**Simple Trading Strategy Prototype**

Build a basic backtesting engine to evaluate the performance of a simple price-based trading strategy. This task is representative of early-stage signal prototyping work, where we focus on hypothesis testing and fast iteration over polished production code. We recommend prioritizing the core requirements before implementing optional features.

**Dataset:**

You'll be working with 1-minute real BTC price data in the following format:

| **timestamp** | **mid\_price** | **volume** |
| --- | --- | --- |
| 1748001600 | 10000.00 | 1.25 |
| 1748001660 | 10010.00 | 0.95 |
| ... | ... | ... |

* timestamp: Unix timestamp
* mid\_price: Mid-price at the end of each minute
* volume: Traded base asset volume (in BTC). **You do not need to use this field unless you'd like to extend the strategy.**

**Task:**

1. **Compute 1 minute relative returns defined as:**  
   return\_t = (price\_t - price\_{t-1}) / price\_{t-1}
2. **Compute a 10-period simple moving average (SMA)** of the price.
3. **Define trading signals:**
   * **Buy signal**: when the price crosses **above** the SMA.
   * **Sell signal**: when the price crosses **below** the SMA.
4. **Backtest the strategy** under these conditions:
   * Start with $100,000 in capital.
   * Trade **only one position at a time**:
     + Buy = go fully long (100% of capital).
     + Sell = go fully short (borrow asset and short with full capital).
     + Switch sides only when the opposite signal appears.
   * No leverage or partial positions.
   * No fees (for now).
5. **Track and output these performance metrics:**
   * Total return (%)
   * Number of trades
   * Maximum drawdown
   * Sharpe ratio (Optional)
6. **Plot a cumulative PnL (equity) curve** over time.
7. (Optional): Add realistic transaction cost (e.g., 0.02% per trade) and re-evaluate result
8. (Optional): run multiple back-tests with different SMA periods and compare performance metrics.

We expect this to take ~**1-2 hours**. Please don’t over-optimize, we’re more interested in your thought process, code structure, and clarity than production-grade code.

**Submission:**

* Your code (.ipynb or .py + README.md)
* Any generated data files (if applicable)
* A few comments about your implementation
* Your interpretation of the results