

# Package ‘strategyr’

September 12, 2026

**Type** Package

**Title** Execution-Oriented Strategy Workflows for Dynamic Trading Decisions

**Version** 0.1.7.1

**Description** Modular, execution-oriented strategy workflows that transform market features, portfolio state, and dynamic quantitative analysis into actionable signals, target positions, portfolio adjustments, and order intents. The package emphasizes path-dependent historical simulation so strategy decisions remain consistent with evolving account state, execution assumptions, and market conditions.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Language** en-US

**Depends** R (>= 4.1.0)

**Imports** data.table, stats, Rcpp

**LinkingTo** Rcpp

**Suggests** testthat, ggplot2, TTR, jsonlite

**URL** <https://github.com/OliverLDS/strategyr>

**BugReports** <https://github.com/OliverLDS/strategyr/issues>

**Collate** 'RcppExports.R' 'utils\_common.R' 'calc\_feat\_fixed\_income.R' 'calc\_feat\_fixed\_income\_conventions.R' 'calc\_feat\_options.R' 'calc\_feat\_stat.R' 'calc\_feat\_breadth.R' 'calc\_feat\_relative\_value.R' 'calc\_feat\_microstructure.R' 'calc\_feat\_futures.R' 'calc\_feat\_fx.R' 'calc\_feat\_options\_surface.R' 'calc\_feat\_credit.R' 'calc\_feat\_trend.R' 'calc\_feat\_momentum.R' 'calc\_feat\_volume.R' 'calc\_feat\_volatility.R' 'calc\_ema\_cross.R' 'calc\_ladder\_index.R' 'portfolio\_adjustment.R' 'portfolio\_backtest.R' 'portfolio\_fixed\_income.R' 'portfolio\_options.R' 'strategy\_mining.R' 'strategy\_public\_definition.R' 'strat\_utils.R'

'strat\_buy\_and\_hold.R' 'strat\_ladder\_bounce.R'  
 'strat\_ladder\_breakout.R' 'strat\_ema\_cross.R'  
 'strat\_ema\_cross\_adx.R' 'strat\_ema\_cross\_slope\_confirm.R'  
 'strat\_ema\_triple\_trend.R' 'strat\_bollinger\_revert.R'  
 'strat\_bollinger\_revert\_rsi.R'  
 'strat\_bollinger\_squeeze\_breakout.R'  
 'strat\_bollinger\_low\_adx\_revert.R' 'strat\_donchian\_breakout.R'  
 'strat\_donchian\_turtle.R' 'strat\_donchian\_retest\_breakout.R'  
 'strat\_rsi\_revert.R' 'strat\_rsi\_trend\_aware\_revert.R'  
 'strat\_rsi\_dynamic\_threshold\_revert.R' 'strat\_rsi\_divergence.R'  
 'strat\_macd\_cross.R' 'strat\_macd\_zero\_line.R'  
 'strat\_macd\_histogram\_momentum.R' 'strat\_atr\_breakout.R'  
 'strat\_atr\_breakout\_trailing\_stop.R' 'strat\_vol\_target.R'  
 'strat\_vol\_target\_regime\_floor.R' 'strat\_trend\_pullback.R'  
 'strat\_trend\_pullback\_atr.R' 'strat\_pair\_spread\_revert.R'  
 'strat\_pair\_spread\_bollinger.R'  
 'strat\_pair\_spread\_half\_life\_revert.R' 'strat\_ratio\_revert.R'  
 'strat\_relative\_strength.R'  
 'strat\_relative\_strength\_dual\_momentum.R'  
 'strat\_relative\_strength\_persistence.R' 'strat\_vwap\_revert.R'  
 'strat\_cross\_sectional\_rank\_allocator.R' 'strat\_fx\_carry.R'  
 'strat\_fx\_carry\_trend.R' 'strat\_fx\_carry\_basket\_rank.R'  
 'strat\_bond\_carry\_roll.R'  
 'strat\_bond\_carry\_roll\_spread\_filter.R'  
 'strat\_bond\_carry\_roll\_duration\_cap.R'  
 'strat\_credit\_spread\_momentum.R' 'strat\_credit\_spread\_revert.R'  
 'strat\_funding\_basis\_convergence.R' 'strat\_curve\_butterfly.R'  
 'strat\_curve\_steepener.R' 'strat\_curve\_steepener\_zscore.R'  
 'strat\_roll\_yield.R' 'strat\_roll\_yield\_momentum.R'  
 'strat\_roll\_yield\_cross\_sectional.R'  
 'strat\_roll\_yield\_mean\_revert.R' 'strat\_iv\_skew.R'  
 'strat\_iv\_skew\_zscore.R' 'strat\_iv\_skew\_realized\_vol\_confirm.R'  
 'strat\_iv\_directional\_overlay.R' 'strat\_iv\_term\_structure.R'  
 'strat\_vol\_carry.R' 'strat\_gamma\_scalp\_support.R'  
 'strat\_regime\_switch.R' 'strat\_straddle.R' 'strat\_strangle.R'  
 'strat\_vertical\_spread.R'

**Config/roxygen2/version** 7.2.3

**NeedsCompilation** yes

**Author** Oliver Zhou [aut, cre]

**Maintainer** Oliver Zhou <oliver.yxzhou@gmail.com>

**Repository** CRAN

**Date/Publication** 2026-09-12 13:30:02 UTC

## Contents

backtest\_portfolio\_weights . . . . . 9

|                                             |    |
|---------------------------------------------|----|
| backtest_rcpp . . . . .                     | 10 |
| build_order_intents . . . . .               | 11 |
| calc_ADX . . . . .                          | 11 |
| calc_aroon . . . . .                        | 12 |
| calc_ATR . . . . .                          | 12 |
| calc_ATR_quantile . . . . .                 | 13 |
| calc_backtest_performance . . . . .         | 13 |
| calc_bid_ask_spread . . . . .               | 14 |
| calc_BollingerBands . . . . .               | 15 |
| calc_bond_accrued_interest . . . . .        | 15 |
| calc_bond_carry . . . . .                   | 16 |
| calc_bond_carry_roll_decomp . . . . .       | 17 |
| calc_bond_cashflows . . . . .               | 17 |
| calc_bond_clean_price . . . . .             | 18 |
| calc_bond_convexity . . . . .               | 18 |
| calc_bond_current_yield . . . . .           | 19 |
| calc_bond_dirty_price . . . . .             | 20 |
| calc_bond_duration . . . . .                | 21 |
| calc_bond_dv01 . . . . .                    | 21 |
| calc_bond_econvexity . . . . .              | 22 |
| calc_bond_eduration . . . . .               | 23 |
| calc_bond_holding_period_return . . . . .   | 24 |
| calc_bond_key_rate_duration . . . . .       | 25 |
| calc_bond_key_rate_duration_tenor . . . . . | 26 |
| calc_bond_mduration . . . . .               | 27 |
| calc_bond_next_coupon . . . . .             | 27 |
| calc_bond_nominal_spread . . . . .          | 28 |
| calc_bond_npv . . . . .                     | 28 |
| calc_bond_prev_coupon . . . . .             | 29 |
| calc_bond_price_change_approx . . . . .     | 30 |
| calc_bond_pv01 . . . . .                    | 31 |
| calc_bond_rate_return_approx . . . . .      | 31 |
| calc_bond_risk_state . . . . .              | 32 |
| calc_bond_roll_down_return . . . . .        | 33 |
| calc_bond_spread_duration . . . . .         | 34 |
| calc_bond_spread_return_approx . . . . .    | 35 |
| calc_bond_total_return_decomp . . . . .     | 35 |
| calc_bond_yield . . . . .                   | 36 |
| calc_bond_zspread . . . . .                 | 37 |
| calc_bond_zspread_return_approx . . . . .   | 38 |
| calc_breadth_ad . . . . .                   | 38 |
| calc_breadth_adl . . . . .                  | 39 |
| calc_breadth_high_low . . . . .             | 39 |
| calc_breadth_ratio . . . . .                | 40 |
| calc_breadth_trin . . . . .                 | 41 |
| calc_CCI . . . . .                          | 41 |
| calc_chaikinAD . . . . .                    | 42 |
| calc_chaikinVolatility . . . . .            | 42 |

|                                          |    |
|------------------------------------------|----|
| calc_CMF . . . . .                       | 43 |
| calc_CMO . . . . .                       | 43 |
| calc_contango_backwardation . . . . .    | 44 |
| calc_coupon_schedule . . . . .           | 44 |
| calc_credit_spread . . . . .             | 45 |
| calc_curve_butterfly . . . . .           | 46 |
| calc_curve_shock . . . . .               | 46 |
| calc_curve_slope . . . . .               | 47 |
| calc_curve_zero_rate . . . . .           | 47 |
| calc_day_count_frac . . . . .            | 48 |
| calc_DEMA . . . . .                      | 48 |
| calc_DonchianChannels . . . . .          | 49 |
| calc_EMA . . . . .                       | 49 |
| calc_EMA_cross . . . . .                 | 50 |
| calc_EMV . . . . .                       | 51 |
| calc_excess_spread . . . . .             | 51 |
| calc_front_next_spread . . . . .         | 52 |
| calc_futures_curve_butterfly . . . . .   | 53 |
| calc_futures_curve_slope . . . . .       | 53 |
| calc_fx_basis . . . . .                  | 54 |
| calc_fx_carry . . . . .                  | 55 |
| calc_fx_forward_points . . . . .         | 55 |
| calc_fx_realized_carry . . . . .         | 56 |
| calc_HMA . . . . .                       | 57 |
| calc_KeltnerChannels . . . . .           | 57 |
| calc_KST . . . . .                       | 58 |
| calc_kurtosis . . . . .                  | 59 |
| calc_ladder_index . . . . .              | 59 |
| calc_log_spread . . . . .                | 60 |
| calc_MACD . . . . .                      | 61 |
| calc_MFI . . . . .                       | 61 |
| calc_microprice . . . . .                | 62 |
| calc_mid_price . . . . .                 | 62 |
| calc_normalize . . . . .                 | 63 |
| calc_OBV . . . . .                       | 63 |
| calc_option_delta . . . . .              | 64 |
| calc_option_forward_moneyness . . . . .  | 64 |
| calc_option_gamma . . . . .              | 65 |
| calc_option_greeks . . . . .             | 66 |
| calc_option_iv . . . . .                 | 67 |
| calc_option_iv_skew . . . . .            | 68 |
| calc_option_iv_term_structure . . . . .  | 69 |
| calc_option_moneyness . . . . .          | 69 |
| calc_option_put_call_iv_spread . . . . . | 70 |
| calc_option_rho . . . . .                | 71 |
| calc_option_risk_state . . . . .         | 71 |
| calc_option_smile_slope . . . . .        | 72 |
| calc_option_theta . . . . .              | 73 |

|                                             |     |
|---------------------------------------------|-----|
| calc_option_vega . . . . .                  | 74  |
| calc_order_imbalance . . . . .              | 74  |
| calc_percent_rank . . . . .                 | 75  |
| calc_period_rate . . . . .                  | 75  |
| calc_position_greeks . . . . .              | 76  |
| calc_price_impact_proxy . . . . .           | 76  |
| calc_ratio . . . . .                        | 77  |
| calc_realized_vol . . . . .                 | 78  |
| calc_relative_strength . . . . .            | 78  |
| calc_ROC . . . . .                          | 79  |
| calc_rolling_beta . . . . .                 | 79  |
| calc_rolling_corr . . . . .                 | 80  |
| calc_rolling_quantile . . . . .             | 81  |
| calc_roll_yield . . . . .                   | 81  |
| calc_RSI . . . . .                          | 82  |
| calc_SAR . . . . .                          | 83  |
| calc_skewness . . . . .                     | 83  |
| calc_slippage_proxy . . . . .               | 84  |
| calc_SMA . . . . .                          | 84  |
| calc_SMI . . . . .                          | 85  |
| calc_spread . . . . .                       | 85  |
| calc_spread_curve_butterfly . . . . .       | 86  |
| calc_spread_curve_slope . . . . .           | 86  |
| calc_StochasticOscillator . . . . .         | 87  |
| calc_strategy_performance_summary . . . . . | 88  |
| calc_term_structure_carry . . . . .         | 89  |
| calc_tracking_error . . . . .               | 90  |
| calc_TRIX . . . . .                         | 90  |
| calc_turnover . . . . .                     | 91  |
| calc_ultimateOscillator . . . . .           | 92  |
| calc_VWAP . . . . .                         | 92  |
| calc_VWMA . . . . .                         | 93  |
| calc_WMA . . . . .                          | 93  |
| calc_WPR . . . . .                          | 94  |
| calc_ZLEMA . . . . .                        | 94  |
| calc_zscore . . . . .                       | 95  |
| compare_strategy_backtests . . . . .        | 95  |
| filter_walk_forward_results . . . . .       | 96  |
| gen_action_plan_rcpp . . . . .              | 97  |
| mine_strategy_assets . . . . .              | 98  |
| mine_strategy_asset_years . . . . .         | 100 |
| mine_strategy_params . . . . .              | 102 |
| mine_strategy_walk_forward . . . . .        | 103 |
| plan_curve_trade_adjustment . . . . .       | 105 |
| plan_delta_neutral_adjustment . . . . .     | 105 |
| plan_duration_neutral_adjustment . . . . .  | 106 |
| plan_portfolio_adjustment . . . . .         | 106 |
| plan_vega_target_adjustment . . . . .       | 107 |

|                                                            |     |
|------------------------------------------------------------|-----|
| rank_strategy_results . . . . .                            | 108 |
| select_strategy_params . . . . .                           | 108 |
| strategy_monitor_definition . . . . .                      | 109 |
| strategy_public_definition . . . . .                       | 110 |
| strat_atr_breakout_action_plan . . . . .                   | 110 |
| strat_atr_breakout_tgt_pos . . . . .                       | 111 |
| strat_atr_breakout_trailing_stop_action_plan . . . . .     | 112 |
| strat_atr_breakout_trailing_stop_tgt_pos . . . . .         | 113 |
| strat_bollinger_low_adx_revert_action_plan . . . . .       | 114 |
| strat_bollinger_low_adx_revert_tgt_pos . . . . .           | 115 |
| strat_bollinger_revert_action_plan . . . . .               | 116 |
| strat_bollinger_revert_rsi_action_plan . . . . .           | 117 |
| strat_bollinger_revert_rsi_tgt_pos . . . . .               | 118 |
| strat_bollinger_revert_tgt_pos . . . . .                   | 119 |
| strat_bollinger_squeeze_breakout_action_plan . . . . .     | 120 |
| strat_bollinger_squeeze_breakout_tgt_pos . . . . .         | 121 |
| strat_bond_carry_roll_action_plan . . . . .                | 122 |
| strat_bond_carry_roll_duration_cap_action_plan . . . . .   | 123 |
| strat_bond_carry_roll_duration_cap_tgt_pos . . . . .       | 125 |
| strat_bond_carry_roll_spread_filter_action_plan . . . . .  | 126 |
| strat_bond_carry_roll_spread_filter_tgt_pos . . . . .      | 128 |
| strat_bond_carry_roll_tgt_pos . . . . .                    | 129 |
| strat_buy_and_hold_action_plan . . . . .                   | 131 |
| strat_buy_and_hold_tgt_pos . . . . .                       | 131 |
| strat_credit_spread_momentum_action_plan . . . . .         | 132 |
| strat_credit_spread_momentum_tgt_pos . . . . .             | 133 |
| strat_credit_spread_revert_action_plan . . . . .           | 134 |
| strat_credit_spread_revert_tgt_pos . . . . .               | 136 |
| strat_cross_sectional_rank_allocator_action_plan . . . . . | 137 |
| strat_cross_sectional_rank_allocator_tgt_pos . . . . .     | 138 |
| strat_curve_butterfly_action_plan . . . . .                | 139 |
| strat_curve_butterfly_tgt_pos . . . . .                    | 140 |
| strat_curve_steepener_action_plan . . . . .                | 141 |
| strat_curve_steepener_tgt_pos . . . . .                    | 142 |
| strat_curve_steepener_zscore_action_plan . . . . .         | 143 |
| strat_curve_steepener_zscore_tgt_pos . . . . .             | 144 |
| strat_donchian_breakout_action_plan . . . . .              | 145 |
| strat_donchian_breakout_tgt_pos . . . . .                  | 146 |
| strat_donchian_retest_breakout_action_plan . . . . .       | 147 |
| strat_donchian_retest_breakout_tgt_pos . . . . .           | 148 |
| strat_donchian_turtle_action_plan . . . . .                | 149 |
| strat_donchian_turtle_tgt_pos . . . . .                    | 150 |
| strat_ema_cross_action_plan . . . . .                      | 150 |
| strat_ema_cross_adx_action_plan . . . . .                  | 152 |
| strat_ema_cross_adx_tgt_pos . . . . .                      | 153 |
| strat_ema_cross_slope_confirm_action_plan . . . . .        | 154 |
| strat_ema_cross_slope_confirm_tgt_pos . . . . .            | 155 |
| strat_ema_cross_tgt_pos . . . . .                          | 156 |

|                                                             |     |
|-------------------------------------------------------------|-----|
| strat_ema_triple_trend_action_plan . . . . .                | 157 |
| strat_ema_triple_trend_tgt_pos . . . . .                    | 158 |
| strat_funding_basis_convergence_action_plan . . . . .       | 159 |
| strat_funding_basis_convergence_tgt_pos . . . . .           | 160 |
| strat_fx_carry_action_plan . . . . .                        | 162 |
| strat_fx_carry_basket_rank_action_plan . . . . .            | 163 |
| strat_fx_carry_basket_rank_tgt_pos . . . . .                | 164 |
| strat_fx_carry_tgt_pos . . . . .                            | 165 |
| strat_fx_carry_trend_action_plan . . . . .                  | 166 |
| strat_fx_carry_trend_tgt_pos . . . . .                      | 167 |
| strat_gamma_scalp_support_action_plan . . . . .             | 168 |
| strat_gamma_scalp_support_tgt_pos . . . . .                 | 170 |
| strat_iv_directional_overlay_action_plan . . . . .          | 171 |
| strat_iv_directional_overlay_tgt_pos . . . . .              | 172 |
| strat_iv_skew_action_plan . . . . .                         | 174 |
| strat_iv_skew_realized_vol_confirm_action_plan . . . . .    | 175 |
| strat_iv_skew_realized_vol_confirm_tgt_pos . . . . .        | 176 |
| strat_iv_skew_tgt_pos . . . . .                             | 178 |
| strat_iv_skew_zscore_action_plan . . . . .                  | 179 |
| strat_iv_skew_zscore_tgt_pos . . . . .                      | 180 |
| strat_iv_term_structure_action_plan . . . . .               | 181 |
| strat_iv_term_structure_tgt_pos . . . . .                   | 183 |
| strat_ladder_bounce_action_plan . . . . .                   | 184 |
| strat_ladder_bounce_tgt_pos . . . . .                       | 185 |
| strat_ladder_breakout_action_plan . . . . .                 | 186 |
| strat_ladder_breakout_tgt_pos . . . . .                     | 187 |
| strat_macd_contrarian_action_plan . . . . .                 | 188 |
| strat_macd_contrarian_tgt_pos . . . . .                     | 189 |
| strat_macd_cross_action_plan . . . . .                      | 190 |
| strat_macd_cross_tgt_pos . . . . .                          | 191 |
| strat_macd_histogram_momentum_action_plan . . . . .         | 192 |
| strat_macd_histogram_momentum_tgt_pos . . . . .             | 193 |
| strat_macd_zero_line_action_plan . . . . .                  | 194 |
| strat_macd_zero_line_tgt_pos . . . . .                      | 195 |
| strat_pair_spread_bollinger_action_plan . . . . .           | 196 |
| strat_pair_spread_bollinger_tgt_pos . . . . .               | 197 |
| strat_pair_spread_half_life_revert_action_plan . . . . .    | 198 |
| strat_pair_spread_half_life_revert_tgt_pos . . . . .        | 199 |
| strat_pair_spread_revert_action_plan . . . . .              | 200 |
| strat_pair_spread_revert_tgt_pos . . . . .                  | 202 |
| strat_ratio_revert_action_plan . . . . .                    | 203 |
| strat_ratio_revert_tgt_pos . . . . .                        | 204 |
| strat_regime_switch_action_plan . . . . .                   | 205 |
| strat_regime_switch_tgt_pos . . . . .                       | 206 |
| strat_relative_strength_action_plan . . . . .               | 208 |
| strat_relative_strength_dual_momentum_action_plan . . . . . | 209 |
| strat_relative_strength_dual_momentum_tgt_pos . . . . .     | 210 |
| strat_relative_strength_persistence_action_plan . . . . .   | 211 |

|                                                          |     |
|----------------------------------------------------------|-----|
| strat_relative_strength_persistence_tgt_pos . . . . .    | 213 |
| strat_relative_strength_tgt_pos . . . . .                | 214 |
| strat_roll_yield_action_plan . . . . .                   | 215 |
| strat_roll_yield_cross_sectional_action_plan . . . . .   | 216 |
| strat_roll_yield_cross_sectional_tgt_pos . . . . .       | 217 |
| strat_roll_yield_mean_revert_action_plan . . . . .       | 218 |
| strat_roll_yield_mean_revert_tgt_pos . . . . .           | 219 |
| strat_roll_yield_momentum_action_plan . . . . .          | 221 |
| strat_roll_yield_momentum_tgt_pos . . . . .              | 222 |
| strat_roll_yield_tgt_pos . . . . .                       | 224 |
| strat_rsi_divergence_action_plan . . . . .               | 225 |
| strat_rsi_divergence_tgt_pos . . . . .                   | 226 |
| strat_rsi_dynamic_threshold_revert_action_plan . . . . . | 227 |
| strat_rsi_dynamic_threshold_revert_tgt_pos . . . . .     | 228 |
| strat_rsi_logr_revert_action_plan . . . . .              | 229 |
| strat_rsi_logr_revert_tgt_pos . . . . .                  | 230 |
| strat_rsi_revert_action_plan . . . . .                   | 231 |
| strat_rsi_revert_tgt_pos . . . . .                       | 232 |
| strat_rsi_trend_aware_revert_action_plan . . . . .       | 233 |
| strat_rsi_trend_aware_revert_tgt_pos . . . . .           | 234 |
| strat_straddle_action_plan . . . . .                     | 235 |
| strat_straddle_tgt_pos . . . . .                         | 236 |
| strat_strangle_action_plan . . . . .                     | 237 |
| strat_strangle_tgt_pos . . . . .                         | 238 |
| strat_trend_pullback_action_plan . . . . .               | 239 |
| strat_trend_pullback_atr_action_plan . . . . .           | 241 |
| strat_trend_pullback_atr_tgt_pos . . . . .               | 242 |
| strat_trend_pullback_tgt_pos . . . . .                   | 243 |
| strat_vertical_spread_action_plan . . . . .              | 244 |
| strat_vertical_spread_tgt_pos . . . . .                  | 245 |
| strat_vol_carry_action_plan . . . . .                    | 247 |
| strat_vol_carry_tgt_pos . . . . .                        | 248 |
| strat_vol_target_action_plan . . . . .                   | 249 |
| strat_vol_target_regime_floor_action_plan . . . . .      | 250 |
| strat_vol_target_regime_floor_tgt_pos . . . . .          | 251 |
| strat_vol_target_tgt_pos . . . . .                       | 252 |
| strat_vwap_revert_action_plan . . . . .                  | 253 |
| strat_vwap_revert_tgt_pos . . . . .                      | 254 |
| summarize_walk_forward_results . . . . .                 | 255 |

---

backtest\_portfolio\_weights

*Backtest Portfolio Target Weights*


---

## Description

Evaluates a panel of portfolio target weights with open-price rebalancing and close-price mark-to-market. This is a minimal portfolio-level companion to the single-instrument backtest engine: strategy logic supplies target weights, while this helper applies path-dependent portfolio accounting.

## Usage

```
backtest_portfolio_weights(
  DT,
  date_col = "date",
  asset_col = "asset",
  open_col = "open",
  close_col = "close",
  target_weight_col = "target_weight",
  initial_equity = 1,
  fee_rt = 0,
  rebalance_tol_weight = 0,
  contract_size_col = "contract_size",
  allow_short = TRUE,
  keep_positions = FALSE
)
```

## Arguments

|                      |                                                                                           |
|----------------------|-------------------------------------------------------------------------------------------|
| DT                   | A long 'data.table' panel containing date, asset, open, close, and target-weight columns. |
| date_col             | Date or timestamp column name.                                                            |
| asset_col            | Asset identifier column name.                                                             |
| open_col             | Open/execution price column name.                                                         |
| close_col            | Close/mark price column name.                                                             |
| target_weight_col    | Target portfolio-weight column name.                                                      |
| initial_equity       | Numeric starting equity.                                                                  |
| fee_rt               | Proportional transaction fee applied to traded notional.                                  |
| rebalance_tol_weight | Absolute weight-difference tolerance below which rebalances are skipped.                  |
| contract_size_col    | Optional contract-size column name. Defaults to one when absent.                          |
| allow_short          | Logical; when 'FALSE', negative target weights error.                                     |
| keep_positions       | Logical; when 'TRUE', return per-date position records.                                   |

**Value**

A list with ‘equity’ and, when requested, ‘positions’ tables.

---

|               |                                      |
|---------------|--------------------------------------|
| backtest_rcpp | <i>Run a Path-Dependent Backtest</i> |
|---------------|--------------------------------------|

---

**Description**

Runs the native path-dependent backtest engine over OHLC data and a target position path.

**Usage**

```
backtest_rcpp(
  timestamp,
  open,
  high,
  low,
  close,
  tgt_pos,
  pos_strat,
  tol_pos,
  strat,
  asset,
  ctr_size = 1,
  ctr_step = 1,
  lev = 10,
  fee_rt = 0,
  fund_rt = 0,
  rec = FALSE
)
```

**Arguments**

|                        |                                         |
|------------------------|-----------------------------------------|
| timestamp              | Numeric timestamps.                     |
| open, high, low, close | Numeric OHLC vectors.                   |
| tgt_pos                | Numeric target-position path.           |
| pos_strat              | Integer strategy position-state vector. |
| tol_pos                | Numeric tolerance vector.               |
| strat                  | Integer strategy identifier.            |
| asset                  | Integer asset identifier.               |
| ctr_size               | Contract size.                          |
| ctr_step               | Contract increment step.                |
| lev                    | Leverage.                               |

|         |                                    |
|---------|------------------------------------|
| fee_rt  | Fee rate.                          |
| fund_rt | Funding rate.                      |
| rec     | Whether to return recorder output. |

**Value**

A backtest result from the native engine.

---

|                     |                                      |
|---------------------|--------------------------------------|
| build_order_intents | <i>Build Portfolio Order Intents</i> |
|---------------------|--------------------------------------|

---

**Description**

Converts rebalancing deltas into a minimal order-intent table suitable for downstream execution adapters.

**Usage**

```
build_order_intents(adjustment_plan, pricing_method = "market")
```

**Arguments**

|                 |                                                           |
|-----------------|-----------------------------------------------------------|
| adjustment_plan | A 'data.table' produced by [plan_portfolio_adjustment()]. |
| pricing_method  | Pricing method recorded on each order intent.             |

**Value**

A 'data.table' containing one row per non-zero rebalance intent.

---

|          |                                               |
|----------|-----------------------------------------------|
| calc_ADX | <i>Add Average Directional Index Features</i> |
|----------|-----------------------------------------------|

---

**Description**

Computes positive and negative directional indicators, directional movement index, and average directional index columns from candle highs, lows, and closes in place.

**Usage**

```
calc_ADX(DT, ns = c(14))
```

**Arguments**

|    |                                                       |
|----|-------------------------------------------------------|
| DT | A 'data.table' containing 'high', 'low', and 'close'. |
| ns | Integer vector of ADX windows.                        |

**Value**

The input ‘DT’, modified by reference and returned invisibly.

---

|            |                           |
|------------|---------------------------|
| calc_aroon | <i>Add Aroon Features</i> |
|------------|---------------------------|

---

**Description**

Computes Aroon up, down, and oscillator columns from candle highs and lows in place.

**Usage**

```
calc_aroon(DT, ns = c(20))
```

**Arguments**

|    |                                             |
|----|---------------------------------------------|
| DT | A ‘data.table’ containing ‘high’ and ‘low’. |
| ns | Integer vector of Aroon windows.            |

**Value**

The input ‘DT’, modified by reference and returned invisibly.

---

|          |                                        |
|----------|----------------------------------------|
| calc_ATR | <i>Add Average True Range Features</i> |
|----------|----------------------------------------|

---

**Description**

Computes Wilder-style ATR columns and log-range ATR columns on a candle ‘data.table’ in place.

