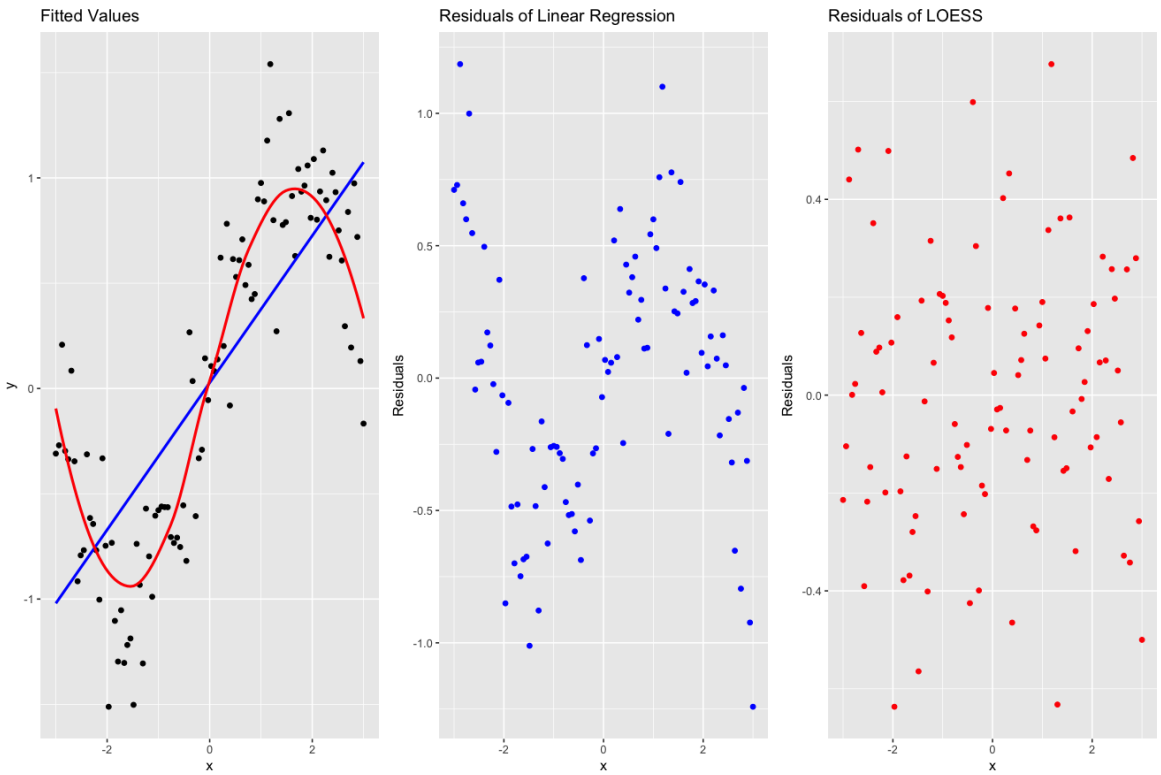
$$r(x) = \mathbb{E}(Y_t \mid X_t = x).$$

- In a parametric regression, $r(x)$ is forced into a specific class, such as $\beta_0 + \beta_1 x$.
- In a nonparametric regression, $r(x)$ is estimated locally from nearby observations.

Econometric interpretation

Nonparametric regression is a local method for recovering the systematic part of Y as a function of X.

Illustration: linear fit versus nonparametric fit



- The linear fit cannot adapt to curvature.
- The local smoother tracks the nonlinear pattern much more closely.
- Residuals from the nonparametric fit often look less structured.

Nadaraya–Watson estimator: definition

The Nadaraya–Watson (NW) estimator is

$$\widehat{r}(x) = \frac{\widehat{m}(x)}{\widehat{g}(x)},$$

where

$$\widehat{m}(x) = \frac{1}{T}\sum_{t=1}^{T} Y_t K_h(x - X_t), \qquad \widehat{g}(x) = \frac{1}{T}\sum_{t=1}^{T} K_h(x - X_t).$$

Equivalently,

$$\widehat{r}(x) = \frac{\sum_{t=1}^{T} Y_t K_h(x - X_t)}{\sum_{t=1}^{T} K_h(x - X_t)}.$$

Interpretation

This is a locally weighted average of the Y_t values, with larger weights placed on observations whose X_t values are close to x.

