STATS 242 / MSE 242: Algorithmic Trading & Quant Strategies (Summer 2011)
Course Reader + Microstructure & Order Book Supplements

Compiled study guide (with formula sheet, implementation notes, and exercises)
Generated 2026-01-09

Included sources

- STATS 242 SUMMER 2011.pdf (course notes)
- MicrostructureNotes.pdf (market microstructure teaching notes)
- stochastic model for orderbook dynamics.pdf (Cont-Stoikov-Talreja, 2010)

Contents

Chapter 1: How to use this reader

This document is a guided path through the STATS 242 / MSE 242 notes, with two supplements:

1) market microstructure teaching notes (what spreads, trades, and quotes mean), and

2) Cont-Stoikov-Talreja's stochastic limit order book (LOB) model (a tractable queueing model).

The goal is not to memorize formulas. The goal is to build a working mental model that survives contact with data and execution costs.

Two tracks: conceptual + implementation

Conceptual track: read each chapter's core ideas and do the sanity checks exercises.

Implementation track: each chapter ends with a mini-lab that maps ideas to a small piece of code or analysis you can actually run.

Key points

- Keep returns, prices, and efficient price distinct.
- Always separate signal quality from implementation (slippage, spread, impact, latency).
- Backtests are fragile; treat them as hypotheses, not trophies.

Mini-lab (implementation)

- Set up one notebook/script that loads data, computes returns, runs diagnostics, and logs every assumption (timezone, trading calendar, corporate actions).

Source pointers

- STATS 242 SUMMER 2011.pdf pp. 1-3
- MicrostructureNotes.pdf pp. 1-3

Chapter 2: Prices, returns, and stylized facts

Everything starts by choosing your primitive variable (price, log price, return, midprice, trade price) and living with the consequences.

This chapter fixes notation and summarizes the empirical shape of returns: fat tails, volatility clustering, and weak linear predictability.

Returns: simple vs log

Simple return: $R_t = P_t/P_{t-1} - 1$.

Log return: $r_t = \ln(1+R_t) = \ln P_t - \ln P_{t-1}$. Log returns add across time: $r_t^{(k)} = r_t + ... + r_{t-k+1}$.

Key points

- Use log returns when you care about time aggregation or modeling.
- Use simple returns when you care about exact PnL arithmetic, but log returns are often fine for small moves.

Formulas

```
R_t = P_t / P_{t-1} - 1
r_t = ln(1 + R_t) = p_t - p_{t-1}, where p_t = ln P_t
r_t^(k) = sum_{j=0}^{k-1} r_{t-j}
```

Mini-lab (implementation)

- Compute both simple and log returns; verify that summed log returns approximate compounded simple returns for small moves.

Normality is the wrong default

Empirically, returns have heavy tails: large moves occur much more often than a Normal model predicts.

Skewness and kurtosis summarize asymmetry and tail thickness; the Jarque-Bera (JB) test combines them into a simple normality diagnostic.

Key points

- A comfortable model hides tails; a useful model confronts them.
- With large samples, tiny deviations from normality will look significant; translate to economic risk.

Formulas

```
JB = n * (SK^2/6 + (K-3)^2/24) ~ chi^2_2 under normality
```

Mini-lab (implementation)

- Run the JB test on daily returns and intraday returns. Compare tails and volatility clustering (autocorrelation of squared returns).

Source pointers

- STATS 242 SUMMER 2011.pdf pp. 10-19

Chapter 3: Random walks, martingales, and what 'predictable' means

Markets can be unpredictable in price levels while still having exploitable structure in volatility, spreads, order flow, and cross-asset relationships.

You need the distinction between a random walk and a martingale to avoid over-reading weak serial correlations.

Random walk vs martingale

Random walk (RW): $X_t = X_{t-1} + eps_t$ (possibly with drift).

Martingale: $E[X_t \mid F_{t-1}] = X_{t-1}$. RW implies martingale under standard conditions, but the reverse is not generally true.

Returns can look uncorrelated while transforms (abs(r), r^2, durations) remain structured.