**Usage**

```
calc_ATR(DT, ns = c(10, 14, 30), hs = c(12, 24))
```

**Arguments**

|    |                                                               |
|----|---------------------------------------------------------------|
| DT | A ‘data.table’ containing ‘high’, ‘low’, and ‘close’.         |
| ns | Integer vector of ATR window sizes for classic ATR columns.   |
| hs | Numeric vector of half-life values for log-range ATR columns. |

**Value**

The input ‘DT’, modified by reference and returned invisibly.

---

|                   |                                  |
|-------------------|----------------------------------|
| calc_ATR_quantile | <i>Add ATR Quantile Features</i> |
|-------------------|----------------------------------|

---

**Description**

Computes rolling quantiles for existing log-range ATR columns on a candle ‘data.table’ in place.

**Usage**

```
calc_ATR_quantile(
  DT,
  hs = c(12, 24),
  window = 300,
  thresholds = c(0.05, 0.1, 0.2, 0.3, 0.7, 0.8, 0.9, 0.95)
)
```

**Arguments**

|            |                                                                              |
|------------|------------------------------------------------------------------------------|
| DT         | A ‘data.table’ containing ‘atr_logr_*’ columns for each requested half-life. |
| hs         | Numeric vector of half-life values whose ATR columns should be summarized.   |
| window     | Integer rolling window size.                                                 |
| thresholds | Numeric vector of quantile probabilities.                                    |

**Value**

The input ‘DT’, modified by reference and returned invisibly.

---

|                           |                                     |
|---------------------------|-------------------------------------|
| calc_backtest_performance | <i>Backtest Performance Metrics</i> |
|---------------------------|-------------------------------------|

---

**Description**

Computes compact performance metrics from a path-dependent backtest equity curve. The primary score is Sortino ratio, using log returns and downside deviation below a minimum acceptable annual return.

**Usage**

```
calc_backtest_performance(
  equity,
  annualization = 252,
  risk_free_return = 0,
  min_acceptable_return = 0
)
```

**Arguments**

- equity                Numeric equity curve.
- annualization      Numeric periods per year.
- risk\_free\_return        Numeric annual risk-free return used for excess return.
- min\_acceptable\_return    Numeric annual return threshold used for downside deviation.

**Value**

A one-row ‘data.table’ with return, risk, drawdown, and Sortino metrics.

---

|                     |                                    |
|---------------------|------------------------------------|
| calc_bid_ask_spread | <i>Add Bid-Ask Spread Features</i> |
|---------------------|------------------------------------|

---

**Description**

Computes absolute and relative bid-ask spread columns in place.

**Usage**

```
calc_bid_ask_spread(DT, bid_col = "bid", ask_col = "ask")
```

**Arguments**

- DT                    A ‘data.table’ containing bid and ask columns.
- bid\_col              Bid-price column name.
- ask\_col              Ask-price column name.

**Value**

The input ‘DT’, modified by reference and returned invisibly.

---

|                     |                                    |
|---------------------|------------------------------------|
| calc_BollingerBands | <i>Add Bollinger Band Features</i> |
|---------------------|------------------------------------|

---

**Description**

Computes Bollinger band mid, standard deviation, upper, lower, and touch flag columns from the 'close' series of a candle 'data.table' in place.

**Usage**

```
calc_BollingerBands(DT, ns = c(20), ks = c(2))
```

**Arguments**

|    |                                             |
|----|---------------------------------------------|
| DT | A 'data.table' containing a 'close' column. |
| ns | Integer vector of rolling window sizes.     |
| ks | Numeric vector of band-width multipliers.   |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

|                            |                                      |
|----------------------------|--------------------------------------|
| calc_bond_accrued_interest | <i>Compute Bond Accrued Interest</i> |
|----------------------------|--------------------------------------|

---

**Description**

Computes accrued coupon interest for a plain-vanilla fixed-coupon bond using the fraction of the current coupon period that has elapsed.

**Usage**

```
calc_bond_accrued_interest(par = 1, c_rate, freq = 2, accrual_frac = 0)
```

**Arguments**

|              |                                                                             |
|--------------|-----------------------------------------------------------------------------|
| par          | Numeric face value.                                                         |
| c_rate       | Numeric annual coupon rate.                                                 |
| freq         | Integer coupon frequency per year.                                          |
| accrual_frac | Numeric fraction of the current coupon period already accrued, on '[0, 1)'. |

**Value**

Numeric scalar accrued interest.

---

|                 |                                  |
|-----------------|----------------------------------|
| calc_bond_carry | <i>Compute Bond Carry Return</i> |
|-----------------|----------------------------------|

---

### Description

Computes a simple carry return estimate as coupon accrual earned over the holding horizon minus financing cost, scaled by the opening dirty price.

### Usage

```
calc_bond_carry(
  par = 1,
  c_rate,
  maturity,
  freq = 2,
  ytm,
  accrual_frac = 0,
  holding_years = 1/freq,
  funding_rate = 0
)
```

### Arguments

|               |                                                                                              |
|---------------|----------------------------------------------------------------------------------------------|
| par           | Numeric face value.                                                                          |
| c_rate        | Numeric annual coupon rate.                                                                  |
| maturity      | Numeric scheduled maturity in years.                                                         |
| freq          | Integer coupon frequency per year.                                                           |
| ytm           | Numeric annualized yield-to-maturity.                                                        |
| accrual_frac  | Numeric fraction of the current coupon period already accrued, on '[0, 1)'.<br>Default is 0. |
| holding_years | Numeric holding horizon in years.<br>Default is 1/freq.                                      |
| funding_rate  | Numeric annualized financing rate applied to the opening dirty price.<br>Default is 0.       |

### Value

Numeric scalar carry return.

---

`calc_bond_carry_roll_decomp`*Decompose Bond Carry And Roll Return*

---

**Description**

Returns carry, roll-down, and their sum as a compact one-row 'data.table'.

**Usage**

```
calc_bond_carry_roll_decomp(carry, roll_down_return)
```

**Arguments**

|                  |                           |
|------------------|---------------------------|
| carry            | Numeric carry return.     |
| roll_down_return | Numeric roll-down return. |

**Value**

One-row 'data.table' with 'carry', 'roll\_down', and 'carry\_roll'.

---

`calc_bond_cashflows`     *Compute Bond Cash Flows*

---

**Description**

Generates level-coupon bond cash flows by period.

**Usage**

```
calc_bond_cashflows(par = 1, c_rate, maturity, freq = 2)
```

**Arguments**

|          |                                    |
|----------|------------------------------------|
| par      | Numeric face value.                |
| c_rate   | Numeric annual coupon rate.        |
| maturity | Numeric maturity in years.         |
| freq     | Integer coupon frequency per year. |

**Value**

Numeric vector of period cash flows excluding time-zero.

---

calc\_bond\_clean\_price    *Compute Bond Clean Price*

---

### Description

Computes the clean price of a plain-vanilla fixed-coupon bond from yield and bond terms.

### Usage

```
calc_bond_clean_price(
    par = 1,
    c_rate,
    maturity,
    freq = 2,
    ytm,
    accrual_frac = 0
)
```

### Arguments

|              |                                                                             |
|--------------|-----------------------------------------------------------------------------|
| par          | Numeric face value.                                                         |
| c_rate       | Numeric annual coupon rate.                                                 |
| maturity     | Numeric scheduled maturity in years.                                        |
| freq         | Integer coupon frequency per year.                                          |
| ytm          | Numeric annualized yield-to-maturity.                                       |
| accrual_frac | Numeric fraction of the current coupon period already accrued, on '[0, 1)'. |

### Value

Numeric scalar clean price.

---

calc\_bond\_convexity    *Compute Bond Convexity*

---

### Description

Computes bond convexity in year-squared units from explicit cash flows and rates, or directly from bond terms.

**Usage**

```
calc_bond_convexity(
  cashflows = NULL,
  rates = NULL,
  par = NULL,
  c_rate = NULL,
  maturity = NULL,
  freq = 2,
  ytm = NULL
)
```

**Arguments**

|           |                                                                 |
|-----------|-----------------------------------------------------------------|
| cashflows | Optional numeric vector of period cash flows.                   |
| rates     | Optional numeric scalar or vector of per-period discount rates. |
| par       | Optional numeric face value for term-based input.               |
| c_rate    | Optional annual coupon rate for term-based input.               |
| maturity  | Optional maturity in years for term-based input.                |
| freq      | Integer compounding frequency per year for term-based input.    |
| ytm       | Optional annualized yield-to-maturity for term-based input.     |

**Value**

Numeric scalar convexity.

---

calc\_bond\_current\_yield

*Compute Bond Current Yield*

---

**Description**

Computes current yield as annual coupon income divided by clean price.

**Usage**

```
calc_bond_current_yield(
  price,
  c_rate,
  par = 1,
  freq = 2,
  price_type = c("clean", "dirty"),
  accrual_frac = 0
)
```

**Arguments**

|              |                                                                              |
|--------------|------------------------------------------------------------------------------|
| price        | Numeric observed bond price.                                                 |
| c_rate       | Numeric annual coupon rate.                                                  |
| par          | Numeric face value.                                                          |
| freq         | Integer coupon frequency per year.                                           |
| price_type   | Character scalar, either "clean" or "dirty".                                 |
| accrual_frac | Numeric fraction of the current coupon period already accrued, on $[0, 1)$ . |

**Value**

Numeric scalar current yield.

---

calc\_bond\_dirty\_price *Compute Bond Dirty Price*

---

**Description**

Computes the dirty price of a plain-vanilla fixed-coupon bond from yield and bond terms. Settlement within the current coupon period is represented by 'accrual\_frac'.

**Usage**

```
calc_bond_dirty_price(
  par = 1,
  c_rate,
  maturity,
  freq = 2,
  ytm,
  accrual_frac = 0
)
```

**Arguments**

|              |                                                                              |
|--------------|------------------------------------------------------------------------------|
| par          | Numeric face value.                                                          |
| c_rate       | Numeric annual coupon rate.                                                  |
| maturity     | Numeric scheduled maturity in years.                                         |
| freq         | Integer coupon frequency per year.                                           |
| ytm          | Numeric annualized yield-to-maturity.                                        |
| accrual_frac | Numeric fraction of the current coupon period already accrued, on $[0, 1)$ . |

**Value**

Numeric scalar dirty price.

---

|                    |                                       |
|--------------------|---------------------------------------|
| calc_bond_duration | <i>Compute Bond Macaulay Duration</i> |
|--------------------|---------------------------------------|

---

**Description**

Computes Macaulay duration in years from explicit cash flows and rates, or directly from bond terms.

**Usage**

```
calc_bond_duration(
  cashflows = NULL,
  rates = NULL,
  par = NULL,
  c_rate = NULL,
  maturity = NULL,
  freq = 2,
  ytm = NULL
)
```

**Arguments**

|           |                                                                 |
|-----------|-----------------------------------------------------------------|
| cashflows | Optional numeric vector of period cash flows.                   |
| rates     | Optional numeric scalar or vector of per-period discount rates. |
| par       | Optional numeric face value for term-based input.               |
| c_rate    | Optional annual coupon rate for term-based input.               |
| maturity  | Optional maturity in years for term-based input.                |
| freq      | Integer compounding frequency per year for term-based input.    |
| ytm       | Optional annualized yield-to-maturity for term-based input.     |

**Value**

Numeric scalar duration in years.

---

|                |                          |
|----------------|--------------------------|
| calc_bond_dv01 | <i>Compute Bond DV01</i> |
|----------------|--------------------------|

---

**Description**

Computes DV01 as the average absolute price change for a one-basis-point annualized yield shock.

**Usage**

```
calc_bond_dv01(
  par = 1,
  c_rate,
  maturity,
  freq = 2,
  ytm,
  accrual_frac = 0,
  bases = 1
)
```

**Arguments**

|              |                                                                             |
|--------------|-----------------------------------------------------------------------------|
| par          | Numeric face value.                                                         |
| c_rate       | Numeric annual coupon rate.                                                 |
| maturity     | Numeric scheduled maturity in years.                                        |
| freq         | Integer coupon frequency per year.                                          |
| ytm          | Numeric annualized yield-to-maturity.                                       |
| accrual_frac | Numeric fraction of the current coupon period already accrued, on '[0, 1)'. |
| bases        | Numeric yield shock in basis points. Defaults to '1'.                       |

**Value**

Numeric scalar DV01.

---

calc\_bond\_econvexity    *Compute Effective Bond Convexity*

---

**Description**

Computes effective convexity by bumping discount rates up and down by a basis point shock.

**Usage**

```
calc_bond_econvexity(
  cashflows = NULL,
  rates = NULL,
  par = NULL,
  c_rate = NULL,
  maturity = NULL,
  freq = 2,
  ytm = NULL,
  bases = 100
)
```

**Arguments**

|           |                                                                 |
|-----------|-----------------------------------------------------------------|
| cashflows | Optional numeric vector of period cash flows.                   |
| rates     | Optional numeric scalar or vector of per-period discount rates. |
| par       | Optional numeric face value for term-based input.               |
| c_rate    | Optional annual coupon rate for term-based input.               |
| maturity  | Optional maturity in years for term-based input.                |
| freq      | Integer compounding frequency per year for term-based input.    |
| ym        | Optional annualized yield-to-maturity for term-based input.     |
| bases     | Numeric shock size in basis points.                             |

**Value**

Numeric scalar effective convexity.

---

|                     |                                        |
|---------------------|----------------------------------------|
| calc_bond_eduration | <i>Compute Effective Bond Duration</i> |
|---------------------|----------------------------------------|

---

**Description**

Computes effective duration by bumping discount rates up and down by a basis point shock.

**Usage**

```
calc_bond_eduration(
  cashflows = NULL,
  rates = NULL,
  par = NULL,
  c_rate = NULL,
  maturity = NULL,
  freq = 2,
  ytm = NULL,
  bases = 100
)
```

**Arguments**

|           |                                                                 |
|-----------|-----------------------------------------------------------------|
| cashflows | Optional numeric vector of period cash flows.                   |
| rates     | Optional numeric scalar or vector of per-period discount rates. |
| par       | Optional numeric face value for term-based input.               |
| c_rate    | Optional annual coupon rate for term-based input.               |
| maturity  | Optional maturity in years for term-based input.                |
| freq      | Integer compounding frequency per year for term-based input.    |
| ym        | Optional annualized yield-to-maturity for term-based input.     |
| bases     | Numeric shock size in basis points.                             |

**Value**

Numeric scalar effective duration.

---

calc\_bond\_holding\_period\_return

*Compute Bond Holding-Period Return*

---

**Description**

Computes holding-period return from beginning and ending bond prices plus coupon income received during the holding period.

**Usage**

```
calc_bond_holding_period_return(
  begin_price,
  end_price,
  coupon_income = 0,
  price_type = c("dirty", "clean"),
  par = 1,
  c_rate = 0,
  freq = 2,
  begin_accrual_frac = 0,
  end_accrual_frac = 0
)
```

**Arguments**

|                    |                                                                             |
|--------------------|-----------------------------------------------------------------------------|
| begin_price        | Numeric beginning bond price.                                               |
| end_price          | Numeric ending bond price.                                                  |
| coupon_income      | Numeric coupon cash received during the holding period.                     |
| price_type         | Character scalar, either "dirty" or "clean".                                |
| par                | Numeric face value.                                                         |
| c_rate             | Numeric annual coupon rate.                                                 |
| freq               | Integer coupon frequency per year.                                          |
| begin_accrual_frac | Numeric fraction of the opening coupon period already accrued, on '[0, 1)'. |
| end_accrual_frac   | Numeric fraction of the closing coupon period already accrued, on '[0, 1)'. |

**Value**

Numeric scalar holding-period return.

---

calc\_bond\_key\_rate\_duration

*Compute Bond Key Rate Duration*


---

## Description

Computes key-rate duration by bumping one point on an explicit per-period discount-rate curve at a time.

## Usage

```
calc_bond_key_rate_duration(
  cashflows = NULL,
  rates = NULL,
  par = NULL,
  c_rate = NULL,
  maturity = NULL,
  freq = 2,
  ytm = NULL,
  accrual_frac = 0,
  key_periods = NULL,
  bases = 1
)
```

## Arguments

|              |                                                                                                    |
|--------------|----------------------------------------------------------------------------------------------------|
| cashflows    | Optional numeric vector of period cash flows.                                                      |
| rates        | Optional numeric scalar or vector of per-period discount rates.                                    |
| par          | Optional numeric face value for term-based input.                                                  |
| c_rate       | Optional annual coupon rate for term-based input.                                                  |
| maturity     | Optional maturity in years for term-based input.                                                   |
| freq         | Integer compounding frequency per year for term-based input and annualized basis-point conversion. |
| ytm          | Optional annualized yield-to-maturity for term-based flat-curve input.                             |
| accrual_frac | Numeric fraction of the current coupon period already accrued, on '[0, 1)'.                        |
| key_periods  | Optional integer vector of period indices to shock. Defaults to all periods.                       |
| bases        | Numeric annualized key-rate shock size in basis points.                                            |

## Value

A 'data.table' with 'key\_period' and 'krd'.

---

 calc\_bond\_key\_rate\_duration\_tenor

*Compute Bond Key Rate Duration By Curve Tenor*


---

## Description

Computes key-rate duration by bumping one annualized zero-rate tenor on an interpolated curve while holding the rest of the curve fixed.

## Usage

```
calc_bond_key_rate_duration_tenor(
    par = 1,
    c_rate,
    maturity,
    freq = 2,
    tenor,
    zero_rate,
    key_tenor,
    shock_bp = 1,
    accrual_frac = 0,
    method = c("linear")
)
```

## Arguments

|              |                                                                             |
|--------------|-----------------------------------------------------------------------------|
| par          | Numeric face value.                                                         |
| c_rate       | Numeric annual coupon rate.                                                 |
| maturity     | Numeric scheduled maturity in years.                                        |
| freq         | Integer coupon frequency per year.                                          |
| tenor        | Numeric vector of curve tenors in years.                                    |
| zero_rate    | Numeric vector of annualized zero rates aligned with ‘tenor’.               |
| key_tenor    | Numeric scalar tenor to shock.                                              |
| shock_bp     | Numeric key-rate shock size in basis points.                                |
| accrual_frac | Numeric fraction of the current coupon period already accrued, on ‘[0, 1)’. |
| method       | Character interpolation method. Currently only “linear” is supported.       |

## Value

Numeric scalar tenor-based key-rate duration.

---

|                     |                                       |
|---------------------|---------------------------------------|
| calc_bond_mduration | <i>Compute Bond Modified Duration</i> |
|---------------------|---------------------------------------|

---

**Description**

Computes modified duration in years from explicit cash flows and rates, or directly from bond terms.

**Usage**

```
calc_bond_mduration(  
  cashflows = NULL,  
  rates = NULL,  
  par = NULL,  
  c_rate = NULL,  
  maturity = NULL,  
  freq = 2,  
  ytm = NULL  
)
```

**Arguments**

|           |                                                                 |
|-----------|-----------------------------------------------------------------|
| cashflows | Optional numeric vector of period cash flows.                   |
| rates     | Optional numeric scalar or vector of per-period discount rates. |
| par       | Optional numeric face value for term-based input.               |
| c_rate    | Optional annual coupon rate for term-based input.               |
| maturity  | Optional maturity in years for term-based input.                |
| freq      | Integer compounding frequency per year for term-based input.    |
| ytm       | Optional annualized yield-to-maturity for term-based input.     |

**Value**

Numeric scalar modified duration in years.

---

|                       |                                 |
|-----------------------|---------------------------------|
| calc_bond_next_coupon | <i>Compute Next Coupon Date</i> |
|-----------------------|---------------------------------|

---

**Description**

Returns the next regular coupon date strictly after settlement.

**Usage**

```
calc_bond_next_coupon(settle_date, maturity_date, freq = 2, issue_date = NULL)
```

**Arguments**

- settle\_date      Scalar settlement date.
- maturity\_date    Scalar maturity date.
- freq             Integer coupon frequency per year.
- issue\_date       Optional scalar issue date used to bound the schedule.

**Value**

A scalar 'Date', or 'NA' if no future coupon exists within the bounded schedule.

---

|                          |                                    |
|--------------------------|------------------------------------|
| calc_bond_nominal_spread | <i>Compute Bond Nominal Spread</i> |
|--------------------------|------------------------------------|

---

**Description**

Computes the simple nominal spread between a bond yield and a benchmark yield, in annualized decimal units.

**Usage**

calc\_bond\_nominal\_spread(ytm, benchmark\_ytm)

**Arguments**

- ytm              Numeric annualized yield-to-maturity.
- benchmark\_ytm   Numeric annualized benchmark yield.

**Value**

Numeric scalar nominal spread.

---

|               |                                       |
|---------------|---------------------------------------|
| calc_bond_npv | <i>Compute Bond Net Present Value</i> |
|---------------|---------------------------------------|

---

**Description**

Computes bond present value from explicit cash flows and per-period rates, or directly from bond terms.

**Usage**

```
calc_bond_npv(
  cashflows = NULL,
  rates = NULL,
  par = NULL,
  c_rate = NULL,
  maturity = NULL,
  freq = 2,
  ytm = NULL
)
```

**Arguments**

|           |                                                                 |
|-----------|-----------------------------------------------------------------|
| cashflows | Optional numeric vector of period cash flows.                   |
| rates     | Optional numeric scalar or vector of per-period discount rates. |
| par       | Optional numeric face value for term-based input.               |
| c_rate    | Optional annual coupon rate for term-based input.               |
| maturity  | Optional maturity in years for term-based input.                |
| freq      | Integer compounding frequency per year for term-based input.    |
| ytm       | Optional annualized yield-to-maturity for term-based input.     |

**Value**

Numeric scalar present value.

---

calc\_bond\_prev\_coupon *Compute Previous Coupon Date*

---

**Description**

Returns the most recent regular coupon date on or before settlement.

**Usage**

```
calc_bond_prev_coupon(settle_date, maturity_date, freq = 2, issue_date = NULL)
```

**Arguments**

|               |                                                        |
|---------------|--------------------------------------------------------|
| settle_date   | Scalar settlement date.                                |
| maturity_date | Scalar maturity date.                                  |
| freq          | Integer coupon frequency per year.                     |
| issue_date    | Optional scalar issue date used to bound the schedule. |

**Value**

A scalar 'Date', or 'NA' if no previous coupon exists within the bounded schedule.

---

calc\_bond\_price\_change\_approx

*Approximate Bond Price Change From Duration And Convexity*


---

### Description

Approximates the absolute bond price change for an annualized yield shock using modified duration and convexity.

### Usage

```
calc_bond_price_change_approx(
  ytm,
  yield_shock,
  par = 1,
  c_rate,
  maturity,
  freq = 2,
  accrual_frac = 0,
  price_type = c("dirty", "clean")
)
```

### Arguments

|              |                                                                                                                      |
|--------------|----------------------------------------------------------------------------------------------------------------------|
| ytm          | Numeric annualized yield-to-maturity.                                                                                |
| yield_shock  | Numeric annualized yield shock in decimal units.                                                                     |
| par          | Numeric face value.                                                                                                  |
| c_rate       | Numeric annual coupon rate.                                                                                          |
| maturity     | Numeric scheduled maturity in years.                                                                                 |
| freq         | Integer coupon frequency per year.                                                                                   |
| accrual_frac | Numeric fraction of the current coupon period already accrued, on '[0, 1)'.<br>'0' means no coupon has been accrued. |
| price_type   | Character scalar, either "dirty" or "clean".                                                                         |

### Value

Numeric scalar approximate absolute price change.

---

|                |                          |
|----------------|--------------------------|
| calc_bond_pv01 | <i>Compute Bond PV01</i> |
|----------------|--------------------------|

---

**Description**

Computes PV01 as the average absolute dirty-price change for a one-basis- point annualized yield shock.

**Usage**

```
calc_bond_pv01(
  par = 1,
  c_rate,
  maturity,
  freq = 2,
  ytm,
  accrual_frac = 0,
  bases = 1
)
```

**Arguments**

|              |                                                                                              |
|--------------|----------------------------------------------------------------------------------------------|
| par          | Numeric face value.                                                                          |
| c_rate       | Numeric annual coupon rate.                                                                  |
| maturity     | Numeric scheduled maturity in years.                                                         |
| freq         | Integer coupon frequency per year.                                                           |
| ytm          | Numeric annualized yield-to-maturity.                                                        |
| accrual_frac | Numeric fraction of the current coupon period already accrued, on '[0, 1)'. Defaults to '0'. |
| bases        | Numeric yield shock in basis points. Defaults to '1'.                                        |

**Value**

Numeric scalar PV01.

---

|                              |                                     |
|------------------------------|-------------------------------------|
| calc_bond_rate_return_approx | <i>Approximate Bond Rate Return</i> |
|------------------------------|-------------------------------------|

---

**Description**

Computes the duration-convexity approximation to bond return from an annualized yield shock.

**Usage**

```
calc_bond_rate_return_approx(duration, delta_y, convexity = 0)
```

**Arguments**

|           |                                                  |
|-----------|--------------------------------------------------|
| duration  | Numeric modified duration.                       |
| delta_y   | Numeric annualized yield shock in decimal units. |
| convexity | Numeric convexity. Defaults to '0'.              |

**Value**

Numeric scalar approximate rate return.

---

|                      |                                |
|----------------------|--------------------------------|
| calc_bond_risk_state | <i>Compute Bond Risk State</i> |
|----------------------|--------------------------------|

---

**Description**

Builds a compact fixed-income risk-state snapshot that can be used by portfolio-adjustment or hedge-planning logic.

**Usage**

```
calc_bond_risk_state(
  par = 1,
  c_rate,
  maturity,
  freq = 2,
  ytm,
  accrual_frac = 0,
  tenor = NULL,
  zero_rate = NULL,
  zspread = NULL
)
```

**Arguments**

|              |                                                                             |
|--------------|-----------------------------------------------------------------------------|
| par          | Numeric face value.                                                         |
| c_rate       | Numeric annual coupon rate.                                                 |
| maturity     | Numeric scheduled maturity in years.                                        |
| freq         | Integer coupon frequency per year.                                          |
| ytm          | Numeric annualized yield-to-maturity.                                       |
| accrual_frac | Numeric fraction of the current coupon period already accrued, on '[0, 1)'. |
| tenor        | Optional numeric vector of curve tenors in years.                           |
| zero_rate    | Optional numeric vector of annualized zero rates aligned with 'tenor'.      |
| zspread      | Optional annualized z-spread in decimal units.                              |

**Value**

A one-row 'data.table' containing bond valuation and risk measures.

---

`calc_bond_roll_down_return`*Compute Bond Roll-Down Return*

---

**Description**

Computes the price return from rolling a bond down the curve over a holding horizon while leaving the annualized yield assumption unchanged.

**Usage**

```
calc_bond_roll_down_return(  
  par = 1,  
  c_rate,  
  maturity,  
  freq = 2,  
  ytm,  
  accrual_frac = 0,  
  holding_years = 1/freq  
)
```

**Arguments**

|               |                                                                             |
|---------------|-----------------------------------------------------------------------------|
| par           | Numeric face value.                                                         |
| c_rate        | Numeric annual coupon rate.                                                 |
| maturity      | Numeric scheduled maturity in years.                                        |
| freq          | Integer coupon frequency per year.                                          |
| ytm           | Numeric annualized yield-to-maturity.                                       |
| accrual_frac  | Numeric fraction of the current coupon period already accrued, on '[0, 1)'. |
| holding_years | Numeric holding horizon in years.                                           |

**Value**

Numeric scalar roll-down return.

---

 calc\_bond\_spread\_duration

*Compute Bond Spread Duration*


---

### Description

Computes spread duration by bumping an annualized z-spread up and down while holding the zero-rate curve fixed.

### Usage

```
calc_bond_spread_duration(
  par = 1,
  c_rate,
  maturity,
  freq = 2,
  tenor,
  zero_rate,
  zspread,
  shock_bp = 1,
  accrual_frac = 0,
  method = c("linear")
)
```

### Arguments

|              |                                                                             |
|--------------|-----------------------------------------------------------------------------|
| par          | Numeric face value.                                                         |
| c_rate       | Numeric annual coupon rate.                                                 |
| maturity     | Numeric scheduled maturity in years.                                        |
| freq         | Integer coupon frequency per year.                                          |
| tenor        | Numeric vector of curve tenors in years.                                    |
| zero_rate    | Numeric vector of annualized zero rates aligned with ‘tenor’.               |
| zspread      | Numeric annualized z-spread in decimal units.                               |
| shock_bp     | Numeric spread shock size in basis points.                                  |
| accrual_frac | Numeric fraction of the current coupon period already accrued, on ‘[0, 1)’. |
| method       | Character interpolation method. Currently only “linear” is supported.       |

### Value

Numeric scalar spread duration.

