An Overview of the System of Backtesting.

Made by Unbiased Trading (Unbiased Corporation).

1. Conceptualization Phase

1.1 Strategy Ideation

- Define the core concept of your trading strategy
    - Specify the exact market inefficiency or anomaly you're targeting
    - List the specific financial instruments you'll trade (e.g., large-cap US stocks, S&P 500 E-mini futures)
    - Determine the precise timeframe for your strategy (e.g., 5-minute bars for intraday, daily closes for swing trading, monthly data for long-term strategies)
- Outline the fundamental principles driving your strategy (e.g., momentum, mean reversion, volatility breakouts)

1.2 Hypothesis Formulation

- Develop a clear, testable hypothesis for your strategy
    - Example: "Stocks with positive earnings surprises and increasing rate of change will outperform the market over the next 3 months"
- List all assumptions underlying your hypothesis

1.3 Initial Research

- See who else has pursued this idea
    - Review academic papers related to your strategy concept
    - Analyze similar strategies in the market, including their strengths and weaknesses
- Identify and list potential data sources
    - For stock data: Consider vendors like Polygon, Alpaca, Yahoo finance

2. Data Preparation

2.1 Data Collection

- Gather comprehensive historical data
    - Ensure you have sufficient history (typically 5-10 years minimum, more for longer-term strategies)
- Verify data quality and accuracy
    - Cross-reference critical datapoints with multiple sources
    - Check for and document any known biases in your data source

2.2 Data Cleaning

- Remove or correct erroneous data points
    - Identify and handle outliers (A margin of 1 to 2% margin of error of a database is acceptable)
    - Correct for known errors (e.g., stock splits recorded on wrong dates)
- Handle corporate actions meticulously
    - Adjust historical prices for splits and dividends
    - Account for mergers, acquisitions.

2.3 Feature Engineering

- Create relevant technical indicators
    - Implement standard indicators if relevant to your strategy (e.g., moving averages, RSI, MACD)
    - Develop custom indicators based on your strategy's unique requirements
- Normalize or scale features appropriately
    - Consider techniques like z-score normalization or min-max scaling
    - Ensure scaling is done using only past data to avoid look-ahead bias

3. Strategy Development

3.1 Define Entry and Exit Rules

- Specify precise conditions for entering trades
    - Example: "Enter long when the 50-day SMA crosses above the 200-day SMA and RSI is below 70"
- Determine comprehensive exit criteria
    - Set specific take-profit levels (e.g., 2:1 reward-to-risk ratio)
    - Implement dynamic stop-loss rules (e.g., 2 ATR below entry for long positions)
    - Consider time-based exits (e.g., close all positions at market close for day trading strategies)
- Develop position sizing rules
    - Implement techniques like fixed fractional or volatility scaling position sizing

    - Adjust position sizes based on volatility or conviction level

3.2 Risk Management Integration

- Implement position sizing rules
    - Consider portfolio-level risk (e.g., ensure no single position exceeds 5% of portfolio value)
    - Incorporate correlation-based position sizing for multi-asset strategies
- Set up multi-layered risk limits
    - Define maximum drawdown thresholds (e.g., pause trading if 10% drawdown is reached)
    - Implement Value at Risk (VaR) limits, using methods like Historical VaR or Monte Carlo VaR
- Design dynamic portfolio allocation rules
    - Develop rules for rebalancing (e.g., rebalance when any position's weight deviates by >2% from target)
    - Implement risk parity or other advanced portfolio construction techniques if applicable

3.3 Parameter Selection

- Identify all key parameters in your strategy
    - List every numerical value that could be optimized (e.g., lookback periods, thresholds)
- Define reasonable ranges for these parameters
    - Use domain knowledge to set realistic bounds (e.g., 10-200 days for moving average periods)
    - Consider computational limitations when setting range granularity
- Plan for potential parameter adaptation
    - Consider how parameters might need to change in different market regimes
    - Develop rules for dynamic parameter adjustment if applicable

4. Initial Backtesting

4.1 Select Backtesting Framework

- Choose appropriate software or programming language
    - Consider factors like ease of use, performance, and documentation
    - Popular options: Python with libraries like ffn or quantstats or a dedicated platform like Ninjatrader 8 for future strategies
- Set up a robust backtesting environment
    - Ensure your system can handle the data volume and computational requirements
    - Implement proper data management and version control practices

4.2 Implement Strategy Logic

- Translate your strategy rules into precise logic
    - Ensure each rule is implemented exactly as intended, with no ambiguity
    - Double-check order of operations for complex conditions
- Verify accurate representation of all aspects
    - Test each component (entry rules, exit rules, position sizing) in isolation
    - Implement realistic order types (market, limit, stop) as would be used in live trading

4.3 Run Initial Backtest

- Execute the backtest on your historical dataset
    - Start with a basic run using initial parameter estimates
- Generate comprehensive preliminary performance metrics
    - Calculate returns (absolute and risk-adjusted)
    - Compute drawdowns, win rates, profit factors
    - Analyse trade durations and frequency