Key points

- Linear autocorrelation in returns is often small; autocorrelation in volatility proxies is often large.
- A statistically significant effect can still be economically useless after costs.

Formulas

```
Martingale: E[X_t | F_{t-1}] = X_{t-1}
RW: X_t = X_{t-1} + eps_t
```

Mini-lab (implementation)

- Compute ACF for returns and squared returns; quantify how much of a strategy's profit comes from volatility regimes rather than direction.

Autocorrelation tests

Omnibus tests (Box-Pierce / Ljung-Box) check multiple autocorrelations at once.

Use them as sanity checks: is 'nothing is going on' plausible for a given lag window?

Key points

- Test multiple lags before obsessing over a single lag.
- With large samples, tiny effects become significant; always translate to bps.

Mini-lab (implementation)

- Run Ljung-Box on returns and on signed order flow (if available). Compare which carries more structure.

Source pointers

- STATS 242 SUMMER 2011.pdf pp. 3-9
- STATS 242 SUMMER 2011.pdf pp. 35-40

Chapter 4: Asset pricing models: CAPM to APT to factor engineering

Even if you do not believe CAPM, the regression toolkit around it is everywhere: estimating betas, isolating idiosyncratic risk, and diagnosing exposures.

APT generalizes CAPM: multiple factors with premia constrained by no-arbitrage logic.

CAPM as a regression template

CAPM expresses excess return as alpha + beta * excess market return + residual.

Use it to measure systematic exposure and whether returns are a disguised market bet.

Key points

- Alpha that vanishes when you add plausible factors is not robust.
- Check residuals for heteroskedasticity and autocorrelation.

Formulas

```
R_t - R_{f,t} = alpha + beta * (M_t - R_{f,t}) + eps_t
```

Mini-lab (implementation)

- Fit CAPM to a strategy's returns; add a volatility factor and see if alpha collapses.

APT / linear factor models

General factor model: R = a + B f + e.

No-arbitrage implies expected returns lie in an affine space: mu = lambda0 * 1 + B * lambda.

Key points

- Be suspicious of factor discoveries that are sample noise.
- Stability across time matters more than in-sample fit.

Formulas

```
R = a + B f + e
mu = lambda0 * 1 + B * lambda
```

Mini-lab (implementation)

- Run PCA on a universe correlation matrix; interpret first components and compare to known factors.

Source pointers

- STATS 242 SUMMER 2011.pdf pp. 41-48

Chapter 5: Momentum, reversal, and technical analysis as statistics

Momentum and contrarian effects are replicated but fragile: horizon, universe, costs, and microstructure effects decide whether they survive.

The notes frame momentum as portfolio sorts (deciles) and discuss why it might exist.

J/K momentum strategies

Classic setup: use J months of past returns to form portfolios, then hold for K months.

Buy recent winners, sell recent losers; evaluate the spread.

Key points

- Monthly momentum does not imply intraday momentum.
- Turnover and shorting constraints can dominate results.

Mini-lab (implementation)

- Implement a decile momentum sort on monthly data; add transaction costs proportional to turnover and observe when the edge disappears.

Patterns as nonparametric estimation

Pattern recognition can be recast as estimating a smooth function m(x) in Y = m(X) + eps using kernel smoothing.

This reframes chart shapes as feature extraction plus classification under noise.

Key points

- Smoothers can introduce lookahead if windowing is wrong.
- Out-of-sample evaluation is mandatory; pattern mining otherwise overfits.

Formulas

```
Y_t = m(X_t) + eps_t
```

Mini-lab (implementation)

- Pick one pattern (e.g., MA crossover) and express it as features; compare a linear model vs a smoother using the same information set.

Source pointers

- STATS 242 SUMMER 2011.pdf pp. 57-62

Chapter 6: Mean reversion and cointegration: relative value you can justify

Pairs trading is where statistics becomes tradeable: define a spread, model it as mean reverting, and trade deviations.

Cointegration justifies why the spread can revert even if individual prices look like random walks.