---

```
calc_bond_spread_return_approx
```

*Approximate Bond Spread Return*

---

**Description**

Computes the first-order approximation to bond return from a spread move.

**Usage**

```
calc_bond_spread_return_approx(spread_duration, delta_s)
```

**Arguments**

spread\_duration      Numeric spread duration.

delta\_s              Numeric annualized spread shock in decimal units.

**Value**

Numeric scalar approximate spread return.

---

```
calc_bond_total_return_decomp
```

*Decompose Bond Total Return*

---

**Description**

Returns the main building blocks of bond total return as a compact one-row ‘data.table’.

**Usage**

```
calc_bond_total_return_decomp(  
  carry,  
  roll_down_return = 0,  
  rate_return = 0,  
  spread_return = 0,  
  residual = 0  
)
```

Arguments

|                  |                                           |
|------------------|-------------------------------------------|
| carry            | Numeric carry return.                     |
| roll_down_return | Numeric roll-down return.                 |
| rate_return      | Numeric rate-driven return.               |
| spread_return    | Numeric spread-driven return.             |
| residual         | Numeric residual return. Defaults to '0'. |

Value

One-row 'data.table' with component returns and 'total\_return'.

---

|                 |                                       |
|-----------------|---------------------------------------|
| calc_bond_yield | <i>Compute Bond Yield To Maturity</i> |
|-----------------|---------------------------------------|

---

Description

Solves annualized yield-to-maturity from a clean or dirty bond price under a plain-vanilla fixed-coupon schedule.

Usage

```
calc_bond_yield(  
  price,  
  par = 1,  
  c_rate,  
  maturity,  
  freq = 2,  
  accrual_frac = 0,  
  price_type = c("dirty", "clean"),  
  interval = NULL  
)
```

Arguments

|              |                                                                                       |
|--------------|---------------------------------------------------------------------------------------|
| price        | Numeric observed bond price.                                                          |
| par          | Numeric face value.                                                                   |
| c_rate       | Numeric annual coupon rate.                                                           |
| maturity     | Numeric scheduled maturity in years.                                                  |
| freq         | Integer coupon frequency per year.                                                    |
| accrual_frac | Numeric fraction of the current coupon period already accrued, on '[0, 1)'.           |
| price_type   | Character scalar, either "dirty" or "clean".                                          |
| interval     | Optional numeric vector of length two giving the search bracket for annualized yield. |

**Value**

Numeric scalar annualized yield-to-maturity.

---

|                   |                              |
|-------------------|------------------------------|
| calc_bond_zspread | <i>Compute Bond Z-Spread</i> |
|-------------------|------------------------------|

---

**Description**

Solves the constant annualized spread that must be added to a per-period spot rate curve so discounted cash flows match an observed clean or dirty price.

**Usage**

```
calc_bond_zspread(
  price,
  spot_rates,
  cashflows = NULL,
  par = NULL,
  c_rate = NULL,
  maturity = NULL,
  freq = 2,
  accrual_frac = 0,
  price_type = c("dirty", "clean"),
  interval = NULL
)
```

**Arguments**

|              |                                                                                      |
|--------------|--------------------------------------------------------------------------------------|
| price        | Numeric observed bond price.                                                         |
| spot_rates   | Numeric scalar or vector of per-period spot rates.                                   |
| cashflows    | Optional numeric vector of period cash flows.                                        |
| par          | Optional numeric face value for term-based input.                                    |
| c_rate       | Optional annual coupon rate for term-based input.                                    |
| maturity     | Optional maturity in years for term-based input.                                     |
| freq         | Integer compounding frequency per year.                                              |
| accrual_frac | Numeric fraction of the current coupon period already accrued, on '[0, 1)'.<br>      |
| price_type   | Character scalar, either "dirty" or "clean".                                         |
| interval     | Optional numeric vector of length two giving the annualized z-spread search bracket. |

**Value**

Numeric scalar annualized z-spread.

---

|                                 |                                         |
|---------------------------------|-----------------------------------------|
| calc_bond_zspread_return_approx | <i>Approximate Bond Z-Spread Return</i> |
|---------------------------------|-----------------------------------------|

---

**Description**

Computes the first-order approximation to bond return from a z-spread move.

**Usage**

```
calc_bond_zspread_return_approx(spread_duration, delta_z)
```

**Arguments**

|                 |                                                     |
|-----------------|-----------------------------------------------------|
| spread_duration | Numeric spread duration.                            |
| delta_z         | Numeric annualized z-spread shock in decimal units. |

**Value**

Numeric scalar approximate z-spread return.

---

|                 |                                        |
|-----------------|----------------------------------------|
| calc_breadth_ad | <i>Compute Advance-Decline Breadth</i> |
|-----------------|----------------------------------------|

---

**Description**

Aggregates per-date advancing and declining counts from a cross-sectional asset panel.

**Usage**

```
calc_breadth_ad(
  DT,
  date_col = "date",
  asset_col = "asset",
  close_col = "close"
)
```

**Arguments**

|           |                                                                 |
|-----------|-----------------------------------------------------------------|
| DT        | A 'data.table' panel containing date, asset, and close columns. |
| date_col  | Date or timestamp column name.                                  |
| asset_col | Asset identifier column name.                                   |
| close_col | Close-price column name.                                        |

**Value**

A 'data.table' with per-date breadth counts and advance-decline difference.

---

|                  |                                    |
|------------------|------------------------------------|
| calc_breadth_adl | <i>Compute Advance-Dcline Line</i> |
|------------------|------------------------------------|

---

**Description**

Builds a cumulative advance-decline line from per-date breadth counts.

**Usage**

```
calc_breadth_adl(  
  DT,  
  date_col = "date",  
  asset_col = "asset",  
  close_col = "close"  
)
```

**Arguments**

|           |                                                                 |
|-----------|-----------------------------------------------------------------|
| DT        | A 'data.table' panel containing date, asset, and close columns. |
| date_col  | Date or timestamp column name.                                  |
| asset_col | Asset identifier column name.                                   |
| close_col | Close-price column name.                                        |

**Value**

A 'data.table' with per-date breadth counts, difference, and cumulative advance-decline line.

---

|                       |                                         |
|-----------------------|-----------------------------------------|
| calc_breadth_high_low | <i>Compute New-High New-Low Breadth</i> |
|-----------------------|-----------------------------------------|

---

**Description**

Aggregates per-date new-high and new-low counts relative to the prior 'n' observations for each asset.

**Usage**

```
calc_breadth_high_low(
  DT,
  date_col = "date",
  asset_col = "asset",
  high_col = "high",
  low_col = "low",
  n = 252
)
```

**Arguments**

|           |                                                                     |
|-----------|---------------------------------------------------------------------|
| DT        | A 'data.table' panel containing date, asset, high, and low columns. |
| date_col  | Date or timestamp column name.                                      |
| asset_col | Asset identifier column name.                                       |
| high_col  | High-price column name.                                             |
| low_col   | Low-price column name.                                              |
| n         | Integer lookback window for prior highs and lows.                   |

**Value**

A 'data.table' with per-date new-high/new-low counts and ratios.

---

|                    |                                     |
|--------------------|-------------------------------------|
| calc_breadth_ratio | <i>Compute Advance-Dcline Ratio</i> |
|--------------------|-------------------------------------|

---

**Description**

Computes the per-date advancing-to-declining ratio from a cross-sectional asset panel.

**Usage**

```
calc_breadth_ratio(
  DT,
  date_col = "date",
  asset_col = "asset",
  close_col = "close"
)
```

**Arguments**

|           |                                                                 |
|-----------|-----------------------------------------------------------------|
| DT        | A 'data.table' panel containing date, asset, and close columns. |
| date_col  | Date or timestamp column name.                                  |
| asset_col | Asset identifier column name.                                   |
| close_col | Close-price column name.                                        |

**Value**

A 'data.table' with per-date breadth counts and ratio.

---

|                   |                             |
|-------------------|-----------------------------|
| calc_breadth_trin | <i>Compute TRIN Breadth</i> |
|-------------------|-----------------------------|

---

**Description**

Computes per-date TRIN (Arms Index) values from advancing and declining counts and their associated traded volume.

**Usage**

```
calc_breadth_trin(  
  DT,  
  date_col = "date",  
  asset_col = "asset",  
  close_col = "close",  
  volume_col = "volume"  
)
```

**Arguments**

|            |                                                                         |
|------------|-------------------------------------------------------------------------|
| DT         | A 'data.table' panel containing date, asset, close, and volume columns. |
| date_col   | Date or timestamp column name.                                          |
| asset_col  | Asset identifier column name.                                           |
| close_col  | Close-price column name.                                                |
| volume_col | Traded-volume column name.                                              |

**Value**

A 'data.table' with per-date advance/decline counts, volume totals, and TRIN.

---

|          |                         |
|----------|-------------------------|
| calc_CCI | <i>Add CCI Features</i> |
|----------|-------------------------|

---

**Description**

Computes Commodity Channel Index columns from candle highs, lows, and closes in place using typical price, its rolling mean, and rolling mean absolute deviation.

**Usage**

```
calc_CCI(DT, ns = c(20), constant = 0.015)
```

**Arguments**

|          |                                                       |
|----------|-------------------------------------------------------|
| DT       | A 'data.table' containing 'high', 'low', and 'close'. |
| ns       | Integer vector of rolling windows.                    |
| constant | Numeric scaling constant in the CCI denominator.      |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

|                |                                                       |
|----------------|-------------------------------------------------------|
| calc_chaikinAD | <i>Add Chaikin Accumulation Distribution Features</i> |
|----------------|-------------------------------------------------------|

---

**Description**

Computes the cumulative Chaikin Accumulation Distribution line from candle highs, lows, closes, and traded volume in place.

**Usage**

```
calc_chaikinAD(DT)
```

**Arguments**

|    |                                                                 |
|----|-----------------------------------------------------------------|
| DT | A 'data.table' containing 'high', 'low', 'close', and 'volume'. |
|----|-----------------------------------------------------------------|

**Value**

The input 'DT', modified by reference and returned invisibly.

---

|                        |                                        |
|------------------------|----------------------------------------|
| calc_chaikinVolatility | <i>Add Chaikin Volatility Features</i> |
|------------------------|----------------------------------------|

---

**Description**

Computes Chaikin Volatility columns from candle highs and lows in place using an EMA of the range followed by a discrete rate of change.

**Usage**

```
calc_chaikinVolatility(DT, ns = c(10))
```

**Arguments**

|    |                                             |
|----|---------------------------------------------|
| DT | A 'data.table' containing 'high' and 'low'. |
| ns | Integer vector of EMA and ROC windows.      |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

`calc_CMF`*Add Chaikin Money Flow Features*

---

**Description**

Computes Chaikin Money Flow columns from candle highs, lows, closes, and traded volume in place.

**Usage**

```
calc_CMF(DT, ns = c(20))
```

**Arguments**

|    |                                                                 |
|----|-----------------------------------------------------------------|
| DT | A 'data.table' containing 'high', 'low', 'close', and 'volume'. |
| ns | Integer vector of CMF windows.                                  |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

`calc_CMO`*Add Chande Momentum Oscillator Features*

---

**Description**

Computes CMO columns from the 'close' series of a candle 'data.table' in place.

**Usage**

```
calc_CMO(DT, ns = c(14))
```

**Arguments**

|    |                                             |
|----|---------------------------------------------|
| DT | A 'data.table' containing a 'close' column. |
| ns | Integer vector of CMO windows.              |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

 calc\_contango\_backwardation

*Compute Contango-Backwardation Regime*


---

### Description

Classifies the front-versus-deferred futures curve regime per date as contango ('1'), backwardation ('-1'), or flat ('0').

### Usage

```
calc_contango_backwardation(
  DT,
  date_col = "date",
  rank_col = "contract_rank",
  price_col = "close",
  rank_front = 1L,
  rank_deferred = 2L
)
```

### Arguments

|               |                                                                                 |
|---------------|---------------------------------------------------------------------------------|
| DT            | A 'data.table' futures panel containing date, contract-rank, and price columns. |
| date_col      | Date or timestamp column name.                                                  |
| rank_col      | Contract-rank column name.                                                      |
| price_col     | Price column name.                                                              |
| rank_front    | Integer front-contract rank.                                                    |
| rank_deferred | Integer deferred-contract rank.                                                 |

### Value

A 'data.table' with one regime code per date.

---

 calc\_coupon\_schedule    *Compute Coupon Schedule*


---

### Description

Generates a regular coupon-payment schedule from issue date to maturity for a plain-vanilla fixed-coupon bond without stub handling.

### Usage

```
calc_coupon_schedule(issue_date, maturity_date, freq = 2)
```

**Arguments**

|               |                                    |
|---------------|------------------------------------|
| issue_date    | Scalar issue date.                 |
| maturity_date | Scalar maturity date.              |
| freq          | Integer coupon frequency per year. |

**Value**

A vector of coupon-payment 'Date's after 'issue\_date' and up to 'maturity\_date'.

---

|                    |                                   |
|--------------------|-----------------------------------|
| calc_credit_spread | <i>Add Credit-Spread Features</i> |
|--------------------|-----------------------------------|

---

**Description**

Computes credit-spread columns in place as the difference between an issuer yield and a benchmark yield.

**Usage**

```
calc_credit_spread(  
  DT,  
  yield_col = "ytm",  
  benchmark_yield_col = "benchmark_ytm",  
  name = NULL  
)
```

**Arguments**

|                     |                                                               |
|---------------------|---------------------------------------------------------------|
| DT                  | A 'data.table' containing issuer and benchmark yield columns. |
| yield_col           | Issuer-yield column name.                                     |
| benchmark_yield_col | Benchmark-yield column name.                                  |
| name                | Optional output column name.                                  |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

|                      |                                      |
|----------------------|--------------------------------------|
| calc_curve_butterfly | <i>Compute Yield Curve Butterfly</i> |
|----------------------|--------------------------------------|

---

**Description**

Computes a simple annualized butterfly measure as the belly yield minus the average of wing yields.

**Usage**

```
calc_curve_butterfly(short_rate, mid_rate, long_rate)
```

**Arguments**

|            |                                     |
|------------|-------------------------------------|
| short_rate | Numeric annualized short-end yield. |
| mid_rate   | Numeric annualized belly yield.     |
| long_rate  | Numeric annualized long-end yield.  |

**Value**

Numeric scalar butterfly measure.

---

|                  |                                |
|------------------|--------------------------------|
| calc_curve_shock | <i>Shock A Zero-Rate Curve</i> |
|------------------|--------------------------------|

---

**Description**

Applies either a parallel or tenor-specific basis-point shock to an annualized zero-rate curve.

**Usage**

```
calc_curve_shock(tenor, zero_rate, shock_bp, shock_tenor = NULL)
```

**Arguments**

|             |                                                                                        |
|-------------|----------------------------------------------------------------------------------------|
| tenor       | Numeric vector of curve tenors in years.                                               |
| zero_rate   | Numeric vector of annualized zero rates aligned with 'tenor'.                          |
| shock_bp    | Numeric shock size in basis points.                                                    |
| shock_tenor | Optional scalar tenor to shock. If 'NULL', applies a parallel shift to the full curve. |

**Value**

A 'data.table' with 'tenor', 'zero\_rate', and 'zero\_rate\_shocked'.

---

|                  |                                  |
|------------------|----------------------------------|
| calc_curve_slope | <i>Compute Yield Curve Slope</i> |
|------------------|----------------------------------|

---

**Description**

Computes the annualized slope between a short and a long point on the yield curve.

**Usage**

```
calc_curve_slope(short_rate, long_rate)
```

**Arguments**

|            |                                     |
|------------|-------------------------------------|
| short_rate | Numeric annualized short-end yield. |
| long_rate  | Numeric annualized long-end yield.  |

**Value**

Numeric scalar curve slope.

---

|                      |                                          |
|----------------------|------------------------------------------|
| calc_curve_zero_rate | <i>Interpolate Zero Rates On A Curve</i> |
|----------------------|------------------------------------------|

---

**Description**

Interpolates annualized zero rates from an input tenor grid onto requested output tenors.

**Usage**

```
calc_curve_zero_rate(tenor_out, tenor, zero_rate, method = c("linear"))
```

**Arguments**

|           |                                                                       |
|-----------|-----------------------------------------------------------------------|
| tenor_out | Numeric vector of requested output tenors in years.                   |
| tenor     | Numeric vector of input curve tenors in years.                        |
| zero_rate | Numeric vector of annualized zero rates aligned with 'tenor'.         |
| method    | Character interpolation method. Currently only "linear" is supported. |

**Value**

Numeric vector of interpolated annualized zero rates.

---

|                     |                                   |
|---------------------|-----------------------------------|
| calc_day_count_frac | <i>Compute Day-Count Fraction</i> |
|---------------------|-----------------------------------|

---

**Description**

Computes the year fraction between two dates under a small set of common fixed-income day-count conventions.

**Usage**

```
calc_day_count_frac(
  start_date,
  end_date,
  convention = c("ACT/365", "ACT/360", "30/360")
)
```

**Arguments**

|            |                                                                            |
|------------|----------------------------------------------------------------------------|
| start_date | Scalar start date.                                                         |
| end_date   | Scalar end date.                                                           |
| convention | Character scalar. Supported values are "ACT/365", "ACT/360", and "30/360". |

**Value**

Numeric scalar year fraction.

---

|           |                                                       |
|-----------|-------------------------------------------------------|
| calc_DEMA | <i>Add Double Exponential Moving Average Features</i> |
|-----------|-------------------------------------------------------|

---

**Description**

Computes DEMA columns from the 'close' series of a candle 'data.table' in place.

**Usage**

```
calc_DEMA(DT, ns = c(10, 20, 50), v = 1)
```

**Arguments**

|    |                                                             |
|----|-------------------------------------------------------------|
| DT | A 'data.table' containing a 'close' column.                 |
| ns | Integer vector of EMA window sizes.                         |
| v  | Numeric blending factor. '1' matches the conventional DEMA. |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

**calc\_DonchianChannels** *Add Donchian Channel Features*

---

**Description**

Computes rolling channel highs, lows, and midpoints from ‘high’ and ‘low’ price columns on a candle ‘data.table’ in place.

**Usage**

```
calc_DonchianChannels(DT, ns = c(20, 55))
```

**Arguments**

|    |                                                  |
|----|--------------------------------------------------|
| DT | A ‘data.table’ containing ‘high’ and ‘low’.      |
| ns | Integer vector of Donchian channel window sizes. |

**Value**

The input ‘DT’, modified by reference and returned invisibly.

---

**calc\_EMA** *Add Exponential Moving Average Features*

---

**Description**

Computes EMA columns from the ‘close’ series of a candle ‘data.table’ in place.

**Usage**

```
calc_EMA(DT, ns = c(5, 8, 9, 10, 12, 20, 21, 26, 30, 50, 100, 200))
```

**Arguments**

|    |                                             |
|----|---------------------------------------------|
| DT | A ‘data.table’ containing a ‘close’ column. |
| ns | Integer vector of EMA window sizes.         |

**Value**

The input ‘DT’, modified by reference and returned invisibly.

---

|                |                                       |
|----------------|---------------------------------------|
| calc_EMA_cross | <i>Add EMA-Cross Strategy Targets</i> |
|----------------|---------------------------------------|

---

### Description

Evaluates a simple EMA-cross trading rule with ATR-based gating and writes either diagnostic columns or a target-position column to a candle 'data.table' in place.

### Usage

```
calc_EMA_cross(
  DT,
  fast = 20,
  slow = 50,
  low_atr_threshold = 5,
  freshness_floor = 18,
  tp_ratio = 0.05,
  sl_ratio = 0.02,
  debug_mode = FALSE
)
```

### Arguments

|                   |                                                                                      |
|-------------------|--------------------------------------------------------------------------------------|
| DT                | A 'data.table' containing the feature columns required by the strategy.              |
| fast              | Integer fast EMA length.                                                             |
| slow              | Integer slow EMA length.                                                             |
| low_atr_threshold | Numeric percentile threshold used by the low-volatility gate.                        |
| freshness_floor   | Integer maximum cross age allowed for a live target.                                 |
| tp_ratio          | Numeric take-profit guard.                                                           |
| sl_ratio          | Numeric stop-loss guard.                                                             |
| debug_mode        | Logical; when 'TRUE', writes diagnostic columns instead of a target-position column. |

### Value

The input 'DT', modified by reference and returned invisibly.

---

|          |                                      |
|----------|--------------------------------------|
| calc_EMV | <i>Add Ease of Movement Features</i> |
|----------|--------------------------------------|

---

**Description**

Computes the raw Ease of Movement series and smoothed moving-average columns from candle highs, lows, and traded volume in place.

**Usage**

```
calc_EMV(DT, ns = c(9), vol_divisor = 10000)
```

**Arguments**

|             |                                                                       |
|-------------|-----------------------------------------------------------------------|
| DT          | A 'data.table' containing 'high', 'low', and 'volume'.                |
| ns          | Integer vector of smoothing windows for the moving-average EMV.       |
| vol_divisor | Numeric divisor applied to 'volume' before the box-ratio calculation. |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

|                    |                                   |
|--------------------|-----------------------------------|
| calc_excess_spread | <i>Add Excess-Spread Features</i> |
|--------------------|-----------------------------------|

---

**Description**

Computes excess-spread columns in place as the difference between a credit spread and a benchmark or sector spread.

**Usage**

```
calc_excess_spread(
  DT,
  spread_col = "credit_spread",
  benchmark_spread_col = "benchmark_spread",
  name = NULL
)
```

**Arguments**

|                      |                                           |
|----------------------|-------------------------------------------|
| DT                   | A 'data.table' containing spread columns. |
| spread_col           | Credit-spread column name.                |
| benchmark_spread_col | Benchmark or sector spread column name.   |
| name                 | Optional output column name.              |

**Value**

The input ‘DT’, modified by reference and returned invisibly.

---

|                                          |
|------------------------------------------|
| calc_front_next_spread                   |
| <i>Compute Front-Next Futures Spread</i> |

---

**Description**

Computes the front-versus-next futures spread per date from a futures curve panel.

**Usage**

```
calc_front_next_spread(  
  DT,  
  date_col = "date",  
  rank_col = "contract_rank",  
  price_col = "close",  
  rank_front = 1L,  
  rank_next = 2L  
)
```

**Arguments**

|            |                                                                                 |
|------------|---------------------------------------------------------------------------------|
| DT         | A ‘data.table’ futures panel containing date, contract-rank, and price columns. |
| date_col   | Date or timestamp column name.                                                  |
| rank_col   | Contract-rank column name. Rank ‘1’ is the front contract.                      |
| price_col  | Price column name.                                                              |
| rank_front | Integer front-contract rank.                                                    |
| rank_next  | Integer deferred-contract rank used as the next contract.                       |

**Value**

A ‘data.table’ with front price, next price, and absolute/relative spread per date.

---

```
calc_futures_curve_butterfly
```

*Compute Futures Curve Butterfly*

---

**Description**

Computes a simple three-point butterfly from selected futures-curve ranks.

**Usage**

```
calc_futures_curve_butterfly(  
  DT,  
  date_col = "date",  
  rank_col = "contract_rank",  
  price_col = "close",  
  ranks = c(1L, 2L, 3L)  
)
```

**Arguments**

|           |                                                                                  |
|-----------|----------------------------------------------------------------------------------|
| DT        | A 'data.table' futures panel containing date, contract-rank, and price columns.  |
| date_col  | Date or timestamp column name.                                                   |
| rank_col  | Contract-rank column name.                                                       |
| price_col | Price column name.                                                               |
| ranks     | Integer vector of length three giving the front, belly, and back contract ranks. |

**Value**

A 'data.table' with one butterfly value per date.

---

```
calc_futures_curve_slope
```

*Compute Futures Curve Slope*

---

**Description**

Computes the per-date linear slope of log futures price against time to expiry.

**Usage**

```
calc_futures_curve_slope(  
  DT,  
  date_col = "date",  
  time_to_expiry_col = "time_to_expiry",  
  price_col = "close"  
)
```

Arguments

|                    |                                                                                  |
|--------------------|----------------------------------------------------------------------------------|
| DT                 | A 'data.table' futures panel containing date, time-to-expiry, and price columns. |
| date_col           | Date or timestamp column name.                                                   |
| time_to_expiry_col | Time-to-expiry column name expressed in years or a consistent time unit.         |
| price_col          | Price column name.                                                               |

Value

A 'data.table' with one slope estimate per date.

---

|               |                              |
|---------------|------------------------------|
| calc_fx_basis | <i>Add FX Basis Features</i> |
|---------------|------------------------------|

---

Description

Computes FX basis columns as the gap between observed forward points and covered-interest-parity-implied forward points.

Usage

```
calc_fx_basis(  
  DT,  
  spot_col = "spot",  
  forward_col = "forward",  
  domestic_rate_col = "r_domestic",  
  foreign_rate_col = "r_foreign",  
  tenor_col = "tenor_years",  
  tenor_tag = "1m"  
)
```

Arguments

|                   |                                                                                   |
|-------------------|-----------------------------------------------------------------------------------|
| DT                | A 'data.table' containing spot, forward, domestic-rate, and foreign-rate columns. |
| spot_col          | Spot-rate column name.                                                            |
| forward_col       | Forward-rate column name.                                                         |
| domestic_rate_col | Domestic annualized rate column name.                                             |
| foreign_rate_col  | Foreign annualized rate column name.                                              |
| tenor_col         | Tenor-in-years column name.                                                       |
| tenor_tag         | Character label appended to the output column name.                               |

Value

The input 'DT', modified by reference and returned invisibly.

---

`calc_fx_carry`*Add FX Carry Features*

---

**Description**

Computes FX carry columns from domestic and foreign interest rates in place.

**Usage**

```
calc_fx_carry(  
  DT,  
  domestic_rate_col = "r_domestic",  
  foreign_rate_col = "r_foreign",  
  tenor_col = NULL,  
  tenor_tag = "1m"  
)
```

**Arguments**

|                                |                                                                                                    |
|--------------------------------|----------------------------------------------------------------------------------------------------|
| <code>DT</code>                | A 'data.table' containing domestic and foreign rate columns.                                       |
| <code>domestic_rate_col</code> | Domestic annualized rate column name.                                                              |
| <code>foreign_rate_col</code>  | Foreign annualized rate column name.                                                               |
| <code>tenor_col</code>         | Optional tenor-in-years column name. If 'NULL', carry is reported as annualized rate differential. |
| <code>tenor_tag</code>         | Character label appended to the output column name.                                                |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

`calc_fx_forward_points`*Add FX Forward-Points Features*

---

**Description**

Computes forward-points columns from spot and forward FX rates in place.

**Usage**

```
calc_fx_forward_points(
  DT,
  spot_col = "spot",
  forward_col = "forward",
  tenor_tag = "1m"
)
```

**Arguments**

|             |                                                     |
|-------------|-----------------------------------------------------|
| DT          | A 'data.table' containing spot and forward columns. |
| spot_col    | Spot-rate column name.                              |
| forward_col | Forward-rate column name.                           |
| tenor_tag   | Character label appended to the output column name. |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

calc\_fx\_realized\_carry

*Add FX Realized-Carry Features*

---

**Description**

Computes realized-carry columns from spot return plus accrual carry over the selected horizon.

**Usage**

```
calc_fx_realized_carry(
  DT,
  spot_col = "spot",
  domestic_rate_col = "r_domestic",
  foreign_rate_col = "r_foreign",
  tenor_col = NULL,
  annualization = 252,
  tenor_tag = "1m"
)
```

**Arguments**

|                   |                                                                   |
|-------------------|-------------------------------------------------------------------|
| DT                | A 'data.table' containing spot and domestic/foreign rate columns. |
| spot_col          | Spot-rate column name.                                            |
| domestic_rate_col | Domestic annualized rate column name.                             |

|                  |                                                                                                      |
|------------------|------------------------------------------------------------------------------------------------------|
| foreign_rate_col | Foreign annualized rate column name.                                                                 |
| tenor_col        | Optional tenor-in-years column name. If 'NULL', a one-period accrual of '1 / annualization' is used. |
| annualization    | Numeric annualization factor used when 'tenor_col' is 'NULL'.                                        |
| tenor_tag        | Character label appended to the output column name.                                                  |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

|          |                                         |
|----------|-----------------------------------------|
| calc_HMA | <i>Add Hull Moving Average Features</i> |
|----------|-----------------------------------------|

---

**Description**

Computes HMA columns from the 'close' series of a candle 'data.table' in place using the standard weighted-moving-average construction.