4.4 Basic Performance Analysis

- Calculate and interpret key metrics
    - Assess Sharpe ratio, Sortino ratio, and maximum drawdown
    - Compare performance to relevant benchmarks (e.g., S&P 500 for US equity strategies)
- Visualize results comprehensively
    - Plot equity curve with drawdowns highlighted
    - Create histograms of returns and trade outcomes

5. Strategy Refinement

5.1 Parameter Optimization

- Conduct thorough parameter optimization
    - Use techniques like random search or Bayesian optimization for large parameter spaces
- Implement robust cross-validation
    - Use time series cross-validation techniques (e.g., rolling origin)
    - Ensure each fold has sufficient data for reliable performance estimation
- Analyze parameter sensitivity rigorously
    - Create surface plots to visualize performance across parameter combinations

    - Identify regions of stability where small parameter changes don't drastically affect performance
    - Check for lucky parameter selection (huge spikes in performance) which demonstrate the strategy might not be robust

5.2 Transaction Cost Modeling

- Incorporate detailed, realistic transaction costs
    - Model variable commission structures (e.g., tiered pricing based on monthly volume)
    - Include exchange fees, clearing fees.
- Account for slippage comprehensively
    - Develop a slippage model based on historical order book data if available
    - Implement volume-dependent slippage for larger orders
    - Review the backtesting cheat sheet for more methods

6. Advanced Backtesting Techniques

6.1 Walk-Forward Analysis

- Implement sophisticated rolling window backtests
    - Use multiple window sizes to assess strategy stability
- Assess strategy performance across different time periods
    - Identify any secular trends in strategy performance
- Evaluate parameter stability over time

6.2 Monte Carlo Simulations

- Monte Carlo simulations:
    - Perform both reshuffle and resample Monte Carlo simulations
        - Reshuffle: Randomize the order of historical trades
        - Resample: Randomly select trades with replacement
    - Run at least 1000 simulations for reliable results
    - Ensure simulations capture a wide range of potential market behaviours
- Analyze distribution of outcomes comprehensively
    - Compute confidence intervals for key performance metrics
    - Assess the probability of extreme outcomes (e.g., probability of 20% drawdown)

6.3 Stress Testing

- Identify historical stress periods relevant to your strategy
    - Include major market crashes, periods of high volatility, and regime shifts

    - Consider asset-specific events (e.g., flash crashes, major geopolitical events)
- Create synthetic stress scenarios
    - Model extreme events not present in your historical data
    - Implement scenarios like sudden liquidity dry-ups or correlation breakdowns

7. Robustness Checks

7.1 Out-of-Sample Testing

- Reserve a significant portion of data for out-of-sample validation
    - Typically 20-30% of most recent data, or use walk-forward validation
- Compare performance metrics rigorously
    - Analyze any degradation in performance from in-sample to out-of-sample
- Adjust or move on if strategy shows significant discrepancies are found
    - Re-evaluate and potentially simplify strategy if out-of-sample performance is poor
    - Move on from the strategy, don't try and overfit it.

7.2 Sensitivity Analysis

- Vary key parameters to understand their impact on performance and if performance is simply lucky parameter choices
    - Use visualization techniques like 3D surface graphs or heat maps
    - Identify influential parameters for optimization
- Test strategy across different market regimes (bull, bear, sideways)
    - Identify regions of parameter stability where performance is consistent
- Be wary of strategies that only work with very specific parameter values

8. Bias Mitigation

8.1 Survivorship Bias

- Include delisted stocks or defunct assets in your universe
    - Obtain comprehensive point-in-time databases that include delisted securities
    - Implement proper handling of delistings in your backtesting engine
- Analyze impact of excluding vs including these assets
    - Quantify the performance difference when including/excluding defunct assets
    - Assess whether strategy inadvertently relies on selecting "survivors"

8.2 Look-Ahead Bias

- Ensure strategy only uses information available at the time of decision
    - Implement strict data alignment, especially for fundamental data
    - Be cautious with rebalancing assumptions, accounting for realistic information availability

Feeling overwhelmed by the complexity of proper backtesting? You're not alone.

While this guide provides a comprehensive overview, implementing a robust backtesting system takes significant time, expertise, and resources. If you're serious about elevating your trading strategy but don't want to spend months (or years) becoming a backtesting expert, we're here to help.

My team of seasoned traders and developers has implemented hundreds of backtests across various strategies and markets. We can help you navigate this complex process, avoid common pitfalls, and accelerate your path to a well-tested, data-driven trading strategy.

Why struggle alone when you can leverage our expertise?

Reach out to book a call with me by clicking here.

Let's discuss how we can help you rigorously test, refine and automate your trading ideas, saving you time and potentially costly mistakes along the way.

Otherwise, I hope you found some value in this overview.

Happy trading
Mounir