Weight representation and local averaging

Define the normalized weights

$$\widehat{W}_t(x) = \frac{K_h(x - X_t)}{\sum_{s=1}^{T} K_h(x - X_s)}, \qquad \sum_{t=1}^{T} \widehat{W}_t(x) = 1.$$

Then the NW estimator becomes

$$\widehat{r}(x) = \sum_{t=1}^{T} \widehat{W}_t(x)\, Y_t.$$

- This form makes the intuition transparent.
- Nonparametric regression is not mysterious: it is just *adaptive local averaging*.

Role of the kernel and bandwidth

The kernel determines how weight decays with distance; the bandwidth determines how large the neighborhood is.

Regressogram as a special case of NW

If the kernel is uniform,

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

then the NW estimator reduces to

$$\widehat{r}(x) = \frac{\sum_{t=1}^{T} Y_t 1(|X_t - x| \leq h)}{\sum_{t=1}^{T} 1(|X_t - x| \leq h)}.$$

Interpretation

This is simply the average of the responses whose regressors fall inside the local window $[x - h, x + h]$.

This special case is often called a **regressogram**. It is easy to understand, but smooth kernels typically behave better because they avoid discontinuous jumps in the fitted curve.

Nadaraya–Watson as local constant least squares

The NW estimator can also be derived as the solution to a weighted least-squares problem:

$$\min_r \sum_{t=1}^{T} (Y_t - r)^2 K_h(x - X_t).$$

The first-order condition is

$$\sum_{t=1}^{T} (Y_t - \widehat{r}(x)) K_h(x - X_t) = 0,$$

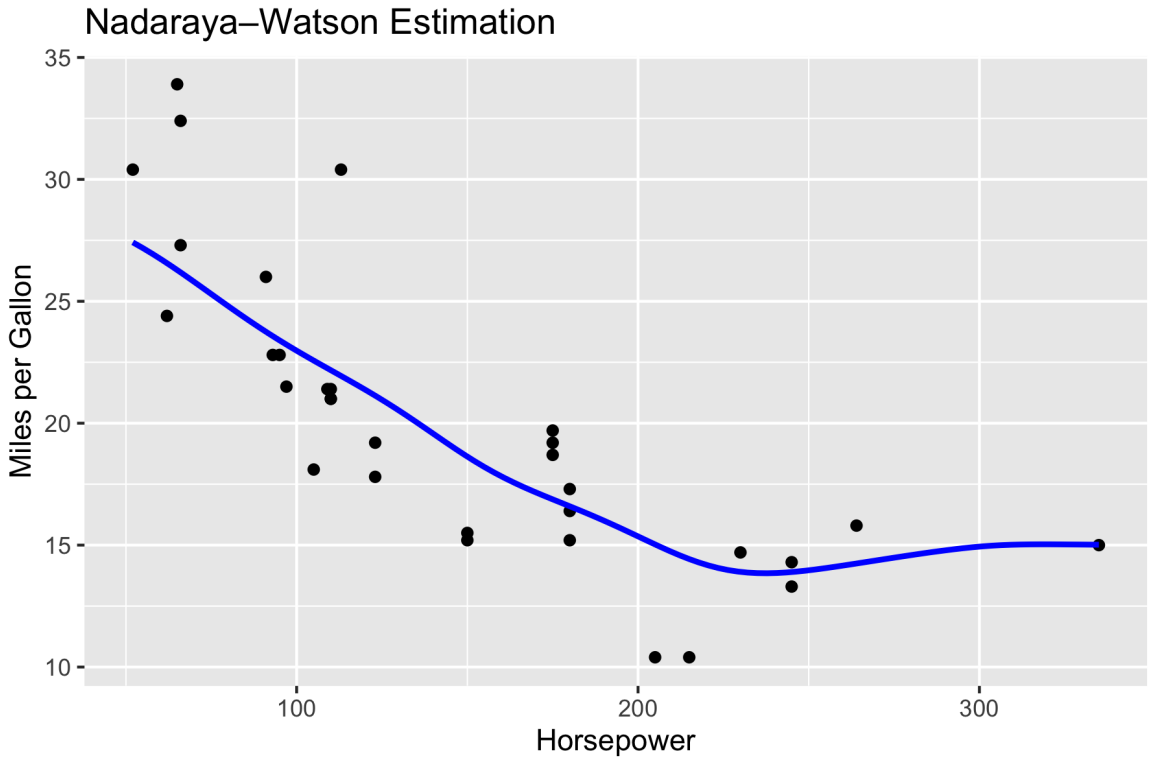
which implies

$$\widehat{r}(x) = \frac{\sum_{t=1}^{T} Y_t K_h(x - X_t)}{\sum_{t=1}^{T} K_h(x - X_t)}.$$

Why this interpretation matters

NW is the *local constant* estimator. It fits a constant inside each neighborhood and moves that neighborhood along the x-axis.