**Usage**

```
calc_HMA(DT, ns = c(9, 16, 20, 50))
```

**Arguments**

|    |                                             |
|----|---------------------------------------------|
| DT | A 'data.table' containing a 'close' column. |
| ns | Integer vector of HMA window sizes.         |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

|                      |                                     |
|----------------------|-------------------------------------|
| calc_KeltnerChannels | <i>Add Keltner Channel Features</i> |
|----------------------|-------------------------------------|

---

**Description**

Computes Keltner channel mid, upper, and lower bands from an EMA centerline and ATR envelope on a candle 'data.table' in place.

**Usage**

```
calc_KeltnerChannels(DT, ns = c(20), ks = c(2))
```

**Arguments**

|    |                                                       |
|----|-------------------------------------------------------|
| DT | A 'data.table' containing 'high', 'low', and 'close'. |
| ns | Integer vector of EMA/ATR window sizes.               |
| ks | Numeric vector of ATR multipliers.                    |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

|          |                                     |
|----------|-------------------------------------|
| calc_KST | <i>Add Know Sure Thing Features</i> |
|----------|-------------------------------------|

---

**Description**

Computes KST and signal columns from weighted moving averages of multiple rate-of-change series on a candle 'data.table' in place.

**Usage**

```
calc_KST(
  DT,
  nss = list(c(10, 10, 10, 15)),
  n_rocss = list(c(10, 15, 20, 30)),
  signal_ns = c(9),
  wtss = list(1:4)
)
```

**Arguments**

|           |                                                                                |
|-----------|--------------------------------------------------------------------------------|
| DT        | A 'data.table' containing a 'close' column.                                    |
| nss       | A list of integer vectors giving the smoothing windows for each ROC component. |
| n_rocss   | A list of integer vectors giving the ROC lags.                                 |
| signal_ns | Integer vector of signal-line windows.                                         |
| wtss      | A list of numeric vectors giving the component weights.                        |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

|               |                                      |
|---------------|--------------------------------------|
| calc_kurtosis | <i>Add Rolling Kurtosis Features</i> |
|---------------|--------------------------------------|

---

**Description**

Computes rolling excess-kurtosis columns in place for the selected input columns.

**Usage**

```
calc_kurtosis(DT, cols = "close", ns = c(20, 60))
```

**Arguments**

|      |                                                       |
|------|-------------------------------------------------------|
| DT   | A 'data.table' containing the selected input columns. |
| cols | Character vector of numeric columns.                  |
| ns   | Integer vector of rolling window sizes.               |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

|                   |                                            |
|-------------------|--------------------------------------------|
| calc_ladder_index | <i>Add Signed Fibonacci Ladder Indices</i> |
|-------------------|--------------------------------------------|

---

**Description**

Detects recent pivot cycles and maps each bar's close to a signed Fibonacci ladder index. Positive and negative signs encode cycle direction, while the absolute value encodes the ladder level selected by the native engine.

**Usage**

```
calc_ladder_index(
  DT,
  span = 3,
  latest_n = NULL,
  refined = TRUE,
  min_swing = 0.05,
  cycle_N = 360L,
  cycle_prefix = NULL,
  center_idx = 9L,
  detailed_report = FALSE
)
```

Arguments

|                 |                                                                          |
|-----------------|--------------------------------------------------------------------------|
| DT              | A candle 'data.table' containing 'datetime', 'high', 'low', and 'close'. |
| span            | Integer pivot span passed to the pivot detector.                         |
| latest_n        | Optional integer tail length for pivot detection.                        |
| refined         | Logical; whether to refine raw pivots before cycle detection.            |
| min_swing       | Minimum relative swing used by the pivot refinement step.                |
| cycle_N         | Integer cycle lookback width in bars.                                    |
| cycle_prefix    | Optional suffix used in generated column names.                          |
| center_idx      | Zero-based native center index for the ladder mapping.                   |
| detailed_report | Logical; when 'TRUE', cycle boundary columns are also added.             |

Value

The input 'DT', modified by reference and returned invisibly.

---

|                 |                                |
|-----------------|--------------------------------|
| calc_log_spread | <i>Add Log-Spread Features</i> |
|-----------------|--------------------------------|

---

Description

Computes log-price spread columns between two positive numeric input columns in place.

Usage

```
calc_log_spread(DT, x_col = "close", y_col = "benchmark_close", name = NULL)
```

Arguments

|       |                                                       |
|-------|-------------------------------------------------------|
| DT    | A 'data.table' containing the selected input columns. |
| x_col | First numeric column name.                            |
| y_col | Second numeric column name.                           |
| name  | Optional output column name.                          |

Value

The input 'DT', modified by reference and returned invisibly.

---

|           |                          |
|-----------|--------------------------|
| calc_MACD | <i>Add MACD Features</i> |
|-----------|--------------------------|

---

**Description**

Computes Moving Average Convergence Divergence, signal, and histogram columns from the 'close' series of a candle 'data.table' in place.

**Usage**

```
calc_MACD(DT, fast = 12, slow = 26, signal = 9)
```

**Arguments**

|        |                                             |
|--------|---------------------------------------------|
| DT     | A 'data.table' containing a 'close' column. |
| fast   | Integer fast EMA window.                    |
| slow   | Integer slow EMA window.                    |
| signal | Integer signal EMA window.                  |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

|          |                                      |
|----------|--------------------------------------|
| calc_MFI | <i>Add Money Flow Index Features</i> |
|----------|--------------------------------------|

---

**Description**

Computes Money Flow Index columns from candle highs, lows, closes, and traded volume in place.

**Usage**

```
calc_MFI(DT, ns = c(14))
```

**Arguments**

|    |                                                                 |
|----|-----------------------------------------------------------------|
| DT | A 'data.table' containing 'high', 'low', 'close', and 'volume'. |
| ns | Integer vector of MFI windows.                                  |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

|                 |                                |
|-----------------|--------------------------------|
| calc_microprice | <i>Add Microprice Features</i> |
|-----------------|--------------------------------|

---

**Description**

Computes microprice columns from bid/ask quotes and queue sizes in place.

**Usage**

```
calc_microprice(  
  DT,  
  bid_col = "bid",  
  ask_col = "ask",  
  bid_size_col = "bid_size",  
  ask_size_col = "ask_size"  
)
```

**Arguments**

|              |                                                                     |
|--------------|---------------------------------------------------------------------|
| DT           | A 'data.table' containing bid, ask, bid-size, and ask-size columns. |
| bid_col      | Bid-price column name.                                              |
| ask_col      | Ask-price column name.                                              |
| bid_size_col | Bid-size column name.                                               |
| ask_size_col | Ask-size column name.                                               |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

|                |                               |
|----------------|-------------------------------|
| calc_mid_price | <i>Add Mid-Price Features</i> |
|----------------|-------------------------------|

---

**Description**

Computes mid-price columns from bid and ask quotes in place.

**Usage**

```
calc_mid_price(DT, bid_col = "bid", ask_col = "ask")
```

**Arguments**

|         |                                                |
|---------|------------------------------------------------|
| DT      | A 'data.table' containing bid and ask columns. |
| bid_col | Bid-price column name.                         |
| ask_col | Ask-price column name.                         |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

|                |                                           |
|----------------|-------------------------------------------|
| calc_normalize | <i>Add Min-Max Normalization Features</i> |
|----------------|-------------------------------------------|

---

**Description**

Computes rolling min-max normalized columns in place for the selected input columns.

**Usage**

```
calc_normalize(DT, cols = "close", ns = c(20, 60))
```

**Arguments**

|      |                                                       |
|------|-------------------------------------------------------|
| DT   | A 'data.table' containing the selected input columns. |
| cols | Character vector of numeric columns to normalize.     |
| ns   | Integer vector of rolling window sizes.               |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

|          |                                       |
|----------|---------------------------------------|
| calc_OBV | <i>Add On-Balance Volume Features</i> |
|----------|---------------------------------------|

---

**Description**

Computes the cumulative on-balance volume series from candle closes and traded volume in place.

**Usage**

```
calc_OBV(DT)
```

**Arguments**

|    |                                                 |
|----|-------------------------------------------------|
| DT | A 'data.table' containing 'close' and 'volume'. |
|----|-------------------------------------------------|

**Value**

The input 'DT', modified by reference and returned invisibly.

---

|                   |                             |
|-------------------|-----------------------------|
| calc_option_delta | <i>Compute Option Delta</i> |
|-------------------|-----------------------------|

---

**Description**

Computes Black-Scholes delta for a European call or put with continuous dividend yield.

**Usage**

```
calc_option_delta(
    S,
    K,
    time_to_expiry,
    r,
    sigma,
    type = c("call", "put"),
    q = 0
)
```

**Arguments**

|                |                                               |
|----------------|-----------------------------------------------|
| S              | Numeric spot price.                           |
| K              | Numeric strike price.                         |
| time_to_expiry | Numeric time to expiry in years.              |
| r              | Numeric annualized risk-free rate.            |
| sigma          | Numeric annualized volatility.                |
| type           | Character scalar, either "call" or "put".     |
| q              | Numeric annualized continuous dividend yield. |

**Value**

Numeric scalar delta.

---

|                               |                                              |
|-------------------------------|----------------------------------------------|
| calc_option_forward_moneyness | <i>Add Option Forward-Moneyness Features</i> |
|-------------------------------|----------------------------------------------|

---

**Description**

Computes forward-based option moneyness columns in place from spot, strike, time to expiry, rates, and dividend yield.

Usage

```
calc_option_forward_moneyness(  
  DT,  
  spot_col = "S",  
  strike_col = "K",  
  expiry_col = "time_to_expiry",  
  rate_col = "r",  
  dividend_col = "q"  
)
```

Arguments

|              |                                                                                |
|--------------|--------------------------------------------------------------------------------|
| DT           | A 'data.table' option chain containing spot, strike, expiry, and rate columns. |
| spot_col     | Underlying spot-price column name.                                             |
| strike_col   | Strike column name.                                                            |
| expiry_col   | Time-to-expiry column name in years.                                           |
| rate_col     | Risk-free rate column name.                                                    |
| dividend_col | Dividend-yield column name.                                                    |

Value

The input 'DT', modified by reference and returned invisibly.

---

|                   |                             |
|-------------------|-----------------------------|
| calc_option_gamma | <i>Compute Option Gamma</i> |
|-------------------|-----------------------------|

---

Description

Computes Black-Scholes gamma for a European option with continuous dividend yield.

Usage

```
calc_option_gamma(S, K, time_to_expiry, r, sigma, q = 0)
```

Arguments

|                |                                               |
|----------------|-----------------------------------------------|
| S              | Numeric spot price.                           |
| K              | Numeric strike price.                         |
| time_to_expiry | Numeric time to expiry in years.              |
| r              | Numeric annualized risk-free rate.            |
| sigma          | Numeric annualized volatility.                |
| q              | Numeric annualized continuous dividend yield. |

Value

Numeric scalar gamma.

---

|                    |                              |
|--------------------|------------------------------|
| calc_option_greeks | <i>Compute Option Greeks</i> |
|--------------------|------------------------------|

---

## Description

Computes Black-Scholes price and Greeks for a European call or put with continuous dividend yield.

## Usage

```
calc_option_greeks(
  S,
  K,
  time_to_expiry,
  r,
  sigma,
  type = c("call", "put"),
  q = 0,
  theta_scale = c("annual", "daily")
)
```

## Arguments

|                |                                                                       |
|----------------|-----------------------------------------------------------------------|
| S              | Numeric spot price.                                                   |
| K              | Numeric strike price.                                                 |
| time_to_expiry | Numeric time to expiry in years.                                      |
| r              | Numeric annualized risk-free rate.                                    |
| sigma          | Numeric annualized volatility.                                        |
| type           | Character scalar, either "call" or "put".                             |
| q              | Numeric annualized continuous dividend yield.                         |
| theta_scale    | Character scalar controlling theta units, either "annual" or "daily". |

## Value

A one-row 'data.table' with 'price', 'delta', 'gamma', 'vega', 'theta', and 'rho'.

---

|                |                                          |
|----------------|------------------------------------------|
| calc_option_iv | <i>Compute Option Implied Volatility</i> |
|----------------|------------------------------------------|

---

**Description**

Solves Black-Scholes implied volatility from an observed European option price with continuous dividend yield.

**Usage**

```
calc_option_iv(  
  price,  
  S,  
  K,  
  time_to_expiry,  
  r,  
  type = c("call", "put"),  
  q = 0,  
  interval = c(1e-06, 5)  
)
```

**Arguments**

|                |                                                                    |
|----------------|--------------------------------------------------------------------|
| price          | Numeric observed option price.                                     |
| S              | Numeric spot price.                                                |
| K              | Numeric strike price.                                              |
| time_to_expiry | Numeric time to expiry in years.                                   |
| r              | Numeric annualized risk-free rate.                                 |
| type           | Character scalar, either "call" or "put".                          |
| q              | Numeric annualized continuous dividend yield.                      |
| interval       | Numeric vector of length two giving the volatility search bracket. |

**Value**

Numeric scalar implied volatility.

---

|                     |                                               |
|---------------------|-----------------------------------------------|
| calc_option_iv_skew | <i>Compute Option Implied-Volatility Skew</i> |
|---------------------|-----------------------------------------------|

---

**Description**

Computes per-date and per-expiry put-minus-call IV skew using strikes nearest to a target absolute forward-moneyness level.

**Usage**

```
calc_option_iv_skew(  
  DT,  
  date_col = "date",  
  expiry_col = "time_to_expiry",  
  type_col = "type",  
  moneyness_col = "option_log_forward_moneyness",  
  iv_col = "iv",  
  target_abs_moneyness = 0.1  
)
```

**Arguments**

|                      |                                                                                       |
|----------------------|---------------------------------------------------------------------------------------|
| DT                   | A 'data.table' option chain containing date, expiry, type, moneyness, and IV columns. |
| date_col             | Date or timestamp column name.                                                        |
| expiry_col           | Time-to-expiry column name.                                                           |
| type_col             | Option-type column name containing 'call'/'put'.                                      |
| moneyness_col        | Forward-moneyness feature column name.                                                |
| iv_col               | Implied-volatility column name.                                                       |
| target_abs_moneyness | Numeric target absolute log-moneyness.                                                |

**Value**

A 'data.table' with IV skew per date and expiry.

---

 calc\_option\_iv\_term\_structure

*Compute Option Implied-Volatility Term Structure*


---

### Description

Computes per-date ATM IV term-structure slope using the nearest-to-ATM option at each expiry.

### Usage

```
calc_option_iv_term_structure(
  DT,
  date_col = "date",
  expiry_col = "time_to_expiry",
  moneyness_col = "option_log_forward_moneyness",
  iv_col = "iv"
)
```

### Arguments

|               |                                                                                 |
|---------------|---------------------------------------------------------------------------------|
| DT            | A 'data.table' option chain containing date, expiry, moneyness, and IV columns. |
| date_col      | Date or timestamp column name.                                                  |
| expiry_col    | Time-to-expiry column name.                                                     |
| moneyness_col | Forward-moneyness feature column name.                                          |
| iv_col        | Implied-volatility column name.                                                 |

### Value

A 'data.table' with ATM-front, ATM-back, and ATM-term-structure slope per date.

---

 calc\_option\_moneyness *Add Option Moneyness Features*


---

### Description

Computes spot-based option moneyness columns in place from underlying spot and strike.

### Usage

```
calc_option_moneyness(DT, spot_col = "S", strike_col = "K")
```

### Arguments

|            |                                                                 |
|------------|-----------------------------------------------------------------|
| DT         | A 'data.table' option chain containing spot and strike columns. |
| spot_col   | Underlying spot-price column name.                              |
| strike_col | Strike column name.                                             |

**Value**

The input ‘DT’, modified by reference and returned invisibly.

---

|                                          |
|------------------------------------------|
| calc_option_put_call_iv_spread           |
| <i>Compute Option Put-Call IV Spread</i> |

---

**Description**

Computes per-date and per-expiry put-minus-call ATM IV spread using the nearest-to-ATM option of each type.

**Usage**

```
calc_option_put_call_iv_spread(  
  DT,  
  date_col = "date",  
  expiry_col = "time_to_expiry",  
  type_col = "type",  
  moneyness_col = "option_log_forward_moneyness",  
  iv_col = "iv"  
)
```

**Arguments**

|               |                                                                                       |
|---------------|---------------------------------------------------------------------------------------|
| DT            | A ‘data.table’ option chain containing date, expiry, type, moneyness, and IV columns. |
| date_col      | Date or timestamp column name.                                                        |
| expiry_col    | Time-to-expiry column name.                                                           |
| type_col      | Option-type column name containing ‘call’/‘put’.                                      |
| moneyness_col | Forward-moneyness feature column name.                                                |
| iv_col        | Implied-volatility column name.                                                       |

**Value**

A ‘data.table’ with ATM put-call IV spread per date and expiry.

---

|                 |                           |
|-----------------|---------------------------|
| calc_option_rho | <i>Compute Option Rho</i> |
|-----------------|---------------------------|

---

**Description**

Computes Black-Scholes rho for a European call or put with continuous dividend yield.

**Usage**

```
calc_option_rho(S, K, time_to_expiry, r, sigma, type = c("call", "put"), q = 0)
```

**Arguments**

|                |                                               |
|----------------|-----------------------------------------------|
| S              | Numeric spot price.                           |
| K              | Numeric strike price.                         |
| time_to_expiry | Numeric time to expiry in years.              |
| r              | Numeric annualized risk-free rate.            |
| sigma          | Numeric annualized volatility.                |
| type           | Character scalar, either "call" or "put".     |
| q              | Numeric annualized continuous dividend yield. |

**Value**

Numeric scalar rho per 1.00 rate change.

---

|                        |                                  |
|------------------------|----------------------------------|
| calc_option_risk_state | <i>Compute Option Risk State</i> |
|------------------------|----------------------------------|

---

**Description**

Builds a compact option risk-state snapshot from Black-Scholes price, implied volatility, and Greeks.

**Usage**

```
calc_option_risk_state(
  S,
  K,
  time_to_expiry,
  r,
  sigma = NULL,
  type = c("call", "put"),
  q = 0,
  price = NULL
)
```

**Arguments**

|                |                                                                   |
|----------------|-------------------------------------------------------------------|
| S              | Numeric spot price.                                               |
| K              | Numeric strike price.                                             |
| time_to_expiry | Numeric time to expiry in years.                                  |
| r              | Numeric annualized risk-free rate.                                |
| sigma          | Optional numeric annualized volatility.                           |
| type           | Character scalar, either "call" or "put".                         |
| q              | Numeric annualized continuous dividend yield.                     |
| price          | Optional observed option price. Required when 'sigma' is missing. |

**Value**

A one-row 'data.table' with observed/model price, implied volatility, IV source, and Greeks.

---

calc\_option\_smile\_slope

*Compute Option Smile Slope*

---

**Description**

Computes per-date and per-expiry smile slope from a linear fit of implied volatility on absolute forward log-moneyness.

**Usage**

```
calc_option_smile_slope(
  DT,
  date_col = "date",
  expiry_col = "time_to_expiry",
  moneyness_col = "option_log_forward_moneyness",
  iv_col = "iv"
)
```

**Arguments**

|               |                                                                                 |
|---------------|---------------------------------------------------------------------------------|
| DT            | A 'data.table' option chain containing date, expiry, moneyness, and IV columns. |
| date_col      | Date or timestamp column name.                                                  |
| expiry_col    | Time-to-expiry column name.                                                     |
| moneyness_col | Forward-moneyness feature column name.                                          |
| iv_col        | Implied-volatility column name.                                                 |

**Value**

A 'data.table' with smile slope per date and expiry.

---

|                   |                             |
|-------------------|-----------------------------|
| calc_option_theta | <i>Compute Option Theta</i> |
|-------------------|-----------------------------|

---

**Description**

Computes Black-Scholes theta for a European call or put with continuous dividend yield.

**Usage**

```
calc_option_theta(  
  S,  
  K,  
  time_to_expiry,  
  r,  
  sigma,  
  type = c("call", "put"),  
  q = 0,  
  scale = c("annual", "daily")  
)
```

**Arguments**

- S                    Numeric spot price.
- K                    Numeric strike price.
- time\_to\_expiry    Numeric time to expiry in years.
- r                    Numeric annualized risk-free rate.
- sigma              Numeric annualized volatility.
- type                Character scalar, either "call" or "put".
- q                    Numeric annualized continuous dividend yield.
- scale                Character scalar, either "annual" or "daily".

**Value**

Numeric scalar theta.

---

|                  |                            |
|------------------|----------------------------|
| calc_option_vega | <i>Compute Option Vega</i> |
|------------------|----------------------------|

---

**Description**

Computes Black-Scholes vega for a European option with continuous dividend yield.

**Usage**

```
calc_option_vega(S, K, time_to_expiry, r, sigma, q = 0)
```

**Arguments**

|                |                                               |
|----------------|-----------------------------------------------|
| S              | Numeric spot price.                           |
| K              | Numeric strike price.                         |
| time_to_expiry | Numeric time to expiry in years.              |
| r              | Numeric annualized risk-free rate.            |
| sigma          | Numeric annualized volatility.                |
| q              | Numeric annualized continuous dividend yield. |

**Value**

Numeric scalar vega per 1.00 volatility change.

---

|                      |                                     |
|----------------------|-------------------------------------|
| calc_order_imbalance | <i>Add Order-Imbalance Features</i> |
|----------------------|-------------------------------------|

---

**Description**

Computes normalized order-imbalance columns from bid and ask queue sizes in place.

**Usage**

```
calc_order_imbalance(DT, bid_size_col = "bid_size", ask_size_col = "ask_size")
```

**Arguments**

|              |                                                          |
|--------------|----------------------------------------------------------|
| DT           | A 'data.table' containing bid-size and ask-size columns. |
| bid_size_col | Bid-size column name.                                    |
| ask_size_col | Ask-size column name.                                    |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

|                   |                                          |
|-------------------|------------------------------------------|
| calc_percent_rank | <i>Add Rolling Percent-Rank Features</i> |
|-------------------|------------------------------------------|

---

**Description**

Computes rolling percent-rank columns in place for the selected input columns. The current observation is ranked against the most recent ‘n’ values, including itself.

**Usage**

```
calc_percent_rank(DT, cols = "close", ns = c(20, 60))
```

**Arguments**

|      |                                                       |
|------|-------------------------------------------------------|
| DT   | A ‘data.table’ containing the selected input columns. |
| cols | Character vector of numeric columns.                  |
| ns   | Integer vector of rolling window sizes.               |

**Value**

The input ‘DT’, modified by reference and returned invisibly.

---

|                  |                                       |
|------------------|---------------------------------------|
| calc_period_rate | <i>Compute Per-Period Yield Rates</i> |
|------------------|---------------------------------------|

---

**Description**

Converts annualized yield-to-maturity values into per-period rates.

**Usage**

```
calc_period_rate(ytm, freq = 2)
```

**Arguments**

|      |                                                |
|------|------------------------------------------------|
| ytm  | Numeric scalar or vector of annualized yields. |
| freq | Integer compounding frequency per year.        |

**Value**

Numeric vector of per-period rates.

---

calc\_position\_greeks    *Compute Position Greeks*

---

### Description

Computes Black-Scholes Greeks per row and scales them by position size and contract size so the result is ready for portfolio aggregation.

### Usage

```
calc_position_greeks(
  option_state,
  contract_size_col = "contract_size",
  units_col = "units"
)
```

### Arguments

**option\_state**    A 'data.table' containing at least 'asset', 'S', 'K', 'time\_to\_expiry', 'r', 'sigma', 'type', and 'units'.

**contract\_size\_col**    Column name containing contract-size multipliers.

**units\_col**    Column name containing signed position units.

### Value

A 'data.table' with per-contract and position-scaled Greek columns.

---

calc\_price\_impact\_proxy    *Add Price-Impact Proxy Features*

---

### Description

Computes an Amihud-style price-impact proxy from absolute returns and dollar volume in place.

### Usage

```
calc_price_impact_proxy(
  DT,
  close_col = "close",
  volume_col = "volume",
  use_log = TRUE
)
```

**Arguments**

|            |                                                      |
|------------|------------------------------------------------------|
| DT         | A 'data.table' containing close and volume columns.  |
| close_col  | Close-price column name.                             |
| volume_col | Traded-volume column name.                           |
| use_log    | Logical; if 'TRUE', uses close-to-close log returns. |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

|            |                           |
|------------|---------------------------|
| calc_ratio | <i>Add Ratio Features</i> |
|------------|---------------------------|

---

**Description**

Computes arithmetic ratio columns between two numeric input columns in place.

**Usage**

```
calc_ratio(DT, x_col = "close", y_col = "benchmark_close", name = NULL)
```

**Arguments**

|       |                                                       |
|-------|-------------------------------------------------------|
| DT    | A 'data.table' containing the selected input columns. |
| x_col | First numeric column name.                            |
| y_col | Second numeric column name.                           |
| name  | Optional output column name.                          |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

|                   |                                         |
|-------------------|-----------------------------------------|
| calc_realized_vol | <i>Add Realized Volatility Features</i> |
|-------------------|-----------------------------------------|

---

**Description**

Computes annualized rolling realized-volatility columns from close-to-close log returns in place.

**Usage**

```
calc_realized_vol(DT, ns = c(10, 20, 60), annualization = 252)
```

**Arguments**

|               |                                                                    |
|---------------|--------------------------------------------------------------------|
| DT            | A 'data.table' containing a 'close' column.                        |
| ns            | Integer vector of rolling windows measured in return observations. |
| annualization | Numeric annualization factor. Defaults to '252'.                   |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

|                        |                                       |
|------------------------|---------------------------------------|
| calc_relative_strength | <i>Add Relative-Strength Features</i> |
|------------------------|---------------------------------------|

---

**Description**

Computes rolling relative-strength columns as the ratio of cumulative asset return to cumulative benchmark return over each lookback window.

**Usage**

```
calc_relative_strength(
  DT,
  x_col = "close",
  y_col = "benchmark_close",
  ns = c(20, 60),
  use_log = TRUE
)
```

**Arguments**

|         |                                                       |
|---------|-------------------------------------------------------|
| DT      | A 'data.table' containing the selected input columns. |
| x_col   | Asset-price column name.                              |
| y_col   | Benchmark-price column name.                          |
| ns      | Integer vector of rolling windows.                    |
| use_log | Logical; if 'TRUE', uses log-return aggregation.      |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

|          |                                    |
|----------|------------------------------------|
| calc_ROC | <i>Add Rate of Change Features</i> |
|----------|------------------------------------|

---

**Description**

Computes rate-of-change columns from the 'close' series of a candle 'data.table' in place.

**Usage**

```
calc_ROC(DT, ns = c(5, 10, 20), scale = 100)
```

**Arguments**

|       |                                                          |
|-------|----------------------------------------------------------|
| DT    | A 'data.table' containing a 'close' column.              |
| ns    | Integer vector of lag windows.                           |
| scale | Numeric scale factor, typically '100' for percent units. |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

|                   |                                  |
|-------------------|----------------------------------|
| calc_rolling_beta | <i>Add Rolling Beta Features</i> |
|-------------------|----------------------------------|

---

**Description**

Computes rolling beta columns from asset and benchmark return series.

**Usage**

```
calc_rolling_beta(  
  DT,  
  x_col = "close",  
  y_col = "benchmark_close",  
  ns = c(20, 60),  
  use_log = TRUE  
)
```

**Arguments**

|         |                                                       |
|---------|-------------------------------------------------------|
| DT      | A 'data.table' containing the selected input columns. |
| x_col   | Asset-price column name.                              |
| y_col   | Benchmark-price column name.                          |
| ns      | Integer vector of rolling windows.                    |
| use_log | Logical; if 'TRUE', uses log returns.                 |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

|                   |                                         |
|-------------------|-----------------------------------------|
| calc_rolling_corr | <i>Add Rolling Correlation Features</i> |
|-------------------|-----------------------------------------|

---

**Description**

Computes rolling correlation columns from asset and benchmark return series.

**Usage**

```
calc_rolling_corr(
  DT,
  x_col = "close",
  y_col = "benchmark_close",
  ns = c(20, 60),
  use_log = TRUE
)
```

**Arguments**

|         |                                                       |
|---------|-------------------------------------------------------|
| DT      | A 'data.table' containing the selected input columns. |
| x_col   | Asset-price column name.                              |
| y_col   | Benchmark-price column name.                          |
| ns      | Integer vector of rolling windows.                    |
| use_log | Logical; if 'TRUE', uses log returns.                 |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

`calc_rolling_quantile` *Add Rolling Quantile Features*

---

**Description**

Computes rolling quantile columns in place for the selected input columns.