Cointegration in one sentence

Two (or more) nonstationary series can have a stationary linear combination. That stationary combination is your spread.

Johansen's test is a standard multivariate test for cointegration rank.

Key points

- Cointegration is regime-sensitive; structural breaks can kill it.
- Model the spread, not the raw prices.

Mini-lab (implementation)

- Pick a pair; estimate hedge ratio; test residual stationarity; retest over rolling windows to detect breaks.

OU model and half-life intuition

The Ornstein-Uhlenbeck (OU) process is a continuous-time mean reversion model; its discrete analog is AR(1).

The mean reversion speed controls how quickly deviations decay; half-life is a convenient summary.

Key points

- A spread that mean reverts slower than your horizon is not a mean-reversion trade.
- OU parameters are easy to estimate, but unstable across regimes.

Formulas

```
AR(1): (X_t - mu) = phi * (X_{t-1} - mu) + eps_t
OU mean: E[X_t|X_0=x] = m + (x-m) * exp(-k t)
Half-life approx ln(2)/k
```

Mini-lab (implementation)

- Estimate OU parameters; compute half-life; build an s-score entry/exit rule; run a cost-aware backtest.

Source pointers

- STATS 242 SUMMER 2011.pdf pp. 35-40
- STATS 242 SUMMER 2011.pdf pp. 63-72

Chapter 7: Portfolio construction and rebalancing under constraints

Mean-variance optimization is simple in algebra and brutal in practice: covariance estimates are noisy and constraints dominate.

The notes discuss optimal weights, tracking error, and practical computation.

Mean-variance and tracking error

Portfolio return is w'R; risk is w'Sigma w. Benchmarking introduces tracking error variance TEV = (w-w_b)'Omega(w-w_b).

Realistic problems add constraints: long-only, leverage, turnover, bounds per asset, sector/factor exposure limits.

Key points

- Without turnover constraints, optimal portfolios can thrash.
- Shrinkage/robust covariance often helps more than fancy objectives.

Formulas

```
R_p = w'R
TEV = (w - w_b)'Omega(w - w_b)
```

Mini-lab (implementation)

- Build a basic MV portfolio; add a turnover penalty and observe stability improvements.

Source pointers

- STATS 242 SUMMER 2011.pdf pp. 49-56
- STATS 242 SUMMER 2011.pdf pp. 76-80

Chapter 8: High-frequency econometrics: durations, ACD, and price-orderflow dynamics

Intraday data behaves differently: microstructure noise, irregular timing, clustering, and execution mechanics.

Durations model time between events; ACD is the duration analogue of GARCH.

Durations and the ACD model

Let x_i be time between trades (or quote updates). ACD writes x_i = psi_i * e_i where psi_i is the conditional mean duration.

psi_i depends on lagged durations and lagged conditional means.

Key points

- Remove deterministic intraday seasonality before fitting ACD.
- Durations act like time deformation for irregular event clocks.

Formulas

```
x_i = psi_i * e_i
psi_i = omega + sum alpha_j x_{i-j} + sum beta_j psi_{i-j}
```

Mini-lab (implementation)

- Fit ACD(1,1) on durations after removing intraday seasonality; inspect residual dependence.

Price changes and signed volume as a VAR

Relate price changes Delta p_i and signed volume (or trade sign) w_i in a VAR to quantify lead-lag between order flow and prices.

Key points

- Order flow often predicts short-term moves more than past returns do.
- Bid-ask bounce can pollute Delta p; prefer midprice for some analyses.

Formulas

```
Delta p_i = sum a_j Delta p_{i-j} + sum b_j w_{i-j} + v_{1i}
w_i = sum c_j Delta p_{i-j} + sum d_j w_{i-j} + v_{2i}
```

Mini-lab (implementation)

- Estimate a small VAR; compute impulse responses; repeat using midprice vs trade price.

Source pointers

- STATS 242 SUMMER 2011.pdf pp. 27-34

Chapter 9: Market microstructure I: what the spread really is

Spreads are not just fees. They encode adverse selection (information), inventory risk, and mechanical costs.

