


-  Liquidation-Reentry Backtest Methodology
  - Technical Documentation for Leveraged ETF Strategy Analysis
    - Executive Summary
    - Mathematical Foundation
      - Core Leverage Mathematics
      - Margin Call Trigger Calculation
      - Daily Interest Accumulation
    - Liquidation-Reentry Process Flow
      - Phase 1: Initial Position Entry
      - Phase 2: Liquidation Event
      - Phase 3: Strategic Cooling Period
      - Phase 4: Position Re-Entry
    - Variables Description
      - Core Portfolio Variables
      - Risk Management Variables
      - Interest & Cost Variables
      - Dividend Variables
      - Position Tracking Variables
    - Chart Analysis Guide
      - Chart 1: Equity Evolution & Liquidation Events
      - Chart 2: Position Status Timeline
      - Chart 3: Drawdown Analysis
    - Performance Metrics
      - Core Performance Indicators
    - Results Interpretation
      - Success Indicators
      - Warning Signals
      - Critical Risk Levels
    - User Guide for New Users
      - Step 1: Run Your First Backtest
      - Step 2: Interpret Main Equity Chart
      - Step 3: Check Key Metrics
      - Step 4: Risk Assessment
    - Conclusion

# Liquidation-Reentry Backtest Methodology

## Technical Documentation for Leveraged ETF Strategy Analysis

### Executive Summary

The Liquidation-Reentry Backtest is a sophisticated simulation engine that models the realistic dynamics of leveraged ETF trading with margin requirements. Unlike traditional backtests that ignore margin calls, this system simulates forced liquidations, capital constraints, and strategic re-entry timing to provide accurate risk assessment and performance analysis.

#### Key Features:

- Realistic margin call simulation with immediate forced liquidation
- Strategic 2-day cooling period before re-entry (risk management choice)
- Compound capital erosion modeling through multiple liquidation cycles
- Comprehensive interest cost and dividend reinvestment tracking
- Institutional-grade risk analytics and performance attribution

### Mathematical Foundation

#### Core Leverage Mathematics

#### Position Sizing Formula:

$$\begin{aligned}\text{Total Position Size} &= \text{Initial Cash} \times \text{Leverage Multiplier} \\ \text{Margin Loan} &= \text{Total Position Size} - \text{Initial Cash} \\ \text{Shares Purchased} &= \text{Total Position Size} \div \text{Current ETF Price}\end{aligned}$$

#### Example with 6.5x Leverage:

Initial Cash: \$15,384,615  
Leverage: 6.5x  
Total Position:  $\$15,384,615 \times 6.5 = \$100,000,000$   
Margin Loan:  $\$100,000,000 - \$15,384,615 = \$84,615,385$   
Shares (at \$400):  $\$100,000,000 \div \$400 = 250,000$  shares

## Margin Call Trigger Calculation

### Maintenance Margin Requirement:

For Portfolio Margin: 15% of portfolio value  
For Reg-T Margin: 25% of portfolio value

Liquidation Trigger:  
 $\text{Current Equity} < (\text{Portfolio Value} \times \text{Maintenance Margin \%})$

Where:  
 $\text{Current Equity} = \text{Portfolio Value} - \text{Margin Loan}$   
 $\text{Portfolio Value} = \text{Shares Held} \times \text{Current ETF Price}$

### Critical Price Calculation:

$\text{Liquidation Price} = \text{Margin Loan} \div (\text{Shares} \times (1 - \text{Maintenance Margin \%}))$

Example with Portfolio Margin (15%):  
 $\text{Liquidation Price} = \$84,615,385 \div (250,000 \times 0.85) = \$398.19$   
 $\text{Price Drop to Liquidation} = (\$400.00 - \$398.19) \div \$400.00 = 0.45\%$

## Daily Interest Accumulation

### Interest Rate Structure:

Margin Rate = Direct from Excel column "FedFunds + 1.5%"  
(Historical rates already include Fed Funds + IBKR spreads)

$\text{Daily Interest Cost} = \text{Margin Loan} \times (\text{Annual Rate} \div 365)$

# Liquidation-Reentry Process Flow

## Phase 1: Initial Position Entry

## Capital Allocation:

```
def enter_position(current_equity, leverage, etf_price):  
    position_value = current_equity * leverage  
    shares_held = position_value / etf_price  
    margin_loan = position_value - current_equity  
    return shares_held, margin_loan, position_value
```