**Usage**

```
calc_rolling_quantile(  
  DT,  
  cols = "close",  
  ns = c(20, 60),  
  probs = c(0.25, 0.5, 0.75)  
)
```

**Arguments**

|       |                                                       |
|-------|-------------------------------------------------------|
| DT    | A 'data.table' containing the selected input columns. |
| cols  | Character vector of numeric columns.                  |
| ns    | Integer vector of rolling window sizes.               |
| probs | Numeric vector of quantile probabilities in '[0, 1]'. |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

`calc_roll_yield` *Compute Annualized Roll Yield*

---

**Description**

Computes the annualized front-versus-deferred roll yield from a futures curve panel. Positive values indicate backwardation and negative values indicate contango under the selected pair.

**Usage**

```
calc_roll_yield(  
  DT,  
  date_col = "date",  
  rank_col = "contract_rank",  
  price_col = "close",  
  time_to_expiry_col = "time_to_expiry",  
  rank_front = 1L,  
  rank_deferred = 2L  
)
```

Arguments

|                    |                                                                                                 |
|--------------------|-------------------------------------------------------------------------------------------------|
| DT                 | A 'data.table' futures panel containing date, contract-rank, price, and time-to-expiry columns. |
| date_col           | Date or timestamp column name.                                                                  |
| rank_col           | Contract-rank column name.                                                                      |
| price_col          | Price column name.                                                                              |
| time_to_expiry_col | Time-to-expiry column name expressed in years.                                                  |
| rank_front         | Integer front-contract rank.                                                                    |
| rank_deferred      | Integer deferred-contract rank.                                                                 |

Value

A 'data.table' with annualized roll yield per date.

---

|          |                                             |
|----------|---------------------------------------------|
| calc_RSI | <i>Add Relative Strength Index Features</i> |
|----------|---------------------------------------------|

---

Description

Computes both classic Wilder-style RSI columns and strategyr-specific log-return RSI columns on a candle 'data.table' in place.

Usage

```
calc_RSI(DT, ns = c(9, 14, 21), hs = c(12, 24))
```

Arguments

|    |                                                                 |
|----|-----------------------------------------------------------------|
| DT | A 'data.table' containing a 'close' column.                     |
| ns | Optional integer vector of classic RSI windows.                 |
| hs | Optional numeric vector of half-life values for log-return RSI. |

Value

The input 'DT', modified by reference and returned invisibly.

---

|          |                                   |
|----------|-----------------------------------|
| calc_SAR | <i>Add Parabolic SAR Features</i> |
|----------|-----------------------------------|

---

**Description**

Computes Parabolic SAR columns from candle highs and lows in place using the classic acceleration step and maximum parameters.

**Usage**

```
calc_SAR(DT, accels = list(c(0.02, 0.2)))
```

**Arguments**

|        |                                                                                             |
|--------|---------------------------------------------------------------------------------------------|
| DT     | A 'data.table' containing 'high' and 'low'.                                                 |
| accels | A list of numeric vectors, each of length two, giving 'c(step, max)' acceleration settings. |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

|               |                                      |
|---------------|--------------------------------------|
| calc_skewness | <i>Add Rolling Skewness Features</i> |
|---------------|--------------------------------------|

---

**Description**

Computes rolling skewness columns in place for the selected input columns.

**Usage**

```
calc_skewness(DT, cols = "close", ns = c(20, 60))
```

**Arguments**

|      |                                                       |
|------|-------------------------------------------------------|
| DT   | A 'data.table' containing the selected input columns. |
| cols | Character vector of numeric columns.                  |
| ns   | Integer vector of rolling window sizes.               |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

|                     |                                    |
|---------------------|------------------------------------|
| calc_slippage_proxy | <i>Add Slippage-Proxy Features</i> |
|---------------------|------------------------------------|

---

**Description**

Computes a simple execution slippage proxy from intrabar range and close price in place.

**Usage**

```
calc_slippage_proxy(
  DT,
  high_col = "high",
  low_col = "low",
  close_col = "close"
)
```

**Arguments**

|           |                                                         |
|-----------|---------------------------------------------------------|
| DT        | A 'data.table' containing high, low, and close columns. |
| high_col  | High-price column name.                                 |
| low_col   | Low-price column name.                                  |
| close_col | Close-price column name.                                |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

|          |                                           |
|----------|-------------------------------------------|
| calc_SMA | <i>Add Simple Moving Average Features</i> |
|----------|-------------------------------------------|

---

**Description**

Computes SMA columns from the 'close' series of a candle 'data.table' in place.

**Usage**

```
calc_SMA(DT, ns = c(5, 8, 9, 10, 12, 20, 21, 26, 30, 50, 100, 200))
```

**Arguments**

|    |                                             |
|----|---------------------------------------------|
| DT | A 'data.table' containing a 'close' column. |
| ns | Integer vector of SMA window sizes.         |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

**calc\_SMI***Add Stochastic Momentum Index Features*

---

**Description**

Computes SMI and signal columns from candle highs, lows, and closes in place using double-smoothed EMA numerators and denominators.

**Usage**

```
calc_SMI(DT, n = 13, nFast = 2, nSlow = 25, nSig = 9, bounded = TRUE)
```

**Arguments**

|         |                                                        |
|---------|--------------------------------------------------------|
| DT      | A 'data.table' containing 'high', 'low', and 'close'.  |
| n       | Integer lookback window for the price range.           |
| nFast   | Integer fast EMA smoothing window.                     |
| nSlow   | Integer slow EMA smoothing window.                     |
| nSig    | Integer signal EMA smoothing window.                   |
| bounded | Logical; if 'TRUE', use current-window highs and lows. |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

**calc\_spread***Add Spread Features*

---

**Description**

Computes arithmetic spread columns between two numeric input columns in place.

**Usage**

```
calc_spread(DT, x_col = "close", y_col = "benchmark_close", name = NULL)
```

**Arguments**

|       |                                                       |
|-------|-------------------------------------------------------|
| DT    | A 'data.table' containing the selected input columns. |
| x_col | First numeric column name.                            |
| y_col | Second numeric column name.                           |
| name  | Optional output column name.                          |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

```
calc_spread_curve_butterfly
```

*Compute Spread-Curve Butterfly*

---

**Description**

Computes a three-point spread-curve butterfly per date using the nearest available tenors.

**Usage**

```
calc_spread_curve_butterfly(
  DT,
  date_col = "date",
  tenor_col = "tenor",
  spread_col = "credit_spread",
  tenors = c(2, 5, 10)
)
```

**Arguments**

|            |                                                                                 |
|------------|---------------------------------------------------------------------------------|
| DT         | A 'data.table' credit panel containing date, tenor, and spread columns.         |
| date_col   | Date or timestamp column name.                                                  |
| tenor_col  | Tenor column name.                                                              |
| spread_col | Credit-spread column name.                                                      |
| tenors     | Numeric vector of length three giving the front, belly, and back target tenors. |

**Value**

A 'data.table' with one spread-curve butterfly value per date.

---

```
calc_spread_curve_slope
```

*Compute Spread-Curve Slope*

---

**Description**

Computes the per-date linear slope of credit spread against tenor.

**Usage**

```
calc_spread_curve_slope(
  DT,
  date_col = "date",
  tenor_col = "tenor",
  spread_col = "credit_spread"
)
```

**Arguments**

|            |                                                                         |
|------------|-------------------------------------------------------------------------|
| DT         | A 'data.table' credit panel containing date, tenor, and spread columns. |
| date_col   | Date or timestamp column name.                                          |
| tenor_col  | Tenor column name.                                                      |
| spread_col | Credit-spread column name.                                              |

**Value**

A 'data.table' with one spread-curve slope estimate per date.

---

|                                           |
|-------------------------------------------|
| calc_StochasticOscillator                 |
| <i>Add Stochastic Oscillator Features</i> |

---

**Description**

Computes fast stochastic '%K' and smoothed '%D' columns from candle highs, lows, and closes in place.

**Usage**

```
calc_StochasticOscillator(DT, ns = c(14), d_ns = c(3))
```

**Arguments**

|      |                                                       |
|------|-------------------------------------------------------|
| DT   | A 'data.table' containing 'high', 'low', and 'close'. |
| ns   | Integer vector of stochastic windows.                 |
| d_ns | Integer vector of '%D' smoothing windows.             |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

calc\_strategy\_performance\_summary

*Standard Backtest Performance Summary*


---

## Description

Computes a broader one-row performance summary from a path-dependent backtest equity curve and, optionally, the strategy target-position path. This is intended for strategy comparison reports and examples where Sortino alone is too narrow.

## Usage

```
calc_strategy_performance_summary(
    equity,
    tgt_pos = NULL,
    annualization = 252,
    risk_free_return = 0,
    min_acceptable_return = 0,
    fee_paid = NULL,
    funding_paid = NULL,
    leverage = NULL,
    recorder = attr(equity, "recorder", exact = TRUE),
    fee_rt = NULL,
    recorder_contract_size = 1,
    turnover_tol = 0
)
```

## Arguments

|                        |                                                                                                                                                                |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| equity                 | Numeric equity curve.                                                                                                                                          |
| tgt_pos                | Optional numeric target-position path aligned with ‘equity’.                                                                                                   |
| annualization          | Numeric periods per year.                                                                                                                                      |
| risk_free_return       | Numeric annual risk-free return used for excess return.                                                                                                        |
| min_acceptable_return  | Numeric annual return threshold used for downside deviation.                                                                                                   |
| fee_paid               | Optional numeric fee-cost vector or scalar.                                                                                                                    |
| funding_paid           | Optional numeric funding-cost vector or scalar.                                                                                                                |
| leverage               | Optional numeric leverage or exposure path aligned with ‘equity’.                                                                                              |
| recorder               | Optional recorder list from ‘attr(backtest_result, "recorder")’. When supplied with ‘fee_rt’, trade fees are estimated from recorded trade quantity and price. |
| fee_rt                 | Optional fee rate used with ‘recorder’ when ‘fee_paid’ is not supplied.                                                                                        |
| recorder_contract_size | Contract-size multiplier used when estimating fees from recorder quantity and price.                                                                           |

turnover\_tol      Numeric tolerance below which target-position changes are ignored for turnover and trade counts.

**Value**

A one-row ‘data.table’ with return, risk, drawdown, exposure, turnover, trade-count, and optional cost metrics.

---

|                                     |
|-------------------------------------|
| calc_term_structure_carry           |
| <i>Compute Term-Structure Carry</i> |

---

**Description**

Computes a simple annualized term-structure carry approximation from the front-versus-deferred futures pair. Positive values indicate positive carry from curve roll-down under backwardation.

**Usage**

```
calc_term_structure_carry(  
  DT,  
  date_col = "date",  
  rank_col = "contract_rank",  
  price_col = "close",  
  time_to_expiry_col = "time_to_expiry",  
  rank_front = 1L,  
  rank_deferred = 2L  
)
```

**Arguments**

DT                      A ‘data.table’ futures panel containing date, contract-rank, price, and time-to-expiry columns.

date\_col                Date or timestamp column name.

rank\_col                Contract-rank column name.

price\_col               Price column name.

time\_to\_expiry\_col      Time-to-expiry column name expressed in years.

rank\_front              Integer front-contract rank.

rank\_deferred          Integer deferred-contract rank.

**Value**

A ‘data.table’ with annualized term-structure carry per date.

---

|                     |                                            |
|---------------------|--------------------------------------------|
| calc_tracking_error | <i>Add Rolling Tracking-Error Features</i> |
|---------------------|--------------------------------------------|

---

**Description**

Computes rolling tracking-error columns from the return difference between an asset and benchmark series.

**Usage**

```
calc_tracking_error(
  DT,
  x_col = "close",
  y_col = "benchmark_close",
  ns = c(20, 60),
  annualization = 252,
  use_log = TRUE
)
```

**Arguments**

|               |                                                       |
|---------------|-------------------------------------------------------|
| DT            | A 'data.table' containing the selected input columns. |
| x_col         | Asset-price column name.                              |
| y_col         | Benchmark-price column name.                          |
| ns            | Integer vector of rolling windows.                    |
| annualization | Numeric annualization factor. Defaults to '252'.      |
| use_log       | Logical; if 'TRUE', uses log returns.                 |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

|           |                          |
|-----------|--------------------------|
| calc_TRIX | <i>Add TRIX Features</i> |
|-----------|--------------------------|

---

**Description**

Computes triple-smoothed exponential trend columns and their signal lines from the 'close' series of a candle 'data.table' in place.

**Usage**

```
calc_TRIX(DT, ns = c(20), signal_ns = c(9), scale = 100)
```

**Arguments**

|           |                                                                                |
|-----------|--------------------------------------------------------------------------------|
| DT        | A 'data.table' containing a 'close' column.                                    |
| ns        | Integer vector of EMA windows used for the triple smoothing.                   |
| signal_ns | Integer vector of signal-line windows.                                         |
| scale     | Numeric scale factor applied to the one-period discrete ROC of the triple EMA. |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

|               |                              |
|---------------|------------------------------|
| calc_turnover | <i>Add Turnover Features</i> |
|---------------|------------------------------|

---

**Description**

Computes traded-value turnover and optional float-normalized turnover ratio in place.

**Usage**

```
calc_turnover(DT, price_col = "close", volume_col = "volume", float_col = NULL)
```

**Arguments**

|            |                                                     |
|------------|-----------------------------------------------------|
| DT         | A 'data.table' containing price and volume columns. |
| price_col  | Price column name.                                  |
| volume_col | Traded-volume column name.                          |
| float_col  | Optional float or shares-outstanding column name.   |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

 calc\_ultimateOscillator

*Add Ultimate Oscillator Features*


---

### Description

Computes Ultimate Oscillator columns from candle highs, lows, and closes in place using weighted buy-pressure to true-range ratios across three windows.

### Usage

```
calc_ultimateOscillator(DT, nss = list(c(7, 14, 28)), wtss = list(c(4, 2, 1)))
```

### Arguments

|      |                                                                                       |
|------|---------------------------------------------------------------------------------------|
| DT   | A 'data.table' containing 'high', 'low', and 'close'.                                 |
| nss  | A list of integer vectors of length three giving the short, medium, and long windows. |
| wtss | A list of numeric vectors of length three giving the corresponding window weights.    |

### Value

The input 'DT', modified by reference and returned invisibly.

---

 calc\_VWAP

*Add Volume-Weighted Average Price Features*


---

### Description

Computes rolling VWAP columns from candle closes and traded volume in place.

### Usage

```
calc_VWAP(DT, ns = c(10))
```

### Arguments

|    |                                                 |
|----|-------------------------------------------------|
| DT | A 'data.table' containing 'close' and 'volume'. |
| ns | Integer vector of VWAP windows.                 |

### Value

The input 'DT', modified by reference and returned invisibly.

---

|           |                                                    |
|-----------|----------------------------------------------------|
| calc_VWMA | <i>Add Volume-Weighted Moving Average Features</i> |
|-----------|----------------------------------------------------|

---

**Description**

Computes rolling VWMA columns from candle closes and traded volume in place.

**Usage**

```
calc_VWMA(DT, ns = c(10))
```

**Arguments**

|    |                                                 |
|----|-------------------------------------------------|
| DT | A 'data.table' containing 'close' and 'volume'. |
| ns | Integer vector of VWMA windows.                 |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

|          |                                             |
|----------|---------------------------------------------|
| calc_WMA | <i>Add Weighted Moving Average Features</i> |
|----------|---------------------------------------------|

---

**Description**

Computes linearly weighted moving-average columns from the 'close' series of a candle 'data.table' in place.

**Usage**

```
calc_WMA(DT, ns = c(5, 8, 9, 10, 12, 20, 21, 26, 30, 50, 100, 200))
```

**Arguments**

|    |                                             |
|----|---------------------------------------------|
| DT | A 'data.table' containing a 'close' column. |
| ns | Integer vector of WMA window sizes.         |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

|          |                                        |
|----------|----------------------------------------|
| calc_WPR | <i>Add Williams Percent R Features</i> |
|----------|----------------------------------------|

---

**Description**

Computes Williams percent R columns from candle highs, lows, and closes in place on the same '[0, 1]' scale used by 'TTR::WPR()'.

**Usage**

```
calc_WPR(DT, ns = c(14))
```

**Arguments**

|    |                                                       |
|----|-------------------------------------------------------|
| DT | A 'data.table' containing 'high', 'low', and 'close'. |
| ns | Integer vector of WPR windows.                        |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

|            |                                                         |
|------------|---------------------------------------------------------|
| calc_ZLEMA | <i>Add Zero-Lag Exponential Moving Average Features</i> |
|------------|---------------------------------------------------------|

---

**Description**

Computes ZLEMA columns from the 'close' series of a candle 'data.table' in place.

**Usage**

```
calc_ZLEMA(DT, ns = c(10, 20, 50))
```

**Arguments**

|    |                                             |
|----|---------------------------------------------|
| DT | A 'data.table' containing a 'close' column. |
| ns | Integer vector of EMA window sizes.         |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

|             |                             |
|-------------|-----------------------------|
| calc_zscore | <i>Add Z-Score Features</i> |
|-------------|-----------------------------|

---

**Description**

Computes rolling z-score columns in place for the selected input columns.

**Usage**

```
calc_zscore(DT, cols = "close", ns = c(20, 60), sample = TRUE)
```

**Arguments**

|        |                                                                         |
|--------|-------------------------------------------------------------------------|
| DT     | A 'data.table' containing the selected input columns.                   |
| cols   | Character vector of numeric columns to standardize.                     |
| ns     | Integer vector of rolling window sizes.                                 |
| sample | Logical; if 'TRUE', uses sample standard deviation. Defaults to 'TRUE'. |

**Value**

The input 'DT', modified by reference and returned invisibly.

---

|                            |                                   |
|----------------------------|-----------------------------------|
| compare_strategy_backtests | <i>Compare Strategy Backtests</i> |
|----------------------------|-----------------------------------|

---

**Description**

Runs multiple 'strat\_\*\_tgt\_pos()' functions over the same market data and execution assumptions, summarizes each path, and ranks the results by a selected metric.

**Usage**

```
compare_strategy_backtests(
  DT,
  strategies,
  from = NULL,
  to = NULL,
  score_col = "sortino",
  keep_paths = FALSE,
  ctr_size = 1,
  ctr_step = 1,
  lev = 10,
  fee_rt = 0,
  fund_rt = 0,
```

```

    tol_pos = 0.1,
    rec = FALSE,
    annualization = 252,
    risk_free_return = 0,
    min_acceptable_return = 0,
    turnover_tol = 0
  )

```

### Arguments

|                                                        |                                                                                                                                                                     |
|--------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DT                                                     | Candle 'data.table' containing 'datetime', 'open', 'high', 'low', and 'close'.                                                                                      |
| strategies                                             | Named list of strategy functions or strategy specs. A spec is a list with 'fun', optional 'params', optional 'strat_id', optional 'asset_id', and optional 'label'. |
| from, to                                               | Optional date filters for the fixed backtesting period.                                                                                                             |
| score_col                                              | Metric used for ranking.                                                                                                                                            |
| keep_paths                                             | Logical; when 'TRUE', include list-columns with target positions and equity curves.                                                                                 |
| ctr_size, ctr_step, lev, fee_rt, fund_rt, tol_pos      | Backtest execution assumptions passed to 'backtest_rcpp()'.                                                                                                         |
| rec                                                    | Logical; when 'TRUE', the backtest engine records execution traces on the equity attribute. Keep this 'FALSE' for larger mining jobs.                               |
| annualization, risk_free_return, min_acceptable_return | Performance metric assumptions passed to 'calc_backtest_performance()'.                                                                                             |
| turnover_tol                                           | Numeric tolerance below which target-position changes are ignored for turnover and trade counts.                                                                    |

### Value

A ranked 'data.table' containing strategy labels, identifiers, and performance summary metrics.

---

filter\_walk\_forward\_results

*Filter Walk-Forward Results for Overfit Risk*

---

### Description

Applies simple out-of-sample robustness gates to walk-forward mining results. This helper is intentionally conservative: it filters on realized test-window summaries rather than on in-sample fit quality.

**Usage**

```
filter_walk_forward_results(
  x,
  score_col = "sortino",
  group_cols = NULL,
  min_windows = 3L,
  min_positive_return_rate = 0.5,
  min_avg_score = -Inf,
  max_avg_score_decay = Inf,
  max_warmup_insufficient_rate = 0,
  min_avg_total_return = -Inf
)
```

**Arguments**

|                              |                                                                                                                       |
|------------------------------|-----------------------------------------------------------------------------------------------------------------------|
| x                            | Result list from 'mine_strategy_walk_forward()' or a walk-forward test-result table.                                  |
| score_col                    | Metric used as the main score column.                                                                                 |
| group_cols                   | Optional parameter columns used to evaluate stability by parameter set. When 'NULL', all rows are evaluated together. |
| min_windows                  | Minimum number of out-of-sample windows required.                                                                     |
| min_positive_return_rate     | Minimum fraction of windows with positive total return.                                                               |
| min_avg_score                | Minimum average out-of-sample score.                                                                                  |
| max_avg_score_decay          | Maximum allowed average train-minus-test score decay.                                                                 |
| max_warmup_insufficient_rate | Maximum fraction of rows with insufficient warmup history.                                                            |
| min_avg_total_return         | Minimum average test-window total return.                                                                             |

**Value**

A ranked 'data.table' of groups that pass the robustness filters, with filter thresholds recorded as columns.

---

gen\_action\_plan\_rcpp    *Generate a Market-Order Action Plan*

---

**Description**

Translates current account state and a target position into an executable market-order action plan.

Usage

```
gen_action_plan_rcpp(  
  ctr_size,  
  ctr_step,  
  lev,  
  last_px,  
  ctr_unit,  
  avg_price,  
  cash,  
  tgt_pos,  
  tol_pos,  
  strat_id,  
  pos_dir = 0L  
)
```

Arguments

|           |                                               |
|-----------|-----------------------------------------------|
| ctr_size  | Contract size.                                |
| ctr_step  | Contract increment step.                      |
| lev       | Leverage.                                     |
| last_px   | Latest market price.                          |
| ctr_unit  | Current contract units held.                  |
| avg_price | Average entry price of the current position.  |
| cash      | Available cash.                               |
| tgt_pos   | Target position in strategy units.            |
| tol_pos   | Position tolerance before action is required. |
| strat_id  | Strategy identifier.                          |
| pos_dir   | Current position direction.                   |

Value

A list describing the action plan required to move toward ‘tgt\_pos’.

---

|                      |                             |
|----------------------|-----------------------------|
| mine_strategy_assets | <i>Mine Strategy Assets</i> |
|----------------------|-----------------------------|

---

Description

Loops over a list of assets for one fixed strategy parameter set and backtesting period, then ranks assets by Sortino ratio by default.

**Usage**

```
mine_strategy_assets(
  market_data_list,
  strategy_fun,
  strategy_params = list(),
  asset_names = NULL,
  from = NULL,
  to = NULL,
  score_col = "sortino",
  keep_paths = FALSE,
  strat_id = 0L,
  asset_id = 0L,
  ctr_size = 1,
  ctr_step = 1,
  lev = 10,
  fee_rt = 0,
  fund_rt = 0,
  tol_pos = 0.1,
  rec = FALSE,
  annualization = 252,
  risk_free_return = 0,
  min_acceptable_return = 0
)
```

**Arguments**

|                                                        |                                                                                                                                       |
|--------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| market_data_list                                       | Named or unnamed list of candle 'data.table's.                                                                                        |
| strategy_fun                                           | Strategy target-position function. It must accept 'DT' as its first argument and return a numeric target-position vector.             |
| strategy_params                                        | Named list of fixed strategy parameters.                                                                                              |
| asset_names                                            | Optional asset labels. Defaults to names of 'market_data_list', or 'asset_1', 'asset_2', ...                                          |
| from, to                                               | Optional date filters for the fixed backtesting period.                                                                               |
| score_col                                              | Metric used for descending ranking.                                                                                                   |
| keep_paths                                             | Logical; when 'TRUE', include list-columns with target positions and equity curves.                                                   |
| strat_id                                               | Integer strategy identifier passed to 'backtest_rcpp()'.                                                                              |
| asset_id                                               | Integer asset identifier passed to 'backtest_rcpp()'.                                                                                 |
| ctr_size, ctr_step, lev, fee_rt, fund_rt, tol_pos      | Backtest execution assumptions passed to 'backtest_rcpp()'.                                                                           |
| rec                                                    | Logical; when 'TRUE', the backtest engine records execution traces on the equity attribute. Keep this 'FALSE' for larger mining jobs. |
| annualization, risk_free_return, min_acceptable_return | Performance metric assumptions passed to 'calc_backtest_performance()'.                                                               |

**Value**

A ranked 'data.table' containing asset labels and performance metrics.

---

mine\_strategy\_asset\_years

*Mine Strategy Parameters Across Asset-Year Pairs*

---

**Description**

First mines the best parameter rows on selected seed assets over the full period. It then evaluates the unique selected parameter rows across every valid asset-year pair, keeps only pairs where strategy total return beats a simple buy-and-hold return, and ranks the survivors by Sortino ratio.

**Usage**

```
mine_strategy_asset_years(
  market_data_list,
  strategy_fun,
  param_grid,
  seed_assets = c("SPY", "AGG", "IAU", "IBIT", "USO", "UUP"),
  seed_n_best = 1L,
  asset_names = NULL,
  years = NULL,
  from = NULL,
  to = NULL,
  min_year_rows = 200L,
  warmup_days = 365L,
  warmup_years = NULL,
  score_col = "sortino",
  keep_paths = FALSE,
  strat_id = 0L,
  asset_id = 0L,
  ctr_size = 1,
  ctr_step = 1,
  lev = 10,
  fee_rt = 0,
  fund_rt = 0,
  tol_pos = 0.1,
  rec = FALSE,
  annualization = 252,
  risk_free_return = 0,
  min_acceptable_return = 0
)
```

**Arguments**

|                                                        |                                                                                                                                          |
|--------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| market_data_list                                       | Named or unnamed list of candle 'data.table's.                                                                                           |
| strategy_fun                                           | Strategy target-position function. It must accept 'DT' as its first argument and return a numeric target-position vector.                |
| param_grid                                             | Non-empty list, 'data.frame', or 'data.table' of parameter values used for seed-asset parameter mining.                                  |
| seed_assets                                            | Character vector of asset names used to select candidate parameter rows.                                                                 |
| seed_n_best                                            | Integer number of top parameter rows retained per seed asset.                                                                            |
| asset_names                                            | Optional asset labels. Defaults to names of 'market_data_list', or 'asset_1', 'asset_2', ...                                             |
| years                                                  | Optional integer vector of calendar years to evaluate. Defaults to all years present after 'from'/'to' filtering.                        |
| from, to                                               | Optional date filters for the fixed backtesting period.                                                                                  |
| min_year_rows                                          | Minimum OHLC rows required for an asset-year pair.                                                                                       |
| warmup_days, warmup_years                              | Warmup history used for signal construction before each asset-year trade window. 'warmup_years', when supplied, overrides 'warmup_days'. |
| score_col                                              | Metric used for descending ranking.                                                                                                      |
| keep_paths                                             | Logical; when 'TRUE', include list-columns with target positions and equity curves.                                                      |
| strat_id                                               | Integer strategy identifier passed to 'backtest_rcpp()'.                                                                                 |
| asset_id                                               | Integer asset identifier passed to 'backtest_rcpp()'.                                                                                    |
| ctr_size, ctr_step, lev, fee_rt, fund_rt, tol_pos      | Backtest execution assumptions passed to 'backtest_rcpp()'.                                                                              |
| rec                                                    | Logical; when 'TRUE', the backtest engine records execution traces on the equity attribute. Keep this 'FALSE' for larger mining jobs.    |
| annualization, risk_free_return, min_acceptable_return | Performance metric assumptions passed to 'calc_backtest_performance()'.                                                                  |

**Details**

Asset-year strategy signals are computed on a wider signal window that can include warmup history before the trade year. The reported performance is always computed on the trade/evaluation asset-year slice only.

**Value**

A list with 'seed\_params', 'candidate\_params', and 'asset\_year\_results' 'data.table's.

---

mine\_strategy\_params    *Mine Strategy Parameters*


---

**Description**

Loops over a strategy parameter grid for one fixed asset and backtesting period, then ranks parameter sets by Sortino ratio by default.