Example: Nadaraya–Watson fit for horsepower and MPG



- The fitted curve tracks the nonlinear relationship between horsepower and fuel efficiency.
- The local fit can bend without imposing a global polynomial.
- This is exactly why nonparametric methods are attractive in exploratory econometric work.

Bias and variance of the NW estimator

For interior points, the leading terms look like

$$\operatorname{Var}\left(\widehat{r}(x)\right) \approx \frac{1}{Th} \frac{\sigma^2(x)}{g(x)} D_K,$$

and

$$\operatorname{Bias}\left(\widehat{r}(x)\right) \approx \frac{h^2}{2} \left[r''(x) + \frac{2r'(x)g'(x)}{g(x)} \right] C_K.$$

- The variance behaves like $(Th)^{-1}$, again reflecting effective local sample size.
- The bias depends not only on the curvature of $r(x)$, but also on the slope of the design density $g(x)$.

Econometric point

Nonparametric regression bias depends on both the target function and the distribution of the regressor.

Boundary bias of the NW estimator

At boundary points, NW performs noticeably worse than in the interior.

- In the interior, the leading bias is of order $O(h^2)$.
- Near the boundary, the leading bias is typically only of order $O(h)$.

Why NW struggles at the boundary

Because the fit is local and symmetric, but near the edge there are no observations on one side of the target point. The local constant fit cannot compensate for that asymmetry.

This is the main motivation for local polynomial regression

Local linear and higher-order local polynomial fits correct boundary distortions much better than the local constant estimator.

Local polynomial idea: replace a local constant by a local Taylor approximation

Suppose $r(z)$ is smooth near x. Then locally we can write

$$r(z) \approx \sum_{j=0}^{p} \alpha_j(x)(z-x)^j, \qquad \alpha_j(x) = \frac{r^{(j)}(x)}{j!}.$$

Instead of fitting a constant in each neighborhood, we fit a local polynomial.

- $p = 0$ gives the NW estimator.
- $p = 1$ gives the local linear estimator.
- $p = 2$ gives the local quadratic estimator.

Why this helps

A local line can tilt to match the slope of the regression function near the boundary, so it reduces one-sided smoothing bias.

Local polynomial weighted least squares

At each target point x, estimate

$$\boldsymbol{\alpha}(x) = \big(\alpha_0(x), \ldots, \alpha_p(x)\big)'$$

by minimizing

$$\min_{\boldsymbol{\alpha}} \sum_{t=1}^{T} \left[Y_t - \sum_{j=0}^{p} \alpha_j (X_t - x)^j \right]^2 K_h(x - X_t).$$

Let

$$\boldsymbol{Z}_t(x) = \big(1, (X_t - x), \ldots, (X_t - x)^p\big)'.$$

Then the criterion is

$$\min_{\boldsymbol{\alpha}} \sum_{t=1}^{T} (Y_t - \boldsymbol{\alpha}' \boldsymbol{Z}_t(x))^2 K_h(x - X_t).$$

- The fitted regression value at x is $\widehat{r}(x) = \widehat{\alpha}_0(x)$.
- The fitted derivatives are read directly from higher coefficients.

Matrix form and derivative estimation

Stack the local polynomial regressors into a design matrix $\boldsymbol{Z}_x$ and let $\boldsymbol{W}_x$ be the diagonal matrix of kernel weights. Then

$$\widehat{\boldsymbol{\alpha}}(x) = (\boldsymbol{Z}_x' \boldsymbol{W}_x \boldsymbol{Z}_x)^{-1} \boldsymbol{Z}_x' \boldsymbol{W}_x \boldsymbol{Y}.$$