Microstructure models separate efficient price from transaction price.

Quotes, trades, and efficient price

Transaction prices include a mechanical component (you pay the spread) and an informational component (quotes move when trades reveal information).

Modeling trick: transaction price = efficient price + noise; infer spread/information intensity from autocovariances.

Key points

- Backtesting on last-trade prices often backtests bid-ask bounce, not alpha.
- Always ask whether you are measuring midprice returns or trade-to-trade returns.

Mini-lab (implementation)

- With quote+trade data, compute returns on midprice vs trade price; compare autocorrelation and volatility.

Source pointers

- STATS 242 SUMMER 2011.pdf pp. 20-26
- MicrostructureNotes.pdf pp. 1-3

Chapter 10: The Roll model: estimating spread from price autocovariance

Roll (1984) gives a simple inference: under fixed spread and random trade direction, lag-1 autocovariance of trade-price changes is negative and reveals the effective spread.

It is a toy model, but a useful diagnostic baseline.

Model and estimator

Assume an efficient price random walk and transaction prices that flip between bid and ask by trade direction.

This generates negative serial dependence in transaction-price changes (bid-ask bounce).

Key points

- Roll breaks with time-varying spreads and informed trading.
- Still useful as a quick spread sanity check from trade prices alone.

Formulas

```
Effective spread idea: s approx 2 * sqrt( -Cov(Delta P_t, Delta P_{t-1}) )
```

Mini-lab (implementation)

- Estimate Roll spread on trade prices; compare to average quoted spread; track across volatility regimes.

Source pointers

- MicrostructureNotes.pdf Chapter 3-4 (Chapter 3 starts ~p. 20; Chapter 4 starts ~p. 23)

Chapter 11: Sequential trade and adverse selection (Glosten-Milgrom style)

Sequential trade models explain why quotes update after trades: trades carry information.

This is a conceptual explanation of permanent vs temporary impact.

Core idea

Liquidity providers set bid/ask quotes as conditional expectations of value given the observed trade (buy/sell), plus/minus costs.

If buys are more likely when value is high, then observing a buy shifts the conditional expectation upward.

Key points

- Information asymmetry widens spreads and creates adverse selection costs for liquidity providers.
- Naive mean reversion at very short horizons can be fading informed flow.

Formulas

```
Ask: A = E[V | Buy] + c
Bid: B = E[V | Sell] - c
```

Mini-lab (implementation)

- Compute price response to trade sign: E[Delta mid | Buy] vs E[Delta mid | Sell]; decompose bounce vs persistent movement.

Source pointers

- MicrostructureNotes.pdf Chapter 5 (starts ~p. 29) and Chapter 6 (starts ~p. 41)

Chapter 12: Limit order books as queueing systems (Cont-Stoikov-Talreja)

Model the LOB as a continuous-time Markov process tracking queue sizes at each price level.

With Poisson arrivals and cancellations, compute conditional probabilities like next midprice move or fill probability.

Model sketch

At each level i ticks from the best quote, limit orders arrive with intensity lambda_i, and cancellations occur with intensity proportional to queue size.

Market orders arrive with their own intensity. The system evolves as births (new limit orders) and deaths (market orders and cancellations).

Key points

- This model is nonstrategic: agents are not optimizing; they generate events stochastically.
- The payoff is tractability: Laplace-transform based computations for conditional probabilities.

Mini-lab (implementation)

- Calibrate lambda_i and cancellation rates from LOB data; compare an imbalance feature to the model's conditional up-move probability.

Routing and execution link

Routing depends on more than fees: fill probability and expected slippage matter when queues are thin and latency exists.

A queue model gives a principled estimate of those probabilities conditional on observed depth.

Key points

- Routing = explicit cost + fill probability + latency risk.
- Execution = signal quality + microstructure reality.

Mini-lab (implementation)

- Build a toy router: given two venues with fees and depth, route to minimize expected implementation shortfall.