**Usage**

```
mine_strategy_params(
  DT,
  strategy_fun,
  param_grid,
  from = NULL,
  to = NULL,
  score_col = "sortino",
  keep_paths = FALSE,
  strat_id = 0L,
  asset_id = 0L,
  ctr_size = 1,
  ctr_step = 1,
  lev = 10,
  fee_rt = 0,
  fund_rt = 0,
  tol_pos = 0.1,
  rec = FALSE,
  annualization = 252,
  risk_free_return = 0,
  min_acceptable_return = 0
)
```

**Arguments**

|              |                                                                                                                           |
|--------------|---------------------------------------------------------------------------------------------------------------------------|
| DT           | Candle 'data.table' containing 'datetime', 'open', 'high', 'low', and 'close'.                                            |
| strategy_fun | Strategy target-position function. It must accept 'DT' as its first argument and return a numeric target-position vector. |
| param_grid   | Non-empty list, 'data.frame', or 'data.table' of parameter values. Lists are expanded with 'expand.grid()'.               |
| from, to     | Optional date filters for the fixed backtesting period.                                                                   |
| score_col    | Metric used for descending ranking.                                                                                       |
| keep_paths   | Logical; when 'TRUE', include list-columns with target positions and equity curves.                                       |
| strat_id     | Integer strategy identifier passed to 'backtest_rcpp()'.                                                                  |
| asset_id     | Integer asset identifier passed to 'backtest_rcpp()'.                                                                     |

ctr\_size, ctr\_step, lev, fee\_rt, fund\_rt, tol\_pos  
 Backtest execution assumptions passed to 'backtest\_rcpp()'.  
 rec Logical; when 'TRUE', the backtest engine records execution traces on the equity attribute. Keep this 'FALSE' for larger mining jobs.  
 annualization, risk\_free\_return, min\_acceptable\_return  
 Performance metric assumptions passed to 'calc\_backtest\_performance()'.

### Value

A ranked 'data.table' containing parameter values and performance metrics.

---

mine\_strategy\_walk\_forward

*Walk-Forward Strategy Mining*

---

### Description

Performs rolling train/test strategy mining. For each window, parameter rows are ranked on the training slice, the best rows are evaluated on the following test slice, and test signals are computed on a wider window that can include warmup history.

### Usage

```
mine_strategy_walk_forward(
  DT,
  strategy_fun,
  param_grid,
  train_years = 3,
  test_years = 1,
  step_years = 1,
  n_best = 1L,
  from = NULL,
  to = NULL,
  min_train_rows = 200L,
  min_test_rows = 50L,
  warmup_days = 365L,
  warmup_years = NULL,
  score_col = "sortino",
  keep_paths = FALSE,
  strat_id = 0L,
  asset_id = 0L,
  ctr_size = 1,
  ctr_step = 1,
  lev = 10,
  fee_rt = 0,
  fund_rt = 0,
  tol_pos = 0.1,
```

```

    rec = FALSE,
    annualization = 252,
    risk_free_return = 0,
    min_acceptable_return = 0,
    turnover_tol = 0
  )

```

## Arguments

|                                                        |                                                                                                                                       |
|--------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| DT                                                     | Candle 'data.table' containing 'datetime', 'open', 'high', 'low', and 'close'.                                                        |
| strategy_fun                                           | Strategy target-position function. It must accept 'DT' as its first argument and return a numeric target-position vector.             |
| param_grid                                             | Non-empty list, 'data.frame', or 'data.table' of parameter values.                                                                    |
| train_years, test_years                                | Numeric train and test window lengths in calendar years.                                                                              |
| step_years                                             | Numeric step between successive train windows in calendar years.                                                                      |
| n_best                                                 | Integer number of top training parameter rows tested in each out-of-sample window.                                                    |
| from, to                                               | Optional date filters for the fixed backtesting period.                                                                               |
| min_train_rows, min_test_rows                          | Minimum OHLC rows required for each train or test slice.                                                                              |
| warmup_days, warmup_years                              | Warmup history used for test signal construction. 'warmup_years', when supplied, overrides 'warmup_days'.                             |
| score_col                                              | Metric used for descending ranking.                                                                                                   |
| keep_paths                                             | Logical; when 'TRUE', include list-columns with target positions and equity curves.                                                   |
| strat_id                                               | Integer strategy identifier passed to 'backtest_rcpp()'.                                                                              |
| asset_id                                               | Integer asset identifier passed to 'backtest_rcpp()'.                                                                                 |
| ctr_size, ctr_step, lev, fee_rt, fund_rt, tol_pos      | Backtest execution assumptions passed to 'backtest_rcpp()'.                                                                           |
| rec                                                    | Logical; when 'TRUE', the backtest engine records execution traces on the equity attribute. Keep this 'FALSE' for larger mining jobs. |
| annualization, risk_free_return, min_acceptable_return | Performance metric assumptions passed to 'calc_backtest_performance()'.                                                               |
| turnover_tol                                           | Numeric tolerance below which target-position changes are ignored for turnover and trade counts.                                      |

## Value

A list with 'windows', 'train\_results', and ranked 'test\_results' 'data.table's.

---

```
plan_curve_trade_adjustment
```

*Plan Curve Trade Adjustment*

---

**Description**

Computes hedge units that move current key-rate exposure toward target key-rate exposure.

**Usage**

```
plan_curve_trade_adjustment(current_krd, target_krd, hedge_krd)
```

**Arguments**

|             |                                                                       |
|-------------|-----------------------------------------------------------------------|
| current_krd | Named numeric vector of current key-rate exposures.                   |
| target_krd  | Named numeric vector of target key-rate exposures.                    |
| hedge_krd   | Named numeric vector of hedge-instrument key-rate exposures per unit. |

**Value**

A ‘data.table’ with tenor, current, target, gap, hedge exposure, and hedge units.

---

```
plan_delta_neutral_adjustment
```

*Plan Delta-Neutral Adjustment*

---

**Description**

Computes hedge units needed to move current delta toward a target delta.

**Usage**

```
plan_delta_neutral_adjustment(current_delta, target_delta = 0, hedge_delta)
```

**Arguments**

|               |                                                  |
|---------------|--------------------------------------------------|
| current_delta | Numeric current portfolio delta.                 |
| target_delta  | Numeric target portfolio delta. Defaults to ‘0’. |
| hedge_delta   | Numeric delta contribution per hedge unit.       |

**Value**

A one-row ‘data.table’ with current, target, gap, hedge exposure, hedge units, absolute hedge units, and hedge action.

---

plan\_duration\_neutral\_adjustment

*Plan Duration-Neutral Adjustment*


---

### Description

Computes the hedge units needed to move current DV01 toward a target DV01.

### Usage

```
plan_duration_neutral_adjustment(current_dv01, target_dv01 = 0, hedge_dv01)
```

### Arguments

|              |                                                 |
|--------------|-------------------------------------------------|
| current_dv01 | Numeric current portfolio DV01.                 |
| target_dv01  | Numeric target portfolio DV01. Defaults to '0'. |
| hedge_dv01   | Numeric DV01 contribution per hedge unit.       |

### Value

A one-row 'data.table' with current, target, gap, and hedge units.

---

plan\_portfolio\_adjustment

*Plan Portfolio Adjustments*


---

### Description

Converts target portfolio weights into per-asset rebalancing deltas using the current portfolio state. This is a minimal multi-asset bridge between quantitative allocation outputs and executable portfolio adjustments.

### Usage

```
plan_portfolio_adjustment(
  portfolio_state,
  equity,
  contract_size_col = "contract_size",
  lot_step_col = "lot_step"
)
```

**Arguments**

|                   |                                                                                            |
|-------------------|--------------------------------------------------------------------------------------------|
| portfolio_state   | A 'data.table' containing at least 'asset', 'price', 'current_units', and 'target_weight'. |
| equity            | Total portfolio equity used to scale target weights into target notionals.                 |
| contract_size_col | Column name containing contract-size multipliers.                                          |
| lot_step_col      | Column name containing minimum trade increments.                                           |

**Value**

A 'data.table' with current, target, and delta notional/unit columns.

---

|                             |                                    |
|-----------------------------|------------------------------------|
| plan_vega_target_adjustment | <i>Plan Vega Target Adjustment</i> |
|-----------------------------|------------------------------------|

---

**Description**

Computes hedge units needed to move current vega toward a target vega.

**Usage**

```
plan_vega_target_adjustment(current_vega, target_vega = 0, hedge_vega)
```

**Arguments**

|              |                                                 |
|--------------|-------------------------------------------------|
| current_vega | Numeric current portfolio vega.                 |
| target_vega  | Numeric target portfolio vega. Defaults to '0'. |
| hedge_vega   | Numeric vega contribution per hedge unit.       |

**Value**

A one-row 'data.table' with current, target, gap, hedge exposure, hedge units, absolute hedge units, and hedge action.

---

rank\_strategy\_results *Rank Strategy Results*


---

### Description

Orders a strategy result table by a selected score column and adds a 'rank' column. This helper is shared by comparison workflows and can also be used on custom result tables.

### Usage

```
rank_strategy_results(result_dt, score_col = "sortino", decreasing = TRUE)
```

### Arguments

|            |                                                             |
|------------|-------------------------------------------------------------|
| result_dt  | A 'data.table' or 'data.frame' containing strategy metrics. |
| score_col  | Column used for ranking.                                    |
| decreasing | Logical; when 'TRUE', larger scores rank better.            |

### Value

A ranked 'data.table'.

---

select\_strategy\_params *Select Strategy Parameter Candidates*


---

### Description

Extracts reusable parameter rows from a ranked mining result. This is useful after parameter, asset-year, or walk-forward mining when the next step is to rerun a strategy with the best discovered settings.

### Usage

```
select_strategy_params(
  x,
  n = 1L,
  param_cols = NULL,
  score_col = "sortino",
  decreasing = TRUE,
  unique_params = TRUE,
  include_metrics = FALSE,
  table = c("auto", "seed_params", "candidate_params", "asset_year_results",
    "train_results", "test_results"),
  rerank = FALSE
)
```

**Arguments**

|                 |                                                                                                                             |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------|
| x               | A mining result 'data.table', or a result list returned by 'mine_strategy_asset_years()' or 'mine_strategy_walk_forward()'. |
| n               | Integer number of rows to keep.                                                                                             |
| param_cols      | Optional character vector of parameter columns. When omitted, known metric and metadata columns are excluded.               |
| score_col       | Metric used for ranking when 'rank' is absent or 'rerank = TRUE'.                                                           |
| decreasing      | Logical; when 'TRUE', larger scores rank better.                                                                            |
| unique_params   | Logical; when 'TRUE', duplicate parameter rows are removed after ranking.                                                   |
| include_metrics | Logical; when 'TRUE', include score and selected diagnostics beside the parameter columns.                                  |
| table           | Which table to use when 'x' is a mining result list. "auto" prefers out-of-sample results.                                  |
| rerank          | Logical; when 'TRUE', rank by 'score_col' before selecting.                                                                 |

**Value**

A 'data.table' containing selected parameter rows.

---

strategy\_monitor\_definition

*Strategy Monitor Definition*


---

**Description**

Returns public-safe monitor metadata for a supported Vox strategy id. This metadata is separate from execution and should be used by monitoring consumers instead of inferring family or regime labels from names.

**Usage**

```
strategy_monitor_definition(strategy_id)
```

**Arguments**

|             |                                                                                                                                                                                                 |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| strategy_id | Character scalar strategy id. Supported ids are "buy_hold", "ema_cross_adx", "ema_cross_slope_confirm", "rsi_revert", "vol_target", "donchian_turtle", "bollinger_revert", and "regime_switch". |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

**Value**

A named list with schema version, strategy id, strategy family, expected regimes, and public regime interpretation.

---

strategy\_public\_definition

*Public Strategy Definition*


---

### Description

Returns the canonical public-safe deterministic strategy definition used by Vox for a supported strategy id.

### Usage

```
strategy_public_definition(id)
```

### Arguments

|    |                                                                                                                                                                                                              |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| id | Character scalar strategy id. Supported ids are "buy_hold", "ema_cross", "ema_cross_adx", "ema_cross_slope_confirm", "donchian_turtle", "bollinger_revert", "rsi_revert", "vol_target", and "regime_switch". |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

### Value

A named list with schema version, public description, target function name, data requirements, re-balance rule, and effective default strategy parameters. Each parameter has 'name', 'value', 'unit', and 'description'; 'value' is a finite numeric scalar or 'NULL'. A 'NULL' value explicitly represents an unbounded setting that is disabled by default in the public JSON-safe contract.

---

strat\_atr\_breakout\_action\_plan

*ATR-Breakout Action Plan*


---

### Description

Applies the ATR breakout rule to the latest bar and translates the resulting target exposure into an executable action plan.

### Usage

```
strat_atr_breakout_action_plan(
  DT,
  state,
  n = 14L,
  atr_mult = 1,
  target_size = 1,
  compute_features = TRUE,
  strat_id = 401L,
  tol_pos = 0.1,
  debug = FALSE
)
```

**Arguments**

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle 'data.table'.                                                                          |
| state            | Named list describing the current trading state.                                                |
| n                | Integer ATR window.                                                                             |
| atr_mult         | Numeric ATR multiple used for the breakout threshold.                                           |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing ATR features are added to 'DT' in place.                          |
| strat_id         | Integer strategy identifier recorded on generated actions.                                      |
| tol_pos          | Numeric tolerance passed to the action planner.                                                 |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A list produced by 'gen\_action\_plan\_rcpp()'.

---

strat\_atr\_breakout\_tgt\_pos  
*ATR-Breakout Target Positions*

---

**Description**

Generates a simple breakout target-position path from close-to-close moves measured against prior ATR. Upside moves larger than 'atr\_mult \* ATR' target a long exposure, while downside moves larger than the same threshold target a short exposure.

**Usage**

```
strat_atr_breakout_tgt_pos(
  DT,
  n = 14L,
  atr_mult = 1,
  target_size = 1,
  compute_features = TRUE,
  debug = FALSE
)
```

**Arguments**

|          |                                                       |
|----------|-------------------------------------------------------|
| DT       | A candle 'data.table'.                                |
| n        | Integer ATR window.                                   |
| atr_mult | Numeric ATR multiple used for the breakout threshold. |

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing ATR features are added to 'DT' in place.                          |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

strat\_atr\_breakout\_trailing\_stop\_action\_plan  
*ATR-Breakout-Trailing-Stop Action Plan*

---

**Description**

Applies the ATR breakout with trailing-stop rule to the latest bar and translates the resulting target exposure into an executable action plan.

**Usage**

```
strat_atr_breakout_trailing_stop_action_plan(
  DT,
  state,
  n = 14L,
  atr_mult = 1,
  trail_mult = 2,
  target_size = 1,
  compute_features = TRUE,
  strat_id = 405L,
  tol_pos = 0.1,
  debug = FALSE
)
```

**Arguments**

|                  |                                                                        |
|------------------|------------------------------------------------------------------------|
| DT               | A candle 'data.table'.                                                 |
| state            | Named list describing the current trading state.                       |
| n                | Integer ATR window.                                                    |
| atr_mult         | Numeric ATR multiple used for the breakout threshold.                  |
| trail_mult       | Numeric ATR multiple used for the trailing stop.                       |
| target_size      | Numeric absolute target exposure.                                      |
| compute_features | Logical; when 'TRUE', missing ATR features are added to 'DT' in place. |
| strat_id         | Integer strategy identifier recorded on generated actions.             |

|         |                                                                                                 |
|---------|-------------------------------------------------------------------------------------------------|
| tol_pos | Numeric tolerance passed to the action planner.                                                 |
| debug   | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A list produced by 'gen\_action\_plan\_rcpp()'.

---

strat\_atr\_breakout\_trailing\_stop\_tgt\_pos  
*ATR-Breakout-Trailing-Stop Target Positions*

---

**Description**

Generates a breakout target-position path from ATR moves with an ATR-based trailing stop. Entries use the current ATR breakout rule, and exits trail the most favorable close by a multiple of current ATR.

**Usage**

```
strat_atr_breakout_trailing_stop_tgt_pos(
  DT,
  n = 14L,
  atr_mult = 1,
  trail_mult = 2,
  target_size = 1,
  compute_features = TRUE,
  debug = FALSE
)
```

**Arguments**

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle 'data.table'.                                                                          |
| n                | Integer ATR window.                                                                             |
| atr_mult         | Numeric ATR multiple used for the breakout threshold.                                           |
| trail_mult       | Numeric ATR multiple used for the trailing stop.                                                |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing ATR features are added to 'DT' in place.                          |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

strat\_bollinger\_low\_adx\_revert\_action\_plan

*Bollinger-Low-ADX-Reversion Action Plan*


---

### Description

Applies the low-ADX Bollinger reversion rule to the latest bar and translates the resulting target exposure into an executable action plan.

### Usage

```
strat_bollinger_low_adx_revert_action_plan(
  DT,
  state,
  n = 20L,
  k = 2,
  adx_n = 14L,
  adx_max = 18,
  target_size = 1,
  compute_features = TRUE,
  strat_id = 312L,
  tol_pos = 0.1,
  debug = FALSE
)
```

### Arguments

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle 'data.table'.                                                                          |
| state            | Named list describing the current trading state.                                                |
| n                | Integer Bollinger window.                                                                       |
| k                | Numeric Bollinger width multiplier.                                                             |
| adx_n            | Integer ADX window.                                                                             |
| adx_max          | Numeric maximum ADX allowed for the reversion regime.                                           |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing Bollinger and ADX features are added to 'DT' in place.            |
| strat_id         | Integer strategy identifier recorded on generated actions.                                      |
| tol_pos          | Numeric tolerance passed to the action planner.                                                 |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

### Value

A list produced by 'gen\_action\_plan\_rcpp()'.

---

strat\_bollinger\_low\_adx\_revert\_tgt\_pos

*Bollinger-Low-ADX-Reversion Target Positions*


---

## Description

Generates a Bollinger-band mean-reversion target-position path that is only active in low-ADX regimes. Open reversion positions are also flattened if ADX rises above the permitted regime threshold.

## Usage

```
strat_bollinger_low_adx_revert_tgt_pos(
  DT,
  n = 20L,
  k = 2,
  adx_n = 14L,
  adx_max = 18,
  target_size = 1,
  compute_features = TRUE,
  debug = FALSE
)
```

## Arguments

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle 'data.table'.                                                                          |
| n                | Integer Bollinger window.                                                                       |
| k                | Numeric Bollinger width multiplier.                                                             |
| adx_n            | Integer ADX window.                                                                             |
| adx_max          | Numeric maximum ADX allowed for the reversion regime.                                           |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing Bollinger and ADX features are added to 'DT' in place.            |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

## Value

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

strat\_bollinger\_revert\_action\_plan

*Bollinger-Reversion Action Plan*


---

## Description

Applies the Bollinger-band mean-reversion rule to the latest bar and translates the resulting target exposure into an executable action plan.

## Usage

```
strat_bollinger_revert_action_plan(
  DT,
  state,
  n = 20L,
  k = 2,
  target_size = 1,
  compute_features = TRUE,
  strat_id = 301L,
  tol_pos = 0.1,
  debug = FALSE
)
```

## Arguments

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle ‘data.table’.                                                                          |
| state            | Named list describing the current trading state.                                                |
| n                | Integer Bollinger window.                                                                       |
| k                | Numeric Bollinger width multiplier.                                                             |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when ‘TRUE’, missing Bollinger features are added to ‘DT’ in place.                    |
| strat_id         | Integer strategy identifier recorded on generated actions.                                      |
| tol_pos          | Numeric tolerance passed to the action planner.                                                 |
| debug            | Logical; when ‘TRUE’, returns a list with the generated target vector and feature column names. |

## Value

A list produced by ‘gen\_action\_plan\_rcpp()’.

---

strat\_bollinger\_revert\_rsi\_action\_plan

*Bollinger-Reversion-RSI Action Plan*


---

## Description

Applies the Bollinger-band-plus-RSI mean-reversion rule to the latest bar and translates the resulting target exposure into an executable action plan.

## Usage

```
strat_bollinger_revert_rsi_action_plan(
  DT,
  state,
  bb_n = 20L,
  bb_k = 2,
  rsi_n = 14L,
  oversold = 30,
  overbought = 70,
  exit_level = 50,
  target_size = 1,
  compute_features = TRUE,
  strat_id = 308L,
  tol_pos = 0.1,
  debug = FALSE
)
```

## Arguments

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle 'data.table'.                                                                          |
| state            | Named list describing the current trading state.                                                |
| bb_n             | Integer Bollinger window.                                                                       |
| bb_k             | Numeric Bollinger width multiplier.                                                             |
| rsi_n            | Integer RSI window.                                                                             |
| oversold         | Numeric RSI threshold used to confirm long entries.                                             |
| overbought       | Numeric RSI threshold used to confirm short entries.                                            |
| exit_level       | Numeric RSI level used as a secondary exit trigger.                                             |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing Bollinger and RSI features are added to 'DT' in place.            |
| strat_id         | Integer strategy identifier recorded on generated actions.                                      |
| tol_pos          | Numeric tolerance passed to the action planner.                                                 |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A list produced by ‘gen\_action\_plan\_rcpp()’.

---

|                                                 |
|-------------------------------------------------|
| strat_bollinger_revert_rsi_tgt_pos              |
| <i>Bollinger-Reversion-RSI Target Positions</i> |

---

**Description**

Generates a Bollinger-band mean-reversion target-position path that requires RSI confirmation before opening long or short reversion trades.

**Usage**

```
strat_bollinger_revert_rsi_tgt_pos(  
  DT,  
  bb_n = 20L,  
  bb_k = 2,  
  rsi_n = 14L,  
  oversold = 30,  
  overbought = 70,  
  exit_level = 50,  
  target_size = 1,  
  compute_features = TRUE,  
  debug = FALSE  
)
```

**Arguments**

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle ‘data.table’.                                                                          |
| bb_n             | Integer Bollinger window.                                                                       |
| bb_k             | Numeric Bollinger width multiplier.                                                             |
| rsi_n            | Integer RSI window.                                                                             |
| oversold         | Numeric RSI threshold used to confirm long entries.                                             |
| overbought       | Numeric RSI threshold used to confirm short entries.                                            |
| exit_level       | Numeric RSI level used as a secondary exit trigger.                                             |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when ‘TRUE’, missing Bollinger and RSI features are added to ‘DT’ in place.            |
| debug            | Logical; when ‘TRUE’, returns a list with the generated target vector and feature column names. |

**Value**

A numeric vector of target positions, or a list when ‘debug = TRUE’.

---

strat\_bollinger\_revert\_tgt\_pos

*Bollinger-Reversion Target Positions*


---

## Description

Generates a simple mean-reversion target-position path from Bollinger band touches. Touching the lower band opens a long target, touching the upper band opens a short target, and open targets are closed when price returns to the mid band.

## Usage

```
strat_bollinger_revert_tgt_pos(
  DT,
  n = 20L,
  k = 2,
  target_size = 1,
  compute_features = TRUE,
  debug = FALSE
)
```

## Arguments

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle 'data.table'.                                                                          |
| n                | Integer Bollinger window.                                                                       |
| k                | Numeric Bollinger width multiplier.                                                             |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing Bollinger features are added to 'DT' in place.                    |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

## Value

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

strat\_bollinger\_squeeze\_breakout\_action\_plan

*Bollinger-Squeeze-Breakout Action Plan*


---

## Description

Applies the squeeze-breakout rule to the latest bar and translates the resulting target exposure into an executable action plan.

## Usage

```
strat_bollinger_squeeze_breakout_action_plan(
  DT,
  state,
  bb_n = 20L,
  bb_k = 2,
  squeeze_width = 0.05,
  target_size = 1,
  compute_features = TRUE,
  strat_id = 309L,
  tol_pos = 0.1,
  debug = FALSE
)
```

## Arguments

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle 'data.table'.                                                                          |
| state            | Named list describing the current trading state.                                                |
| bb_n             | Integer Bollinger window.                                                                       |
| bb_k             | Numeric Bollinger width multiplier.                                                             |
| squeeze_width    | Numeric relative Bollinger-width threshold used to arm the breakout.                            |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing Bollinger features are added to 'DT' in place.                    |
| strat_id         | Integer strategy identifier recorded on generated actions.                                      |
| tol_pos          | Numeric tolerance passed to the action planner.                                                 |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

## Value

A list produced by 'gen\_action\_plan\_rcpp()'.

---

strat\_bollinger\_squeeze\_breakout\_tgt\_pos

*Bollinger-Squeeze-Breakout Target Positions*


---

## Description

Generates a volatility-compression breakout target-position path. The strategy arms when Bollinger-band width falls below a squeeze threshold, then enters in the breakout direction when price breaches the outer band during width expansion. Open positions are closed on a return to the mid band.

## Usage

```
strat_bollinger_squeeze_breakout_tgt_pos(
  DT,
  bb_n = 20L,
  bb_k = 2,
  squeeze_width = 0.05,
  target_size = 1,
  compute_features = TRUE,
  debug = FALSE
)
```

## Arguments

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle ‘data.table’.                                                                          |
| bb_n             | Integer Bollinger window.                                                                       |
| bb_k             | Numeric Bollinger width multiplier.                                                             |
| squeeze_width    | Numeric relative Bollinger-width threshold used to arm the breakout.                            |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when ‘TRUE’, missing Bollinger features are added to ‘DT’ in place.                    |
| debug            | Logical; when ‘TRUE’, returns a list with the generated target vector and feature column names. |

## Value

A numeric vector of target positions, or a list when ‘debug = TRUE’.

---

strat\_bond\_carry\_roll\_action\_plan

*Bond-Carry-and-Roll Action Plan*


---

### Description

Applies the bond carry-and-roll rule to the latest row and translates the resulting target exposure into an executable action plan.

### Usage

```
strat_bond_carry_roll_action_plan(
  DT,
  state,
  par_col = "par",
  c_rate_col = "c_rate",
  maturity_col = "maturity",
  freq_col = "freq",
  ytm_col = "ytm",
  accrual_frac_col = NULL,
  holding_years_col = NULL,
  funding_rate_col = NULL,
  long_threshold = 0,
  short_threshold = 0,
  target_size = 1,
  compute_features = TRUE,
  strat_id = 602L,
  tol_pos = 0.1,
  debug = FALSE
)
```

### Arguments

|                   |                                                                                                                       |
|-------------------|-----------------------------------------------------------------------------------------------------------------------|
| DT                | A 'data.table' containing bond carry-and-roll inputs or precomputed 'bond_carry' and 'bond_roll_down_return' columns. |
| state             | Named list describing the current trading state.                                                                      |
| par_col           | Face-value column name.                                                                                               |
| c_rate_col        | Coupon-rate column name.                                                                                              |
| maturity_col      | Maturity column name in years.                                                                                        |
| freq_col          | Coupon-frequency column name.                                                                                         |
| ytm_col           | Yield-to-maturity column name.                                                                                        |
| accrual_frac_col  | Optional accrual-fraction column name.                                                                                |
| holding_years_col | Optional holding-horizon column name in years.                                                                        |

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| funding_rate_col | Optional financing-rate column name.                                                            |
| long_threshold   | Numeric threshold above which carry-and-roll is treated as positive enough to go long.          |
| short_threshold  | Numeric threshold below which carry-and-roll is treated as negative enough to go short.         |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing carry and roll-down features are added to 'DT' in place.          |
| strat_id         | Integer strategy identifier recorded on generated actions.                                      |
| tol_pos          | Numeric tolerance passed to the action planner.                                                 |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A list produced by 'gen\_action\_plan\_rcpp()'.

---

strat\_bond\_carry\_roll\_duration\_cap\_action\_plan  
*Bond-Carry-and-Roll-Duration-Cap Action Plan*

---

**Description**

Applies the duration-capped bond carry-and-roll rule to the latest row and translates the resulting target exposure into an executable action plan.