## Daily Monitoring:

- Daily equity calculation: Portfolio Value - Margin Loan
- Daily interest accrual: Margin Loan × Daily Rate
- Dividend reinvestment: Additional Shares = Dividend ÷ ETF Price
- Maintenance margin check: Equity ≥ Portfolio Value × 15%

## Phase 2: Liquidation Event

### Liquidation Trigger:

```
def check_margin_call(portfolio_value, margin_loan, maintenance_pct):  
    current_equity = portfolio_value - margin_loan  
    required_equity = portfolio_value * maintenance_pct  
    return current_equity < required_equity
```

### Liquidation Execution:

```
def execute_liquidation(shares_held, current_price, margin_loan):  
    gross_proceeds = shares_held * current_price  
    net_equity = gross_proceeds - margin_loan  
    return net_equity
```

## Phase 3: Strategic Cooling Period

### 2-Day Wait Implementation:

```
def cooling_period(liquidation_date):  
    wait_days = 2 # Strategic choice for risk management  
    earliest_reentry = liquidation_date + timedelta(days=wait_days)  
    return earliest_reentry
```

## Strategic Rationale:

- Volatility cooling: Allows market volatility to subside
- Emotional reset: Prevents impulsive immediate re-entry
- Technical analysis: Time to reassess market conditions
- Risk management: Avoids chasing momentum into further losses

## Phase 4: Position Re-Entry

### Re-Entry Conditions:

```
def can_reenter(current_equity, min_threshold=1000):  
    return current_equity >= min_threshold
```

# Variables Description

This section provides comprehensive documentation of all variables in the backtest output data, with mathematical formulas, example calculations, and corresponding code implementations.

## Core Portfolio Variables

### 1. Portfolio\_Value

$$\text{Portfolio\_Value} = \text{Shares\_Held} \times \text{Current\_ETF\_Price}$$

**Example:**  $882,378.88 \text{ shares} \times 113.33 = 100,000,000$

### Code Implementation:

```
# From historical_backtest.py line 201  
portfolio_value = shares_held * current_price
```

### 2. Margin\_Loan (Compounds Daily)

$$\text{New\_Margin\_Loan} = \text{Previous\_Margin\_Loan} + \text{Daily\_Interest\_Cost}$$

**Example:**  $50,002,205 + 2,206 = \$50,004,411$

### Code Implementation:

```
# From historical_backtest.py lines 194-196
daily_interest_cost = margin_loan * daily_interest_rate
margin_loan += daily_interest_cost
total_interest_paid += daily_interest_cost
```

## 3. Equity (Net Worth)

```
Equity = Portfolio_Value - Outstanding_Margin_Loan
```

**Example:**  $100,264,709 - 50,004,411 = \$50,260,298$

### Code Implementation:

```
# From historical_backtest.py line 202
current_equity_in_position = portfolio_value - margin_loan
```

## Risk Management Variables

## 4. Maintenance\_Margin\_Required

```
Maintenance_Required = Portfolio_Value × Maintenance_Margin_%
For Reg-T: Portfolio_Value × 25%
For Portfolio Margin: Portfolio_Value × 15%
```

**Example:**  $100,264,709 \times 25\% = 25,066,177$

### Code Implementation:

```
# From historical_backtest.py line 203
maintenance_margin_required = portfolio_value *
(margin_params['maintenance_margin_pct'] / 100.0)
```

## 5. Is\_Margin\_Call (Boolean Logic)

```
Margin_Call = TRUE if Equity < Maintenance_Required  
Margin_Call = FALSE if Equity ≥ Maintenance_Required
```

**Example:** 50,260,298 > 25,066,177 → NO margin call

### Code Implementation:

```
# From historical_backtest.py line 205  
is_margin_call = current_equity_in_position < maintenance_margin_required
```

## 6. Margin\_Call\_Price (Critical Price Level)

```
Margin_Call_Price = Margin_Loan ÷ (Shares × (1 - Maintenance_Margin_%))
```