Source pointers

- stochastic model for orderbook dynamics.pdf (15 pages)
- STATS 242 SUMMER 2011.pdf pp. 20-26

Chapter 13: Execution algorithms and market impact (Almgren-Chriss and friends)

Most alpha is tiny; execution costs are not. Execution is a control problem: choose a schedule balancing impact cost against price risk.

Splitting orders is simultaneously cost minimization, risk control, and information hiding.

Implementation shortfall and impact decomposition

Implementation shortfall: difference between decision price and average execution price plus explicit fees.

Permanent impact reflects lasting price movement; temporary impact reflects transient liquidity consumption.

Key points

- If you do not measure implementation shortfall, you do not know if you made money.
- Optimal schedules depend on volatility, liquidity, and risk aversion.

Mini-lab (implementation)

- Simulate a liquidation problem with a simple impact model and Brownian price risk; compare aggressive vs passive schedules.

Almgren-Chriss optimization (high level)

Divide the horizon into N intervals; choose shares to trade each interval.

Objective often minimizes E[cost] + lambda * Var(cost), producing a smooth schedule that becomes more front-loaded as risk aversion increases.

Key points

- The optimal solution is not always VWAP; it depends on parameters.
- Fit parameters from TCA, but monitor because conditions change.

Mini-lab (implementation)

- Implement a simple AC solver and plot schedules for multiple risk aversions; stress-test with a volatility spike.

Source pointers

- STATS 242 SUMMER 2011.pdf pp. 81-125

Chapter 14: Putting it together: a robust quant workflow

A strategy is a chain of assumptions. Your job is to make the chain thick enough that one weak link does not snap everything.

This chapter is a checklist for turning models into engineering.

Backtest like a skeptic

Separate research from production: different data, different failure modes.

Use walk-forward evaluation. Model costs explicitly. Track degradation from crowding, regime shifts, and volatility changes.

Key points

- Report edge in bps after costs, not raw returns.
- Turnover and capacity are first-class metrics.
- If you cannot explain the mechanism, assume luck until proven otherwise.

Mini-lab (implementation)

- Build a strategy report template: signal, data spec, assumptions, cost model, sensitivity, failure modes, monitoring metrics.

Source pointers

- STATS 242 SUMMER 2011.pdf pp. 81-125
- MicrostructureNotes.pdf pp. 1-3

Appendix A: Notation cheat sheet

Symbol	Meaning
P_t	Price at time t
p_t	Log price, $p_t = \ln(P_t)$
R_t	Simple return, $P_t/P_{t-1} - 1$
r_t	Log return, $\ln(1+R_t)$
Delta p_i	Intraday price change (choose mid or trade price consistently)
w_i	Signed volume or trade sign proxy
w	Portfolio weights (vector)
Sigma, Omega	Covariance matrices (returns; tracking error etc.)
A, B	Ask and bid quotes
V	Fundamental (efficient) value in microstructure models

Appendix B: Exercises and sanity checks

- Returns & tails: compute skewness, kurtosis, and JB; compare Normal vs Student-t fit.
- Microstructure noise: compare midprice returns vs trade-to-trade returns; quantify bid-ask bounce via lag-1 autocovariance.
- Roll spread: estimate effective spread and compare to quoted spread; track by volatility regime.
- Pairs trading: estimate hedge ratio; test residual stationarity; fit OU; compute half-life and an s-score; run a cost-aware backtest.
- Momentum: implement monthly J/K sorts; add turnover costs; test robustness across universes.
- ACD: fit ACD(1,1) to durations after removing intraday seasonality; inspect residuals.
- VAR: estimate Delta p / w VAR; compute impulse responses; repeat with midprice vs trade price.
- Execution: simulate Almgren-Chriss schedules under multiple risk aversions; evaluate implementation shortfall distribution.
- LOB model (if data available): estimate arrival/cancel intensities by level; compute conditional up-move probability and compare to depth imbalance heuristics.

Tip: treat every exercise as a hypothesis test. If a result disappears when you add costs, randomize labels, or change the regime, it probably was not robust.