

Quantitative Data Engineer Take-Home Test Report

I've developed a complete data processing and backtesting system that transforms raw tick data into actionable trading insights.

The solution handles market data processing, implements a mean-reversion trading strategy with robust risk controls, and includes stress testing capabilities - all in a modular, scalable framework.

System Architecture & Design

Modular Structure

The system is built with two main components that work together:

Market Data Processor - data preparation engine

- Loads and cleans raw tick data
- Transforms ticks into 1-minute OHLCV bars
- Calculates technical indicators (20-period MA and volatility)
- Stores processed data efficiently in Parquet format

Backtester - trading strategy laboratory

- Implements mean-reversion trading logic
- Manages positions and portfolio tracking
- Enforces risk constraints in real-time
- Provides performance analytics

How It Works Together

Raw Ticks → OHLCV Bars → Technical Indicators → Trading Signals → Backtest Execution → Performance Metrics → Stress Testing

Performance Considerations

The solution prioritizes efficiency through:

- **Smart Data Handling:** Uses pandas vectorized operations instead of slow loops
- **Symbol-Specific Processing:** Calculates indicators separately for each stock
- **Efficient Storage:** Parquet format reduces file size by 70-80% compared to CSV
- **Memory Management:** Processes data in chunks rather than loading everything at once

Risk Implementation

The system includes professional-grade risk controls:

Position Limits

- No more than 1,000 shares per symbol at any time
- Automatic position sizing based on available capital

Loss Protection

- 2% daily loss limit - if breached, all positions are automatically liquidated
- End-of-day flat requirement - no overnight positions held

Real-time Monitoring

- Continuous portfolio valuation during market hours
- Immediate execution of risk rules when triggers are hit

AI Usage Documentation

I used AI assistance primarily for:

Code Optimization

- Prompt: "Optimize this pandas rolling calculation for better performance"
- Used to improve the efficiency of volatility calculations

Debugging Assistance

- Prompt: "Why is this groupby operation returning unexpected results?"
- Helped identify and fix a data grouping issue

Architecture Validation

- Prompt: "Review this backtest class structure for potential edge cases"
- Confirmed the risk management approach was comprehensive

Next Steps: Scaling to Production

1. **Live Data Integration:** Connect to market data feeds (Bloomberg, Reuters)
2. **Alert System:** Add email/SMS notifications for risk breaches
3. **Web Dashboard:** Build a React frontend for strategy monitoring

Conclusion

This solution provides a robust foundation for quantitative trading research. It demonstrates professional-grade data processing, realistic backtesting capabilities, and meaningful risk management - all while maintaining the flexibility to expand into a production system.