**Example:** 50,004,411 ÷ (882,378.88 × 0.75) = 75.56

### Code Implementation:

```
# From historical_backtest.py line 260  
'Margin_Call_Price': margin_loan / (shares_held * (1 -  
margin_params['maintenance_margin_pct'] / 100.0)) if shares_held > 0 else 0
```

## Interest & Cost Variables

## 7. Daily\_Interest\_Cost

```
Daily_Rate = FedFunds + 1.5% ÷ 365  
Daily_Interest_Cost = Outstanding_Margin_Loan × Daily_Rate
```

**Example:** 50,002,205 × (5.277,215

### Code Implementation:

```
# From historical_backtest.py lines 187-189  
margin_rate = row['FedFunds + 1.5%'] / 100.0 # Convert percentage to decimal  
daily_interest_rate = margin_rate / 365  
daily_interest_cost = margin_loan * daily_interest_rate
```

## 8. Fed\_Funds\_Rate

Historical Fed\_Funds\_Rate (%)

**Example:** 3.77% (historical rate from 2001)

### Code Implementation:

```
# From historical_backtest.py line 186
fed_funds_rate = row['FedFunds (%)'] / 100.0
```

## 9. Margin\_Rate

Margin\_Rate = Direct from Excel "FedFunds + 1.5%" column  
(Pre-calculated rates including Fed Funds + IBKR spreads)

**Example:** 5.27% (Fed Funds 3.77% + IBKR spread 1.5%)

### Code Implementation:

```
# From historical_backtest.py lines 187-188
margin_rate = row['FedFunds + 1.5%'] / 100.0 # Convert percentage to decimal
daily_interest_rate = margin_rate / 365
```

## Dividend Variables

## 10. Dividend\_Payment

Dividend\_Received = Current\_Shares × Dividend\_Per\_Share  
Additional\_Shares = Dividend\_Received ÷ Current\_ETF\_Price

**Example:** Usually \$0.00 except quarterly dividend dates

### Code Implementation:



```
# From historical_backtest.py lines 198-203
if dividend_payment > 0:
    dividend_received = shares_held * dividend_payment
    total_dividends_received += dividend_received
    additional_shares = dividend_received / current_price
    shares_held += additional_shares
```

## Position Tracking Variables

### 11. Shares\_Held (Dynamic with Dividend Reinvestment)

```
New_Shares = Previous_Shares + Dividend_Reinvestment_Shares
```

**Example:** 882,378.88 shares (constant unless dividends paid)

#### Code Implementation:

```
# From historical_backtest.py line 66
shares = initial_investment / initial_price
# Plus dividend reinvestment as shown above
```

### 12. Current\_Equity (Real-Time Net Worth)

```
Current_Equity = Portfolio_Value - Margin_Loan
```

**Example:** \$50,260,298 (same as Equity variable)

#### Code Implementation:

```
# From historical_backtest.py line 249
current_equity = current_equity_in_position # Update equity
```

### 13. In\_Position (Boolean Status)

```
In_Position = TRUE when actively holding leveraged position
In_Position = FALSE during cooling periods or insufficient capital
```

## Code Implementation:

```
# From historical_backtest.py lines 160-165
if not in_position and current_equity >= min_equity_threshold:
    # Enter new position logic
    in_position = True
```

## 14. Wait\_Days\_Remaining (Cooling Period Counter)

```
Wait_Days_Remaining = Days left in mandatory 2-day cooling period
```

**Example:** 2, 1, 0 (countdown after liquidation)

## Code Implementation:

```
# From historical_backtest.py lines 142-143
if wait_days_remaining > 0:
    wait_days_remaining -= 1
```

## 15. Cycle\_Number (Position Sequence)

```
Cycle_Number = Sequential counter of position entry attempts
```

**Example:** 1 (first position), 2 (after first liquidation), etc.

## Code Implementation:

```
# From historical_backtest.py line 167
cycle_number += 1
```

## 16. Days\_In\_Position (Position Duration)

```
Days_In_Position = Counter of days current position has been held
```

**Example:** 1, 2, 3... (resets to 0 on new position)

## Code Implementation:

```
# From historical_backtest.py line 191
days_in_current_position += 1
```

## 17. Position\_Status (Categorical State)

```
Position_Status ∈ {
    'Active_Position',
    'Liquidated',
    'Position_Entered',
    'Waiting_After_Liquidation',
    'Insufficient_Equity'
}
```

