Trend-Following Analysis at BFIX and WMR Fixing Windows

After reviewing the paper "Trend-following market behaviour at the 4pm London time BFIX and WMR fixing windows," I'll implement a Python solution to test the strategy described, integrating with Bloomberg for market data.

The paper investigates an interesting pattern where FX rates during the fixing windows tend to follow prior trends, creating an opportunity for traders to potentially improve execution by strategically choosing between BFIX and WMR fixing benchmarks based on the direction of the trend.

Implementation Strategy

I'll create a complete Python implementation that:

Connects to Bloomberg to retrieve BFIX and WMR fixing rates

Calculates trend signals based on the methodology in the paper

Simulates the trading strategy (buy at BFIX, sell at WMR for positive trends; sell at BFIX, buy at WMR for negative trends)

Analyzes and visualizes the results

Implementation Details

The code provides a comprehensive implementation of the strategy described in the paper. Here's a breakdown of the key components:

1. Data Collection

Connects to Bloomberg via the pdblp library to retrieve BFIX (3:30pm and 4pm London) and WMR (4pm London) fixing rates

Falls back to synthetic data generation if Bloomberg is unavailable

Focuses on major currency pairs as recommended in the paper (EURUSD, GBPUSD, USDJPY, EURJPY, GBPJPY, EURGBP)

2. Trend Signal Calculation

Implements the 10-day lookback window methodology described in the paper

Calculates the difference between the current 3:30pm BFIX and its 10-day moving average

Applies a trigger level (default 0.01%) to determine trend direction

3. Strategy Simulation

Buys at BFIX and sells at WMR when trend is positive

Sells at BFIX and buys at WMR when trend is negative

Calculates returns in basis points (10000x percentage change)

Computes cumulative returns and key performance metrics

4. Analysis

Splits data into pre- and post-WMR methodology change (February 16, 2015)

Tests parameter sensitivity (lookback window and trigger level)

Examines intraday price movements based on trend signals

Creates summary statistics for performance evaluation

5. Visualization

Plots cumulative returns over time with methodology change highlighted

Shows average returns by trend direction for each currency pair

Visualizes intraday price movements based on trend signals

Generates parameter sensitivity charts

Production Guidelines

To implement this strategy in a production environment:

Daily Monitoring: Monitor the 3:30pm London BFIX rates for your target currency pairs

Signal Generation: Calculate the 10-day trend signal as per the methodology:

Copy

Trend = Sign(ln(3:30pm BFIX) - ln(10-day average of 3:30pm BFIX))

Execution Decision:

For trend-following flow with WMR benchmark: Execute at BFIX instead

For counter-trend flow with BFIX benchmark: Execute at WMR instead

Currency Focus: Expect higher benefits in the most liquid currency pairs (EURUSD, GBPUSD, USDJPY)

Risk Management: Be aware of the volatility of this strategy, especially during market stress periods

Key Considerations

Transaction Costs: As noted in the paper, this strategy primarily benefits market participants who already trade at fixing benchmarks with no additional transaction costs for switching between BFIX and WMR.

Post-Methodology Change: The pattern became much stronger after February 2015 when WMR extended its calculation window.

Robustness: The strategy works best on the most actively traded currency pairs, particularly USD and JPY crosses.

Parameter Sensitivity: The paper suggests lookback windows of 5-10 days work best, which this implementation can validate using historical data.

To use this code in a live trading environment, you would need:

A Bloomberg terminal with the appropriate licenses for FX data

Scheduled execution of the signal generation code before 3:30pm London time daily

Integration with your order management system to route orders to the appropriate fixing mechanism

Would you like me to explain any specific part of the implementation in more detail?