**Usage**

```
strat_bond_carry_roll_duration_cap_action_plan(
  DT,
  state,
  par_col = "par",
  c_rate_col = "c_rate",
  maturity_col = "maturity",
  freq_col = "freq",
  ytm_col = "ytm",
  duration_col = NULL,
  duration_max = NULL,
  duration_min = NULL,
  accrual_frac_col = NULL,
  holding_years_col = NULL,
  funding_rate_col = NULL,
```

```

    long_threshold = 0,
    short_threshold = 0,
    target_size = 1,
    compute_features = TRUE,
    strat_id = 613L,
    tol_pos = 0.1,
    debug = FALSE
  )

```

### Arguments

|                   |                                                                                                                       |
|-------------------|-----------------------------------------------------------------------------------------------------------------------|
| DT                | A 'data.table' containing bond carry-and-roll inputs or precomputed 'bond_carry' and 'bond_roll_down_return' columns. |
| state             | Named list describing the current trading state.                                                                      |
| par_col           | Face-value column name.                                                                                               |
| c_rate_col        | Coupon-rate column name.                                                                                              |
| maturity_col      | Maturity column name in years.                                                                                        |
| freq_col          | Coupon-frequency column name.                                                                                         |
| ytm_col           | Yield-to-maturity column name.                                                                                        |
| duration_col      | Optional precomputed duration column name. Defaults to a computed modified-duration column.                           |
| duration_max      | Optional maximum duration allowed for active positions.                                                               |
| duration_min      | Optional minimum duration allowed for active positions.                                                               |
| accrual_frac_col  | Optional accrual-fraction column name.                                                                                |
| holding_years_col | Optional holding-horizon column name in years.                                                                        |
| funding_rate_col  | Optional financing-rate column name.                                                                                  |
| long_threshold    | Numeric threshold above which carry-and-roll is treated as positive enough to go long.                                |
| short_threshold   | Numeric threshold below which carry-and-roll is treated as negative enough to go short.                               |
| target_size       | Numeric absolute target exposure.                                                                                     |
| compute_features  | Logical; when 'TRUE', missing carry, roll-down, and duration features are added to 'DT' in place.                     |
| strat_id          | Integer strategy identifier recorded on generated actions.                                                            |
| tol_pos           | Numeric tolerance passed to the action planner.                                                                       |
| debug             | Logical; when 'TRUE', returns a list with the generated target vector and feature column names.                       |

### Value

A list produced by 'gen\_action\_plan\_rcpp()'.

---

|                                                          |
|----------------------------------------------------------|
| strat_bond_carry_roll_duration_cap_tgt_pos               |
| <i>Bond-Carry-and-Roll-Duration-Cap Target Positions</i> |

---

**Description**

Generates a bond carry-and-roll target-position path with an optional duration bucket or cap. The base signal is carry plus roll-down return, but positions are only allowed when modified duration remains within the requested range.

**Usage**

```
strat_bond_carry_roll_duration_cap_tgt_pos(  
  DT,  
  par_col = "par",  
  c_rate_col = "c_rate",  
  maturity_col = "maturity",  
  freq_col = "freq",  
  ytm_col = "ytm",  
  duration_col = NULL,  
  duration_max = NULL,  
  duration_min = NULL,  
  accrual_frac_col = NULL,  
  holding_years_col = NULL,  
  funding_rate_col = NULL,  
  long_threshold = 0,  
  short_threshold = 0,  
  target_size = 1,  
  compute_features = TRUE,  
  debug = FALSE  
)
```

**Arguments**

|              |                                                                                                                       |
|--------------|-----------------------------------------------------------------------------------------------------------------------|
| DT           | A 'data.table' containing bond carry-and-roll inputs or precomputed 'bond_carry' and 'bond_roll_down_return' columns. |
| par_col      | Face-value column name.                                                                                               |
| c_rate_col   | Coupon-rate column name.                                                                                              |
| maturity_col | Maturity column name in years.                                                                                        |
| freq_col     | Coupon-frequency column name.                                                                                         |
| ytm_col      | Yield-to-maturity column name.                                                                                        |
| duration_col | Optional precomputed duration column name. Defaults to a computed modified-duration column.                           |
| duration_max | Optional maximum duration allowed for active positions.                                                               |

|                   |                                                                                                   |
|-------------------|---------------------------------------------------------------------------------------------------|
| duration_min      | Optional minimum duration allowed for active positions.                                           |
| accrual_frac_col  | Optional accrual-fraction column name.                                                            |
| holding_years_col | Optional holding-horizon column name in years.                                                    |
| funding_rate_col  | Optional financing-rate column name.                                                              |
| long_threshold    | Numeric threshold above which carry-and-roll is treated as positive enough to go long.            |
| short_threshold   | Numeric threshold below which carry-and-roll is treated as negative enough to go short.           |
| target_size       | Numeric absolute target exposure.                                                                 |
| compute_features  | Logical; when 'TRUE', missing carry, roll-down, and duration features are added to 'DT' in place. |
| debug             | Logical; when 'TRUE', returns a list with the generated target vector and feature column names.   |

**Value**

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

strat\_bond\_carry\_roll\_spread\_filter\_action\_plan  
*Bond-Carry-and-Roll-Spread-Filter Action Plan*

---

**Description**

Applies the bond carry-and-roll rule with optional spread filters to the latest row and translates the resulting target exposure into an executable action plan.

**Usage**

```
strat_bond_carry_roll_spread_filter_action_plan(
  DT,
  state,
  par_col = "par",
  c_rate_col = "c_rate",
  maturity_col = "maturity",
  freq_col = "freq",
  ytm_col = "ytm",
  spread_col = "credit_spread",
  accrual_frac_col = NULL,
  holding_years_col = NULL,
```

```

    funding_rate_col = NULL,
    long_threshold = 0,
    short_threshold = 0,
    min_long_spread = NULL,
    max_short_spread = NULL,
    target_size = 1,
    compute_features = TRUE,
    strat_id = 606L,
    tol_pos = 0.1,
    debug = FALSE
)

```

### Arguments

|                   |                                                                                                                       |
|-------------------|-----------------------------------------------------------------------------------------------------------------------|
| DT                | A 'data.table' containing bond carry-and-roll inputs or precomputed 'bond_carry' and 'bond_roll_down_return' columns. |
| state             | Named list describing the current trading state.                                                                      |
| par_col           | Face-value column name.                                                                                               |
| c_rate_col        | Coupon-rate column name.                                                                                              |
| maturity_col      | Maturity column name in years.                                                                                        |
| freq_col          | Coupon-frequency column name.                                                                                         |
| ytm_col           | Yield-to-maturity column name.                                                                                        |
| spread_col        | Credit-spread column name used for optional trade filters.                                                            |
| accrual_frac_col  | Optional accrual-fraction column name.                                                                                |
| holding_years_col | Optional holding-horizon column name in years.                                                                        |
| funding_rate_col  | Optional financing-rate column name.                                                                                  |
| long_threshold    | Numeric threshold above which carry-and-roll is treated as positive enough to go long.                                |
| short_threshold   | Numeric threshold below which carry-and-roll is treated as negative enough to go short.                               |
| min_long_spread   | Optional minimum spread required to allow long exposure.                                                              |
| max_short_spread  | Optional maximum spread allowed to permit short exposure.                                                             |
| target_size       | Numeric absolute target exposure.                                                                                     |
| compute_features  | Logical; when 'TRUE', missing carry and roll-down features are added to 'DT' in place.                                |
| strat_id          | Integer strategy identifier recorded on generated actions.                                                            |
| tol_pos           | Numeric tolerance passed to the action planner.                                                                       |
| debug             | Logical; when 'TRUE', returns a list with the generated target vector and feature column names.                       |

**Value**

A list produced by ‘gen\_action\_plan\_rcpp()’.

---

|                                                           |
|-----------------------------------------------------------|
| strat_bond_carry_roll_spread_filter_tgt_pos               |
| <i>Bond-Carry-and-Roll-Spread-Filter Target Positions</i> |

---

**Description**

Generates a bond carry-and-roll target-position path that applies optional spread filters before allowing long or short exposure.

**Usage**

```
strat_bond_carry_roll_spread_filter_tgt_pos(  
  DT,  
  par_col = "par",  
  c_rate_col = "c_rate",  
  maturity_col = "maturity",  
  freq_col = "freq",  
  ytm_col = "ytm",  
  spread_col = "credit_spread",  
  accrual_frac_col = NULL,  
  holding_years_col = NULL,  
  funding_rate_col = NULL,  
  long_threshold = 0,  
  short_threshold = 0,  
  min_long_spread = NULL,  
  max_short_spread = NULL,  
  target_size = 1,  
  compute_features = TRUE,  
  debug = FALSE  
)
```

**Arguments**

|              |                                                                                                                       |
|--------------|-----------------------------------------------------------------------------------------------------------------------|
| DT           | A ‘data.table’ containing bond carry-and-roll inputs or precomputed ‘bond_carry’ and ‘bond_roll_down_return’ columns. |
| par_col      | Face-value column name.                                                                                               |
| c_rate_col   | Coupon-rate column name.                                                                                              |
| maturity_col | Maturity column name in years.                                                                                        |
| freq_col     | Coupon-frequency column name.                                                                                         |
| ytm_col      | Yield-to-maturity column name.                                                                                        |
| spread_col   | Credit-spread column name used for optional trade filters.                                                            |

|                   |                                                                                                 |
|-------------------|-------------------------------------------------------------------------------------------------|
| accrual_frac_col  | Optional accrual-fraction column name.                                                          |
| holding_years_col | Optional holding-horizon column name in years.                                                  |
| funding_rate_col  | Optional financing-rate column name.                                                            |
| long_threshold    | Numeric threshold above which carry-and-roll is treated as positive enough to go long.          |
| short_threshold   | Numeric threshold below which carry-and-roll is treated as negative enough to go short.         |
| min_long_spread   | Optional minimum spread required to allow long exposure.                                        |
| max_short_spread  | Optional maximum spread allowed to permit short exposure.                                       |
| target_size       | Numeric absolute target exposure.                                                               |
| compute_features  | Logical; when 'TRUE', missing carry and roll-down features are added to 'DT' in place.          |
| debug             | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

|                               |                                             |
|-------------------------------|---------------------------------------------|
| strat_bond_carry_roll_tgt_pos | <i>Bond-Carry-and-Roll Target Positions</i> |
|-------------------------------|---------------------------------------------|

---

**Description**

Generates a simple bond target-position path from carry plus roll-down return. Positive carry-and-roll targets a long exposure and negative carry-and-roll targets a short exposure.

**Usage**

```
strat_bond_carry_roll_tgt_pos(
  DT,
  par_col = "par",
  c_rate_col = "c_rate",
  maturity_col = "maturity",
  freq_col = "freq",
  ytm_col = "ytm",
  accrual_frac_col = NULL,
```

```

    holding_years_col = NULL,
    funding_rate_col = NULL,
    long_threshold = 0,
    short_threshold = 0,
    target_size = 1,
    compute_features = TRUE,
    debug = FALSE
  )

```

### Arguments

|                   |                                                                                                                       |
|-------------------|-----------------------------------------------------------------------------------------------------------------------|
| DT                | A 'data.table' containing bond carry-and-roll inputs or precomputed 'bond_carry' and 'bond_roll_down_return' columns. |
| par_col           | Face-value column name.                                                                                               |
| c_rate_col        | Coupon-rate column name.                                                                                              |
| maturity_col      | Maturity column name in years.                                                                                        |
| freq_col          | Coupon-frequency column name.                                                                                         |
| ytm_col           | Yield-to-maturity column name.                                                                                        |
| accrual_frac_col  | Optional accrual-fraction column name.                                                                                |
| holding_years_col | Optional holding-horizon column name in years.                                                                        |
| funding_rate_col  | Optional financing-rate column name.                                                                                  |
| long_threshold    | Numeric threshold above which carry-and-roll is treated as positive enough to go long.                                |
| short_threshold   | Numeric threshold below which carry-and-roll is treated as negative enough to go short.                               |
| target_size       | Numeric absolute target exposure.                                                                                     |
| compute_features  | Logical; when 'TRUE', missing carry and roll-down features are added to 'DT' in place.                                |
| debug             | Logical; when 'TRUE', returns a list with the generated target vector and feature column names.                       |

### Value

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

```
strat_buy_and_hold_action_plan
```

*Buy-And-Hold Action Plan*

---

**Description**

Applies the buy-and-hold target rule to the latest bar and translates the resulting target exposure into an executable action plan.

**Usage**

```
strat_buy_and_hold_action_plan(
  DT,
  state,
  value = 1,
  strat_id = 1L,
  tol_pos = 0
)
```

**Arguments**

|          |                                                            |
|----------|------------------------------------------------------------|
| DT       | A candle ‘data.table’ containing ‘datetime’.               |
| state    | Named list describing the current trading state.           |
| value    | Numeric scalar target exposure.                            |
| strat_id | Integer strategy identifier recorded on generated actions. |
| tol_pos  | Numeric tolerance passed to the action planner.            |

**Value**

A list produced by ‘gen\_action\_plan\_rcpp()’.

---

```
strat_buy_and_hold_tgt_pos
```

*Buy-And-Hold Target Positions*

---

**Description**

Generates a constant target exposure path for a simple buy-and-hold strategy.

**Usage**

```
strat_buy_and_hold_tgt_pos(DT, value = 1)
```

**Arguments**

|       |                                              |
|-------|----------------------------------------------|
| DT    | A candle 'data.table' containing 'datetime'. |
| value | Numeric scalar target exposure.              |

**Value**

A numeric vector of target positions with length 'nrow(DT)'.

---

strat\_credit\_spread\_momentum\_action\_plan  
*Credit-Spread-Momentum Action Plan*

---

**Description**

Applies the credit-spread momentum rule to the latest row and translates the resulting target exposure into an executable action plan.

**Usage**

```
strat_credit_spread_momentum_action_plan(
  DT,
  state,
  spread_col = "credit_spread",
  n = 20L,
  yield_col = "ytm",
  benchmark_yield_col = "benchmark_ytm",
  momentum_col = NULL,
  long_threshold = 0,
  short_threshold = 0,
  target_size = 1,
  compute_features = TRUE,
  strat_id = 609L,
  tol_pos = 0.1,
  debug = FALSE
)
```

**Arguments**

|                     |                                                                                |
|---------------------|--------------------------------------------------------------------------------|
| DT                  | A 'data.table' containing credit-spread inputs or a precomputed spread column. |
| state               | Named list describing the current trading state.                               |
| spread_col          | Credit-spread column name.                                                     |
| n                   | Integer momentum lookback window.                                              |
| yield_col           | Issuer-yield column used when 'spread_col' must be built.                      |
| benchmark_yield_col | Benchmark-yield column used when 'spread_col' must be built.                   |

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| momentum_col     | Optional precomputed momentum column name.                                                      |
| long_threshold   | Numeric threshold above which momentum is treated as positive enough to go long.                |
| short_threshold  | Numeric threshold below which momentum is treated as negative enough to go short.               |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing spread and momentum features are added to 'DT' in place.          |
| strat_id         | Integer strategy identifier recorded on generated actions.                                      |
| tol_pos          | Numeric tolerance passed to the action planner.                                                 |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A list produced by 'gen\_action\_plan\_rcpp()'.

---

strat\_credit\_spread\_momentum\_tgt\_pos  
*Credit-Spread-Momentum Target Positions*

---

**Description**

Generates a target-position path from momentum in credit spreads. Positive spread momentum targets a long widener exposure and negative spread momentum targets a short or tightening exposure.

**Usage**

```
strat_credit_spread_momentum_tgt_pos(
  DT,
  spread_col = "credit_spread",
  n = 20L,
  yield_col = "ytm",
  benchmark_yield_col = "benchmark_ytm",
  momentum_col = NULL,
  long_threshold = 0,
  short_threshold = 0,
  target_size = 1,
  compute_features = TRUE,
  debug = FALSE
)
```

**Arguments**

|                     |                                                                                                 |
|---------------------|-------------------------------------------------------------------------------------------------|
| DT                  | A 'data.table' containing credit-spread inputs or a precomputed spread column.                  |
| spread_col          | Credit-spread column name.                                                                      |
| n                   | Integer momentum lookback window.                                                               |
| yield_col           | Issuer-yield column used when 'spread_col' must be built.                                       |
| benchmark_yield_col | Benchmark-yield column used when 'spread_col' must be built.                                    |
| momentum_col        | Optional precomputed momentum column name.                                                      |
| long_threshold      | Numeric threshold above which momentum is treated as positive enough to go long.                |
| short_threshold     | Numeric threshold below which momentum is treated as negative enough to go short.               |
| target_size         | Numeric absolute target exposure.                                                               |
| compute_features    | Logical; when 'TRUE', missing spread and momentum features are added to 'DT' in place.          |
| debug               | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

strat\_credit\_spread\_revert\_action\_plan  
*Credit-Spread-Reversion Action Plan*

---

**Description**

Applies the credit-spread mean-reversion rule to the latest row and translates the resulting target exposure into an executable action plan.

**Usage**

```
strat_credit_spread_revert_action_plan(
  DT,
  state,
  signal_col = "excess_spread",
  z_n = 20L,
  entry_z = 2,
  exit_z = 0.5,
  spread_col = "credit_spread",
  benchmark_spread_col = "benchmark_spread",
```

```

    yield_col = "ytm",
    benchmark_yield_col = "benchmark_ytm",
    target_size = 1,
    compute_features = TRUE,
    sample = TRUE,
    strat_id = 610L,
    tol_pos = 0.1,
    debug = FALSE
  )

```

### Arguments

|                      |                                                                                                 |
|----------------------|-------------------------------------------------------------------------------------------------|
| DT                   | A 'data.table' containing credit-spread inputs or a precomputed signal column.                  |
| state                | Named list describing the current trading state.                                                |
| signal_col           | Credit signal column name. Defaults to 'excess_spread'.                                         |
| z_n                  | Integer rolling window used for z-scores.                                                       |
| entry_z              | Numeric absolute z-score threshold used for entries.                                            |
| exit_z               | Numeric absolute z-score threshold used for exits.                                              |
| spread_col           | Credit-spread column used when 'signal_col' must be built.                                      |
| benchmark_spread_col | Benchmark or sector spread column used when 'signal_col = "excess_spread"' must be built.       |
| yield_col            | Issuer-yield column used when 'spread_col' must be built.                                       |
| benchmark_yield_col  | Benchmark-yield column used when 'spread_col' must be built.                                    |
| target_size          | Numeric absolute target exposure.                                                               |
| compute_features     | Logical; when 'TRUE', missing signal and z-score features are added to 'DT' in place.           |
| sample               | Logical; if 'TRUE', rolling z-score uses sample standard deviation.                             |
| strat_id             | Integer strategy identifier recorded on generated actions.                                      |
| tol_pos              | Numeric tolerance passed to the action planner.                                                 |
| debug                | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

### Value

A list produced by 'gen\_action\_plan\_rcpp()'.

---

strat\_credit\_spread\_revert\_tgt\_pos

*Credit-Spread-Reversion Target Positions*


---

## Description

Generates a mean-reversion target-position path from standardized credit spread or excess-spread signals. Wide positive excess spread targets a long tightening exposure, while strongly negative excess spread targets a short exposure.

## Usage

```
strat_credit_spread_revert_tgt_pos(
  DT,
  signal_col = "excess_spread",
  z_n = 20L,
  entry_z = 2,
  exit_z = 0.5,
  spread_col = "credit_spread",
  benchmark_spread_col = "benchmark_spread",
  yield_col = "ytm",
  benchmark_yield_col = "benchmark_ytm",
  target_size = 1,
  compute_features = TRUE,
  sample = TRUE,
  debug = FALSE
)
```

## Arguments

|                      |                                                                                           |
|----------------------|-------------------------------------------------------------------------------------------|
| DT                   | A 'data.table' containing credit-spread inputs or a precomputed signal column.            |
| signal_col           | Credit signal column name. Defaults to 'excess_spread'.                                   |
| z_n                  | Integer rolling window used for z-scores.                                                 |
| entry_z              | Numeric absolute z-score threshold used for entries.                                      |
| exit_z               | Numeric absolute z-score threshold used for exits.                                        |
| spread_col           | Credit-spread column used when 'signal_col' must be built.                                |
| benchmark_spread_col | Benchmark or sector spread column used when 'signal_col = "excess_spread"' must be built. |
| yield_col            | Issuer-yield column used when 'spread_col' must be built.                                 |
| benchmark_yield_col  | Benchmark-yield column used when 'spread_col' must be built.                              |
| target_size          | Numeric absolute target exposure.                                                         |

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| compute_features | Logical; when 'TRUE', missing signal and z-score features are added to 'DT' in place.           |
| sample           | Logical; if 'TRUE', rolling z-score uses sample standard deviation.                             |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

strat\_cross\_sectional\_rank\_allocator\_action\_plan

*Cross-Sectional-Rank-Allocator Action Plan*

---

**Description**

Applies the latest cross-sectional target weights to a portfolio state and converts them into a portfolio-adjustment table plus order intents.

**Usage**

```
strat_cross_sectional_rank_allocator_action_plan(
  DT,
  portfolio_state,
  equity,
  date_col = "date",
  asset_col = "asset",
  signal_col = "score",
  long_n = 1L,
  short_n = 0L,
  gross_exposure = 1,
  strat_id = 507L,
  pricing_method = "market",
  debug = FALSE
)
```

**Arguments**

|                 |                                                                                    |
|-----------------|------------------------------------------------------------------------------------|
| DT              | A panel 'data.table' containing 'date', 'asset', and a signal column.              |
| portfolio_state | A 'data.table' containing current per-asset portfolio state.                       |
| equity          | Numeric total portfolio equity used to scale target weights into target notionals. |
| date_col        | Date column used for cross-sectional grouping.                                     |
| asset_col       | Asset identifier column.                                                           |
| signal_col      | Ranking signal column.                                                             |

|                |                                                                                                 |
|----------------|-------------------------------------------------------------------------------------------------|
| long_n         | Integer number of assets to allocate long each date.                                            |
| short_n        | Integer number of assets to allocate short each date.                                           |
| gross_exposure | Numeric gross exposure allocated across each side.                                              |
| strat_id       | Integer strategy identifier recorded on generated order intents.                                |
| pricing_method | Pricing method passed to 'build_order_intents()'.                                               |
| debug          | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A list with 'adjustment\_plan', 'order\_intents', and the latest target weights.

---

strat\_cross\_sectional\_rank\_allocator\_tgt\_pos  
*Cross-Sectional-Rank-Allocator Target Positions*

---

**Description**

Generates per-row target weights from cross-sectional signal ranks within each date. Top-ranked assets receive positive target weights and optionally bottom-ranked assets receive negative target weights.

**Usage**

```
strat_cross_sectional_rank_allocator_tgt_pos(
  DT,
  date_col = "date",
  asset_col = "asset",
  signal_col = "score",
  long_n = 1L,
  short_n = 0L,
  gross_exposure = 1,
  debug = FALSE
)
```

**Arguments**

|                |                                                                                                 |
|----------------|-------------------------------------------------------------------------------------------------|
| DT             | A panel 'data.table' containing 'date', 'asset', and a signal column.                           |
| date_col       | Date column used for cross-sectional grouping.                                                  |
| asset_col      | Asset identifier column.                                                                        |
| signal_col     | Ranking signal column.                                                                          |
| long_n         | Integer number of assets to allocate long each date.                                            |
| short_n        | Integer number of assets to allocate short each date.                                           |
| gross_exposure | Numeric gross exposure allocated across each side.                                              |
| debug          | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A numeric vector of target weights aligned with ‘DT’, or a list when ‘debug = TRUE’.

---

|                                    |
|------------------------------------|
| strat_curve_butterfly_action_plan  |
| <i>Curve-Butterfly Action Plan</i> |

---

**Description**

Applies the curve-butterfly rule to the latest row and translates the resulting target exposure into an executable action plan.

**Usage**

```
strat_curve_butterfly_action_plan(  
  DT,  
  state,  
  short_rate_col = "short_rate",  
  mid_rate_col = "mid_rate",  
  long_rate_col = "long_rate",  
  long_threshold = 0,  
  short_threshold = 0,  
  target_size = 1,  
  compute_features = TRUE,  
  strat_id = 608L,  
  tol_pos = 0.1,  
  debug = FALSE  
)
```

**Arguments**

|                  |                                                                                             |
|------------------|---------------------------------------------------------------------------------------------|
| DT               | A ‘data.table’ containing the three curve points or a precomputed ‘curve_butterfly’ column. |
| state            | Named list describing the current trading state.                                            |
| short_rate_col   | Short-end rate column name.                                                                 |
| mid_rate_col     | Belly rate column name.                                                                     |
| long_rate_col    | Long-end rate column name.                                                                  |
| long_threshold   | Numeric threshold above which butterfly is treated as positive enough to go long.           |
| short_threshold  | Numeric threshold below which butterfly is treated as negative enough to go short.          |
| target_size      | Numeric absolute target exposure.                                                           |
| compute_features | Logical; when ‘TRUE’, missing butterfly features are added to ‘DT’ in place.                |

|          |                                                                                                 |
|----------|-------------------------------------------------------------------------------------------------|
| strat_id | Integer strategy identifier recorded on generated actions.                                      |
| tol_pos  | Numeric tolerance passed to the action planner.                                                 |
| debug    | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A list produced by 'gen\_action\_plan\_rcpp()'.

---

|                                         |
|-----------------------------------------|
| strat_curve_butterfly_tgt_pos           |
| <i>Curve-Butterfly Target Positions</i> |

---

**Description**

Generates a target-position path from a simple yield-curve butterfly measure. Positive butterfly values target a long-butterfly exposure and negative values target a short-butterfly exposure.

**Usage**

```
strat_curve_butterfly_tgt_pos(  
  DT,  
  short_rate_col = "short_rate",  
  mid_rate_col = "mid_rate",  
  long_rate_col = "long_rate",  
  long_threshold = 0,  
  short_threshold = 0,  
  target_size = 1,  
  compute_features = TRUE,  
  debug = FALSE  
)
```

**Arguments**

|                 |                                                                                             |
|-----------------|---------------------------------------------------------------------------------------------|
| DT              | A 'data.table' containing the three curve points or a precomputed 'curve_butterfly' column. |
| short_rate_col  | Short-end rate column name.                                                                 |
| mid_rate_col    | Belly rate column name.                                                                     |
| long_rate_col   | Long-end rate column name.                                                                  |
| long_threshold  | Numeric threshold above which butterfly is treated as positive enough to go long.           |
| short_threshold | Numeric threshold below which butterfly is treated as negative enough to go short.          |
| target_size     | Numeric absolute target exposure.                                                           |

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| compute_features | Logical; when 'TRUE', missing butterfly features are added to 'DT' in place.                    |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

|                                    |
|------------------------------------|
| strat_curve_steepener_action_plan  |
| <i>Curve-Steepener Action Plan</i> |

---

**Description**

Applies the curve-steepener rule to the latest row and translates the resulting target exposure into an executable action plan.

**Usage**

```
strat_curve_steepener_action_plan(  
  DT,  
  state,  
  short_rate_col = "short_rate",  
  long_rate_col = "long_rate",  
  long_threshold = 0,  
  short_threshold = 0,  
  target_size = 1,  
  compute_features = TRUE,  
  strat_id = 603L,  
  tol_pos = 0.1,  
  debug = FALSE  
)
```

**Arguments**

|                 |                                                                                               |
|-----------------|-----------------------------------------------------------------------------------------------|
| DT              | A 'data.table' containing short- and long-rate columns or a precomputed 'curve_slope' column. |
| state           | Named list describing the current trading state.                                              |
| short_rate_col  | Short-end rate column name.                                                                   |
| long_rate_col   | Long-end rate column name.                                                                    |
| long_threshold  | Numeric threshold above which slope is treated as steepening enough to go long.               |
| short_threshold | Numeric threshold below which slope is treated as flattening enough to go short.              |
| target_size     | Numeric absolute target exposure.                                                             |

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| compute_features | Logical; when 'TRUE', missing slope features are added to 'DT' in place.                        |
| strat_id         | Integer strategy identifier recorded on generated actions.                                      |
| tol_pos          | Numeric tolerance passed to the action planner.                                                 |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A list produced by 'gen\_action\_plan\_rcpp()'.

---

strat\_curve\_steepener\_tgt\_pos  
*Curve-Steepener Target Positions*

---

**Description**

Generates a simple curve-steepener target-position path from yield-curve slope. Positive slope targets a steepener exposure and negative slope targets a flattener exposure.

**Usage**

```
strat_curve_steepener_tgt_pos(
  DT,
  short_rate_col = "short_rate",
  long_rate_col = "long_rate",
  long_threshold = 0,
  short_threshold = 0,
  target_size = 1,
  compute_features = TRUE,
  debug = FALSE
)
```

**Arguments**

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A 'data.table' containing short- and long-rate columns or a precomputed 'curve_slope' column.   |
| short_rate_col   | Short-end rate column name.                                                                     |
| long_rate_col    | Long-end rate column name.                                                                      |
| long_threshold   | Numeric threshold above which slope is treated as steepening enough to go long.                 |
| short_threshold  | Numeric threshold below which slope is treated as flattening enough to go short.                |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing slope features are added to 'DT' in place.                        |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

```
strat_curve_steepener_zscore_action_plan
```

*Curve-Steepener-Z-Score Action Plan*

---

**Description**

Applies the curve-slope z-score rule to the latest row and translates the resulting target exposure into an executable action plan.