### Code Implementation:

```
# From historical_backtest.py lines 225-245
daily_result['Position_Status'] = 'Liquidated' # On margin call
daily_result['Position_Status'] = 'Position_Entered' # On new entry
daily_result['Position_Status'] = 'Active_Position' # Normal trading
```

## 18. ETF\_Price (Market Data)

ETF\_Price = Current market price of selected ETF (SPY or VTI)

**Example:** 113.33, 113.63, \$113.71 (daily price changes)

### Code Implementation:

```
# From historical_backtest.py line 185
current_price = row[price_col]
```

# Chart Analysis Guide

## Chart 1: Equity Evolution & Liquidation Events

**Primary Blue Line - Equity Trajectory:**

$$\text{Daily\_Equity}(t) = \text{Shares}(t) \times \text{Price}(t) - \text{Margin\_Loan}(t)$$

## Visual Elements:

- Red Triangles (▼): Liquidation events showing exact loss amounts
- Green Triangles (▲): Re-entry events with reduced capital
- Line Discontinuities: Represent forced exits and strategic re-entries

## Chart 2: Position Status Timeline

### Color-Coded Status Indicators:

Green Dots: Active leveraged position (at risk)  
Red Dots: Liquidation day (loss realization)  
Orange Dots: Cooling period (strategic wait)  
Blue Dots: Re-entry day (new position establishment)  
Gray Dots: Insufficient capital (strategy termination)

## Chart 3: Drawdown Analysis

### Drawdown Calculation:

$\text{Running\_Max}(t) = \max(\text{Equity}(0) \text{ to } \text{Equity}(t))$   
 $\text{Drawdown}(t) = (\text{Equity}(t) - \text{Running\_Max}(t)) \div \text{Running\_Max}(t) \times 100$

### Recovery Mathematics:

To recover from 50% loss: need 100% gain  
To recover from 60% loss: need 150% gain  
To recover from 80% loss: need 400% gain

# Performance Metrics

## Core Performance Indicators

### Total Return:

Total Return = (Final Equity - Initial Cash) ÷ Initial Cash × 100  
CAGR = ((Final ÷ Initial)^(1/Years) - 1) × 100

## Risk-Adjusted Returns:

Sharpe Ratio = (Portfolio Return - Risk Free Rate) ÷ Portfolio Volatility  
Sortino Ratio = (Portfolio Return - Risk Free Rate) ÷ Downside Volatility

# Results Interpretation

## Success Indicators

- ✓ Total Return > 0%
- ✓ Liquidation Rate < 30%
- ✓ Average Survival > 180 days
- ✓ Sharpe Ratio > 0.5
- ✓ Max Drawdown < 50%

## Warning Signals

- ⚠ Liquidation Rate 30-70%
- ⚠ Average Survival 30-180 days
- ⚠ Sharpe Ratio 0-0.5
- ⚠ Max Drawdown 50-80%

## Critical Risk Levels

- 🔴 Liquidation Rate > 70%
- 🔴 Average Survival < 30 days
- 🔴 Sharpe Ratio < 0
- 🔴 Max Drawdown > 80%
- 🔴 Total Return < -50%

# User Guide for New Users

## Step 1: Run Your First Backtest

1. Select ETF (SPY or VTI)
2. Choose modest leverage (2-3x for beginners)
3. Set reasonable start date (2010 or later)
4. Review initial position summary before running

## Step 2: Interpret Main Equity Chart

- Blue line going up: Strategy is working
- Red triangles: You lost money and got liquidated
- Green triangles: You re-entered with less capital
- Flat periods: Cooling period (strategic wait)

## Step 3: Check Key Metrics

- Total Return: Did you make or lose money overall?
- Total Liquidations: How many times did you get forced out?
- Max Drawdown: What's the worst loss you experienced?
- Time in Market: What percentage of time were you invested?

## Step 4: Risk Assessment

- If liquidation rate > 50%, reduce leverage significantly
- If max drawdown > 60%, strategy may be too risky
- If negative total return, consider different approach

# Conclusion

This methodology provides a comprehensive framework for realistic leveraged trading simulation. By incorporating margin call mechanics, strategic cooling periods, and compound capital effects, the system delivers institutional-grade analysis for professional risk assessment and strategy development.