If $\boldsymbol{e}_1 = (1, 0, \ldots, 0)'$, then

$$\widehat{r}(x) = \boldsymbol{e}_1' \widehat{\boldsymbol{\alpha}}(x).$$

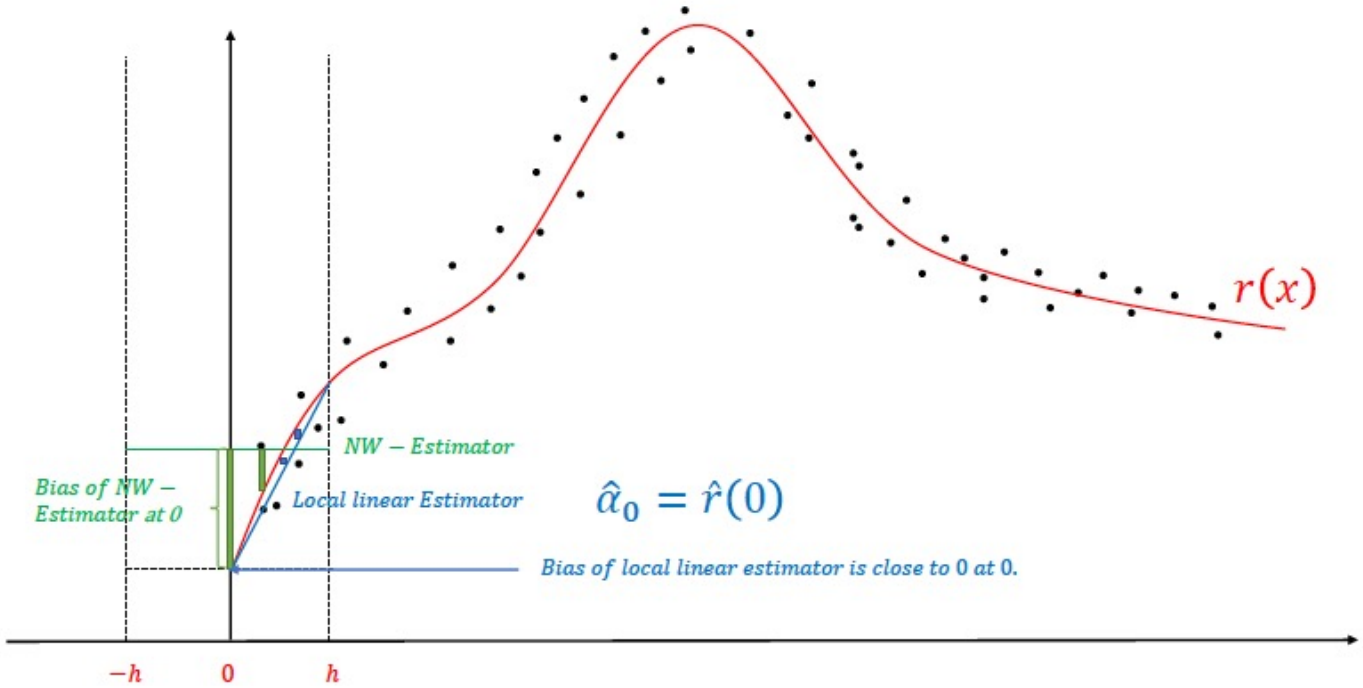
More generally,

$$\widehat{r}^{(\nu)}(x) = \nu! \, \widehat{\alpha}_\nu(x), \qquad \nu \leq p.$$

Practical meaning

One local weighted least-squares fit can estimate the function and several derivatives at once.

Local linear versus Nadaraya–Watson



- NW is the local constant fit.
- Local linear smoothing usually has smaller boundary bias.
- In practice, local linear regression is often the default nonparametric smoother for regression.

R block: density estimation and local regression in practice

```
# Univariate density estimation
x <- as.numeric(ret_xts)
plot(density(x))

# Multivariate KDE
library(ks)
X <- cbind(ftse_ret, dax_ret)
kde_fit <- kde(X)
plot(kde_fit, display = "filled.contour")

# Nadaraya-Watson / local polynomial smoothing
library(KernSmooth)
bw <- dpill(hp, mpg)
nw_fit <- locpoly(hp, mpg, bandwidth = bw, degree = 0)
ll_fit <- locpoly(hp, mpg, bandwidth = bw, degree = 1)
quad_fit <- locpoly(hp, mpg, bandwidth = bw, degree = 2)
```

The practical workhorse tasks are: choose a bandwidth, check sensitivity, compare fits, and resist the temptation to over-interpret small wiggles.

Summary and next-step interpretation

Part 1: multivariate volatility

We studied the conditional covariance matrix and saw four main strategies: EWMA, DVEC/BEKK, correlation-based decompositions, and Cholesky orthogonalization.

Part 2: nonparametric density estimation

We defined the kernel density estimator, derived its leading bias and variance terms, and saw why bandwidth choice dominates kernel choice.

Part 3: nonparametric regression

We introduced the Nadaraya–Watson estimator and then improved it through local polynomial smoothing, especially to reduce boundary bias.

Preview

The next lecture can build naturally on today's nonparametric tools with more applications, implementation, and links to robust inference.