**Usage**

```
strat_curve_steepener_zscore_action_plan(
  DT,
  state,
  short_rate_col = "short_rate",
  long_rate_col = "long_rate",
  z_n = 252L,
  long_z = 1,
  short_z = -1,
  mode = c("momentum", "reversion"),
  target_size = 1,
  compute_features = TRUE,
  strat_id = 605L,
  tol_pos = 0.1,
  debug = FALSE
)
```

**Arguments**

|                  |                                                                                                                         |
|------------------|-------------------------------------------------------------------------------------------------------------------------|
| DT               | A 'data.table' containing short- and long-rate columns or precomputed 'curve_slope' and 'zscore_curve_slope_*' columns. |
| state            | Named list describing the current trading state.                                                                        |
| short_rate_col   | Short-end rate column name.                                                                                             |
| long_rate_col    | Long-end rate column name.                                                                                              |
| z_n              | Integer rolling z-score window.                                                                                         |
| long_z           | Numeric upper z-score threshold.                                                                                        |
| short_z          | Numeric lower z-score threshold.                                                                                        |
| mode             | Character strategy mode: 'momentum' or 'reversion'.                                                                     |
| target_size      | Numeric absolute target exposure.                                                                                       |
| compute_features | Logical; when 'TRUE', missing slope and z-score features are added to 'DT' in place.                                    |

|          |                                                                                                 |
|----------|-------------------------------------------------------------------------------------------------|
| strat_id | Integer strategy identifier recorded on generated actions.                                      |
| tol_pos  | Numeric tolerance passed to the action planner.                                                 |
| debug    | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A list produced by 'gen\_action\_plan\_rcpp()'.

---

|                                                 |
|-------------------------------------------------|
| strat_curve_steepener_zscore_tgt_pos            |
| <i>Curve-Steepener-Z-Score Target Positions</i> |

---

**Description**

Generates a curve-steepener target-position path from the rolling z-score of curve slope, with either momentum or reversion interpretation.

**Usage**

```
strat_curve_steepener_zscore_tgt_pos(  
  DT,  
  short_rate_col = "short_rate",  
  long_rate_col = "long_rate",  
  z_n = 252L,  
  long_z = 1,  
  short_z = -1,  
  mode = c("momentum", "reversion"),  
  target_size = 1,  
  compute_features = TRUE,  
  debug = FALSE  
)
```

**Arguments**

|                |                                                                                                                         |
|----------------|-------------------------------------------------------------------------------------------------------------------------|
| DT             | A 'data.table' containing short- and long-rate columns or precomputed 'curve_slope' and 'zscore_curve_slope_*' columns. |
| short_rate_col | Short-end rate column name.                                                                                             |
| long_rate_col  | Long-end rate column name.                                                                                              |
| z_n            | Integer rolling z-score window.                                                                                         |
| long_z         | Numeric upper z-score threshold.                                                                                        |
| short_z        | Numeric lower z-score threshold.                                                                                        |
| mode           | Character strategy mode: 'momentum' or 'reversion'.                                                                     |
| target_size    | Numeric absolute target exposure.                                                                                       |

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| compute_features | Logical; when 'TRUE', missing slope and z-score features are added to 'DT' in place.            |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

strat\_donchian\_breakout\_action\_plan  
*Donchian-Breakout Action Plan*

---

**Description**

Applies the Donchian breakout rule to the latest bar and translates the resulting target exposure into an executable action plan.

**Usage**

```
strat_donchian_breakout_action_plan(
  DT,
  state,
  n = 20L,
  target_size = 1,
  compute_features = TRUE,
  strat_id = 302L,
  tol_pos = 0.1,
  debug = FALSE
)
```

**Arguments**

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle 'data.table'.                                                                          |
| state            | Named list describing the current trading state.                                                |
| n                | Integer Donchian window.                                                                        |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing Donchian features are added to 'DT' in place.                     |
| strat_id         | Integer strategy identifier recorded on generated actions.                                      |
| tol_pos          | Numeric tolerance passed to the action planner.                                                 |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A list produced by ‘gen\_action\_plan\_rcpp()’.

---

|                                           |
|-------------------------------------------|
| strat_donchian_breakout_tgt_pos           |
| <i>Donchian-Breakout Target Positions</i> |

---

**Description**

Generates a simple breakout target-position path from Donchian channel breaches. A close above the prior upper channel opens or maintains a long target, and a close below the prior lower channel opens or maintains a short target.

**Usage**

```
strat_donchian_breakout_tgt_pos(  
  DT,  
  n = 20L,  
  target_size = 1,  
  compute_features = TRUE,  
  debug = FALSE  
)
```

**Arguments**

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle ‘data.table’.                                                                          |
| n                | Integer Donchian window.                                                                        |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when ‘TRUE’, missing Donchian features are added to ‘DT’ in place.                     |
| debug            | Logical; when ‘TRUE’, returns a list with the generated target vector and feature column names. |

**Value**

A numeric vector of target positions, or a list when ‘debug = TRUE’.

---

strat\_donchian\_retest\_breakout\_action\_plan

*Donchian-Retest-Breakout Action Plan*


---

## Description

Applies the Donchian retest-breakout rule to the latest bar and translates the resulting target exposure into an executable action plan.

## Usage

```
strat_donchian_retest_breakout_action_plan(
  DT,
  state,
  n = 20L,
  retest_buffer = 0,
  confirm_n = 5L,
  target_size = 1,
  compute_features = TRUE,
  strat_id = 313L,
  tol_pos = 0.1,
  debug = FALSE
)
```

## Arguments

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle 'data.table'.                                                                          |
| state            | Named list describing the current trading state.                                                |
| n                | Integer Donchian window.                                                                        |
| retest_buffer    | Numeric distance around the broken channel allowed for a valid retest.                          |
| confirm_n        | Integer maximum number of bars to wait for the retest.                                          |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing Donchian features are added to 'DT' in place.                     |
| strat_id         | Integer strategy identifier recorded on generated actions.                                      |
| tol_pos          | Numeric tolerance passed to the action planner.                                                 |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

## Value

A list produced by 'gen\_action\_plan\_rcpp()'.

---

strat\_donchian\_retest\_breakout\_tgt\_pos

*Donchian-Retest-Breakout Target Positions*


---

### Description

Generates a Donchian breakout target-position path that waits for a retest of the broken channel boundary before entering. Breakouts arm a pending state, and entries are only activated if price retests the broken level within a limited confirmation window.

### Usage

```
strat_donchian_retest_breakout_tgt_pos(
  DT,
  n = 20L,
  retest_buffer = 0,
  confirm_n = 5L,
  target_size = 1,
  compute_features = TRUE,
  debug = FALSE
)
```

### Arguments

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle 'data.table'.                                                                          |
| n                | Integer Donchian window.                                                                        |
| retest_buffer    | Numeric distance around the broken channel allowed for a valid retest.                          |
| confirm_n        | Integer maximum number of bars to wait for the retest.                                          |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing Donchian features are added to 'DT' in place.                     |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

### Value

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

strat\_donchian\_turtle\_action\_plan

*Donchian-Turtle Action Plan*


---

## Description

Applies the Donchian Turtle rule to the latest bar and translates the resulting target exposure into an executable action plan.

## Usage

```
strat_donchian_turtle_action_plan(
  DT,
  state,
  entry_n = 55L,
  exit_n = 20L,
  target_size = 1,
  compute_features = TRUE,
  strat_id = 307L,
  tol_pos = 0.1,
  debug = FALSE
)
```

## Arguments

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle 'data.table'.                                                                          |
| state            | Named list describing the current trading state.                                                |
| entry_n          | Integer Donchian entry window.                                                                  |
| exit_n           | Integer Donchian exit window.                                                                   |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing Donchian features are added to 'DT' in place.                     |
| strat_id         | Integer strategy identifier recorded on generated actions.                                      |
| tol_pos          | Numeric tolerance passed to the action planner.                                                 |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

## Value

A list produced by 'gen\_action\_plan\_rcpp()'.

---

strat\_donchian\_turtle\_tgt\_pos

*Donchian-Turtle Target Positions*


---

### Description

Generates a Turtle-style breakout target-position path using a long-window Donchian entry breakout and a shorter-window Donchian exit breakout.

### Usage

```
strat_donchian_turtle_tgt_pos(
  DT,
  entry_n = 55L,
  exit_n = 20L,
  target_size = 1,
  compute_features = TRUE,
  debug = FALSE
)
```

### Arguments

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle 'data.table'.                                                                          |
| entry_n          | Integer Donchian entry window.                                                                  |
| exit_n           | Integer Donchian exit window.                                                                   |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing Donchian features are added to 'DT' in place.                     |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

### Value

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

strat\_ema\_cross\_action\_plan

*EMA-Cross Action Plan*


---

### Description

Applies the EMA-cross target rule to the latest bar and translates the resulting target exposure into an executable action plan.

**Usage**

```
strat_ema_cross_action_plan(
  DT,
  state,
  fast = 20L,
  slow = 50L,
  low_atr_threshold = 5L,
  freshness_floor = 18L,
  tp_ratio = 0.05,
  sl_ratio = 0.02,
  compute_features = TRUE,
  atr_h = 12L,
  atr_window = 300L,
  strat_id = 101L,
  tol_pos = 0.1,
  debug = FALSE
)
```

**Arguments**

|                   |                                                                                |
|-------------------|--------------------------------------------------------------------------------|
| DT                | A candle 'data.table'.                                                         |
| state             | Named list describing the current trading state.                               |
| fast              | Integer fast EMA length.                                                       |
| slow              | Integer slow EMA length.                                                       |
| low_atr_threshold | Numeric percentile threshold used to define the low-volatility gate.           |
| freshness_floor   | Integer maximum cross age allowed for a live target.                           |
| tp_ratio          | Numeric take-profit guard used by 'calc_EMA_cross()'.                          |
| sl_ratio          | Numeric stop-loss guard used by 'calc_EMA_cross()'.                            |
| compute_features  | Logical; when 'TRUE', missing EMA and ATR features are added to 'DT' in place. |
| atr_h             | Numeric ATR half-life used by the low-volatility gate.                         |
| atr_window        | Integer rolling window used by the ATR quantile.                               |
| strat_id          | Integer strategy identifier recorded on generated actions.                     |
| tol_pos           | Numeric tolerance passed to the action planner.                                |
| debug             | Logical; when 'TRUE', returns a list with the generated target column name.    |

**Value**

A list produced by 'gen\_action\_plan\_rcpp()'.

---

strat\_ema\_cross\_adx\_action\_plan

*EMA-Cross-ADX Action Plan*


---

### Description

Applies the EMA-cross-plus-ADX rule to the latest bar and translates the resulting target exposure into an executable action plan.

### Usage

```
strat_ema_cross_adx_action_plan(
  DT,
  state,
  fast = 20L,
  slow = 50L,
  adx_n = 14L,
  adx_threshold = 20,
  target_size = 1,
  compute_features = TRUE,
  strat_id = 103L,
  tol_pos = 0.1,
  debug = FALSE
)
```

### Arguments

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle ‘data.table’.                                                                          |
| state            | Named list describing the current trading state.                                                |
| fast             | Integer fast EMA length.                                                                        |
| slow             | Integer slow EMA length.                                                                        |
| adx_n            | Integer ADX window.                                                                             |
| adx_threshold    | Numeric ADX threshold required for live targets.                                                |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when ‘TRUE’, missing EMA and ADX features are added to ‘DT’ in place.                  |
| strat_id         | Integer strategy identifier recorded on generated actions.                                      |
| tol_pos          | Numeric tolerance passed to the action planner.                                                 |
| debug            | Logical; when ‘TRUE’, returns a list with the generated target vector and feature column names. |

### Value

A list produced by ‘gen\_action\_plan\_rcpp()’.

---

|                                       |
|---------------------------------------|
| strat_ema_cross_adx_tgt_pos           |
| <i>EMA-Cross-ADX Target Positions</i> |

---

**Description**

Generates a simple EMA-cross target-position path filtered by ADX trend strength. EMA direction drives the side, and low-ADX regimes stay flat.

**Usage**

```
strat_ema_cross_adx_tgt_pos(  
  DT,  
  fast = 20L,  
  slow = 50L,  
  adx_n = 14L,  
  adx_threshold = 20,  
  target_size = 1,  
  compute_features = TRUE,  
  debug = FALSE  
)
```

**Arguments**

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle 'data.table'.                                                                          |
| fast             | Integer fast EMA length.                                                                        |
| slow             | Integer slow EMA length.                                                                        |
| adx_n            | Integer ADX window.                                                                             |
| adx_threshold    | Numeric ADX threshold required for live targets.                                                |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing EMA and ADX features are added to 'DT' in place.                  |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

strat\_ema\_cross\_slope\_confirm\_action\_plan

*EMA-Cross-Slope-Confirm Action Plan*


---

### Description

Applies the EMA-cross-plus-slope-confirmation rule to the latest bar and translates the resulting target exposure into an executable action plan.

### Usage

```
strat_ema_cross_slope_confirm_action_plan(
  DT,
  state,
  fast = 20L,
  slow = 50L,
  slope_lag = 1L,
  target_size = 1,
  compute_features = TRUE,
  strat_id = 104L,
  tol_pos = 0.1,
  debug = FALSE
)
```

### Arguments

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle 'data.table'.                                                                          |
| state            | Named list describing the current trading state.                                                |
| fast             | Integer fast EMA length.                                                                        |
| slow             | Integer slow EMA length.                                                                        |
| slope_lag        | Integer lag used to estimate EMA slope.                                                         |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing EMA features are added to 'DT' in place.                          |
| strat_id         | Integer strategy identifier recorded on generated actions.                                      |
| tol_pos          | Numeric tolerance passed to the action planner.                                                 |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

### Value

A list produced by 'gen\_action\_plan\_rcpp()'.

---

strat\_ema\_cross\_slope\_confirm\_tgt\_pos

*EMA-Cross-Slope-Confirm Target Positions*


---

## Description

Generates a target-position path from EMA alignment and EMA slope confirmation. A long target requires the fast EMA above the slow EMA and both EMAs sloping upward, while a short target requires the reverse.

## Usage

```
strat_ema_cross_slope_confirm_tgt_pos(
  DT,
  fast = 20L,
  slow = 50L,
  slope_lag = 1L,
  target_size = 1,
  compute_features = TRUE,
  debug = FALSE
)
```

## Arguments

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle 'data.table'.                                                                          |
| fast             | Integer fast EMA length.                                                                        |
| slow             | Integer slow EMA length.                                                                        |
| slope_lag        | Integer lag used to estimate EMA slope.                                                         |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing EMA features are added to 'DT' in place.                          |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

## Value

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

strat\_ema\_cross\_tgt\_pos

*EMA-Cross Target Positions*


---

### Description

Generates target positions from a simple EMA-cross strategy with ATR-based gating. The wrapper can either reuse precomputed features or compute the required EMA and ATR features in place.

### Usage

```
strat_ema_cross_tgt_pos(
  DT,
  fast = 20L,
  slow = 50L,
  low_atr_threshold = 5L,
  freshness_floor = 18L,
  tp_ratio = 0.05,
  sl_ratio = 0.02,
  compute_features = TRUE,
  atr_h = 12L,
  atr_window = 300L,
  debug = FALSE
)
```

### Arguments

|                   |                                                                                |
|-------------------|--------------------------------------------------------------------------------|
| DT                | A candle 'data.table'.                                                         |
| fast              | Integer fast EMA length.                                                       |
| slow              | Integer slow EMA length.                                                       |
| low_atr_threshold | Numeric percentile threshold used to define the low-volatility gate.           |
| freshness_floor   | Integer maximum cross age allowed for a live target.                           |
| tp_ratio          | Numeric take-profit guard used by 'calc_ema_cross()'.                          |
| sl_ratio          | Numeric stop-loss guard used by 'calc_ema_cross()'.                            |
| compute_features  | Logical; when 'TRUE', missing EMA and ATR features are added to 'DT' in place. |
| atr_h             | Numeric ATR half-life used by the low-volatility gate.                         |
| atr_window        | Integer rolling window used by the ATR quantile.                               |
| debug             | Logical; when 'TRUE', returns a list with the generated target column name.    |

### Value

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

```
strat_ema_triple_trend_action_plan
```

*EMA-Triple-Trend Action Plan*

---

## Description

Applies the triple-EMA trend rule to the latest bar and translates the resulting target exposure into an executable action plan.

## Usage

```
strat_ema_triple_trend_action_plan(
  DT,
  state,
  fast = 20L,
  mid = 50L,
  slow = 100L,
  target_size = 1,
  compute_features = TRUE,
  strat_id = 106L,
  tol_pos = 0.1,
  debug = FALSE
)
```

## Arguments

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle 'data.table'.                                                                          |
| state            | Named list describing the current trading state.                                                |
| fast             | Integer fast EMA window.                                                                        |
| mid              | Integer medium EMA window.                                                                      |
| slow             | Integer slow EMA window.                                                                        |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing EMA features are added to 'DT' in place.                          |
| strat_id         | Integer strategy identifier recorded on generated actions.                                      |
| tol_pos          | Numeric tolerance passed to the action planner.                                                 |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

## Value

A list produced by 'gen\_action\_plan\_rcpp()'.

---

strat\_ema\_triple\_trend\_tgt\_pos

*EMA-Triple-Trend Target Positions*


---

### Description

Generates a simple target-position path from triple-EMA alignment. Long exposure is targeted when the fast EMA is above the medium EMA and the medium EMA is above the slow EMA. Short exposure is targeted for the reverse ordering.

### Usage

```
strat_ema_triple_trend_tgt_pos(
  DT,
  fast = 20L,
  mid = 50L,
  slow = 100L,
  target_size = 1,
  compute_features = TRUE,
  debug = FALSE
)
```

### Arguments

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle 'data.table'.                                                                          |
| fast             | Integer fast EMA window.                                                                        |
| mid              | Integer medium EMA window.                                                                      |
| slow             | Integer slow EMA window.                                                                        |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing EMA features are added to 'DT' in place.                          |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

### Value

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

strat\_funding\_basis\_convergence\_action\_plan

*Funding-Basis-Convergence Action Plan*


---

## Description

Applies the funding-and-basis convergence rule to the latest row and translates the resulting target exposure into an executable action plan.

## Usage

```
strat_funding_basis_convergence_action_plan(
  DT,
  state,
  z_n = 20L,
  entry_z = 2,
  exit_z = 0.5,
  basis_col = NULL,
  spot_col = "spot",
  forward_col = "forward",
  domestic_rate_col = "r_domestic",
  foreign_rate_col = "r_foreign",
  tenor_col = "tenor_years",
  tenor_tag = "1m",
  funding_col = NULL,
  funding_weight = 1,
  target_size = 1,
  compute_features = TRUE,
  sample = TRUE,
  strat_id = 611L,
  tol_pos = 0.1,
  debug = FALSE
)
```

## Arguments

|             |                                                                                                       |
|-------------|-------------------------------------------------------------------------------------------------------|
| DT          | A 'data.table' containing a precomputed basis column or the inputs needed to build an FX-style basis. |
| state       | Named list describing the current trading state.                                                      |
| z_n         | Integer rolling window used for z-scores.                                                             |
| entry_z     | Numeric absolute z-score threshold used for entries.                                                  |
| exit_z      | Numeric absolute z-score threshold used for exits.                                                    |
| basis_col   | Optional basis column name. Defaults to 'fx_basis_<tag>'.                                             |
| spot_col    | Spot-price column name used when basis must be built.                                                 |
| forward_col | Forward-price column name used when basis must be built.                                              |

|                   |                                                                                                 |
|-------------------|-------------------------------------------------------------------------------------------------|
| domestic_rate_col | Domestic-rate column used when basis must be built.                                             |
| foreign_rate_col  | Foreign-rate column used when basis must be built.                                              |
| tenor_col         | Tenor-in-years column used when basis must be built.                                            |
| tenor_tag         | Character label appended to the built basis column name.                                        |
| funding_col       | Optional funding-rate column added into the signal.                                             |
| funding_weight    | Numeric scaling applied to 'funding_col'.                                                       |
| target_size       | Numeric absolute target exposure.                                                               |
| compute_features  | Logical; when 'TRUE', missing basis, signal, and z-score features are added to 'DT' in place.   |
| sample            | Logical; if 'TRUE', rolling z-score uses sample standard deviation.                             |
| strat_id          | Integer strategy identifier recorded on generated actions.                                      |
| tol_pos           | Numeric tolerance passed to the action planner.                                                 |
| debug             | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A list produced by 'gen\_action\_plan\_rcpp()'.

---

strat\_funding\_basis\_convergence\_tgt\_pos  
*Funding-Basis-Convergence Target Positions*

---

**Description**

Generates a mean-reversion target-position path from a basis-plus-funding dislocation signal. The strategy fades extreme negative or positive standardized dislocations and exits as the signal normalizes.

**Usage**

```
strat_funding_basis_convergence_tgt_pos(
  DT,
  z_n = 20L,
  entry_z = 2,
  exit_z = 0.5,
  basis_col = NULL,
  spot_col = "spot",
  forward_col = "forward",
  domestic_rate_col = "r_domestic",
  foreign_rate_col = "r_foreign",
```

```

    tenor_col = "tenor_years",
    tenor_tag = "1m",
    funding_col = NULL,
    funding_weight = 1,
    target_size = 1,
    compute_features = TRUE,
    sample = TRUE,
    debug = FALSE
  )

```

### Arguments

|                   |                                                                                                       |
|-------------------|-------------------------------------------------------------------------------------------------------|
| DT                | A 'data.table' containing a precomputed basis column or the inputs needed to build an FX-style basis. |
| z_n               | Integer rolling window used for z-scores.                                                             |
| entry_z           | Numeric absolute z-score threshold used for entries.                                                  |
| exit_z            | Numeric absolute z-score threshold used for exits.                                                    |
| basis_col         | Optional basis column name. Defaults to 'fx_basis_<tag>'.                                             |
| spot_col          | Spot-price column name used when basis must be built.                                                 |
| forward_col       | Forward-price column name used when basis must be built.                                              |
| domestic_rate_col | Domestic-rate column used when basis must be built.                                                   |
| foreign_rate_col  | Foreign-rate column used when basis must be built.                                                    |
| tenor_col         | Tenor-in-years column used when basis must be built.                                                  |
| tenor_tag         | Character label appended to the built basis column name.                                              |
| funding_col       | Optional funding-rate column added into the signal.                                                   |
| funding_weight    | Numeric scaling applied to 'funding_col'.                                                             |
| target_size       | Numeric absolute target exposure.                                                                     |
| compute_features  | Logical; when 'TRUE', missing basis, signal, and z-score features are added to 'DT' in place.         |
| sample            | Logical; if 'TRUE', rolling z-score uses sample standard deviation.                                   |
| debug             | Logical; when 'TRUE', returns a list with the generated target vector and feature column names.       |

### Value

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

strat\_fx\_carry\_action\_plan

*FX-Carry Action Plan*


---

## Description

Applies the FX-carry rule to the latest row and translates the resulting target exposure into an executable action plan.

## Usage

```
strat_fx_carry_action_plan(
  DT,
  state,
  domestic_rate_col = "r_domestic",
  foreign_rate_col = "r_foreign",
  tenor_col = NULL,
  tenor_tag = "1m",
  long_threshold = 0,
  short_threshold = 0,
  target_size = 1,
  compute_features = TRUE,
  strat_id = 601L,
  tol_pos = 0.1,
  debug = FALSE
)
```

## Arguments

|                   |                                                                                 |
|-------------------|---------------------------------------------------------------------------------|
| DT                | A 'data.table' containing FX carry inputs or a precomputed 'fx_carry_*' column. |
| state             | Named list describing the current trading state.                                |
| domestic_rate_col | Domestic annualized rate column name.                                           |
| foreign_rate_col  | Foreign annualized rate column name.                                            |
| tenor_col         | Optional tenor-in-years column name.                                            |
| tenor_tag         | Character label appended to the carry column name.                              |
| long_threshold    | Numeric threshold above which carry is treated as positive enough to go long.   |
| short_threshold   | Numeric threshold below which carry is treated as negative enough to go short.  |
| target_size       | Numeric absolute target exposure.                                               |
| compute_features  | Logical; when 'TRUE', missing carry features are added to 'DT' in place.        |

|          |                                                                                                 |
|----------|-------------------------------------------------------------------------------------------------|
| strat_id | Integer strategy identifier recorded on generated actions.                                      |
| tol_pos  | Numeric tolerance passed to the action planner.                                                 |
| debug    | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A list produced by 'gen\_action\_plan\_rcpp()'.

---

|                                         |
|-----------------------------------------|
| strat_fx_carry_basket_rank_action_plan  |
| <i>FX-Carry-Basket-Rank Action Plan</i> |

---

**Description**

Applies the latest cross-sectional FX carry target weights to a portfolio state and converts them into a portfolio-adjustment table plus order intents.

**Usage**

```
strat_fx_carry_basket_rank_action_plan(  
  DT,  
  portfolio_state,  
  equity,  
  date_col = "date",  
  asset_col = "asset",  
  signal_col = "fx_carry",  
  long_n = 1L,  
  short_n = 1L,  
  gross_exposure = 1,  
  strat_id = 511L,  
  pricing_method = "market",  
  debug = FALSE  
)
```

**Arguments**

|                 |                                                                                    |
|-----------------|------------------------------------------------------------------------------------|
| DT              | A panel 'data.table' already summarized to one row per asset-date.                 |
| portfolio_state | A 'data.table' containing current per-asset portfolio state.                       |
| equity          | Numeric total portfolio equity used to scale target weights into target notionals. |
| date_col        | Date column used for cross-sectional grouping.                                     |
| asset_col       | Asset identifier column.                                                           |
| signal_col      | FX carry signal column used for ranking.                                           |
| long_n          | Integer number of assets to allocate long each date.                               |

|                |                                                                                                 |
|----------------|-------------------------------------------------------------------------------------------------|
| short_n        | Integer number of assets to allocate short each date.                                           |
| gross_exposure | Numeric gross exposure allocated across each side.                                              |
| strat_id       | Integer strategy identifier recorded on generated order intents.                                |
| pricing_method | Pricing method passed to 'build_order_intents()'.                                               |
| debug          | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A list with 'adjustment\_plan', 'order\_intents', and the latest target weights.

---

strat\_fx\_carry\_basket\_rank\_tgt\_pos  
*FX-Carry-Basket-Rank Target Positions*

---

**Description**

Generates per-row target weights by ranking an FX carry signal across assets within each date. Top-ranked assets receive positive weights and bottom-ranked assets receive negative weights.

**Usage**

```
strat_fx_carry_basket_rank_tgt_pos(
  DT,
  date_col = "date",
  asset_col = "asset",
  signal_col = "fx_carry",
  long_n = 1L,
  short_n = 1L,
  gross_exposure = 1,
  debug = FALSE
)
```

**Arguments**

|                |                                                                                                 |
|----------------|-------------------------------------------------------------------------------------------------|
| DT             | A panel 'data.table' already summarized to one row per asset-date.                              |
| date_col       | Date column used for cross-sectional grouping.                                                  |
| asset_col      | Asset identifier column.                                                                        |
| signal_col     | FX carry signal column used for ranking.                                                        |
| long_n         | Integer number of assets to allocate long each date.                                            |
| short_n        | Integer number of assets to allocate short each date.                                           |
| gross_exposure | Numeric gross exposure allocated across each side.                                              |
| debug          | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

strat\_fx\_carry\_tgt\_pos

*FX-Carry Target Positions*


---

**Description**

Generates a simple FX-carry target-position path. Positive carry targets a long exposure and negative carry targets a short exposure.

**Usage**

```
strat_fx_carry_tgt_pos(
  DT,
  domestic_rate_col = "r_domestic",
  foreign_rate_col = "r_foreign",
  tenor_col = NULL,
  tenor_tag = "1m",
  long_threshold = 0,
  short_threshold = 0,
  target_size = 1,
  compute_features = TRUE,
  debug = FALSE
)
```

**Arguments**

|                   |                                                                                                 |
|-------------------|-------------------------------------------------------------------------------------------------|
| DT                | A 'data.table' containing FX carry inputs or a precomputed 'fx_carry_*' column.                 |
| domestic_rate_col | Domestic annualized rate column name.                                                           |
| foreign_rate_col  | Foreign annualized rate column name.                                                            |
| tenor_col         | Optional tenor-in-years column name.                                                            |
| tenor_tag         | Character label appended to the carry column name.                                              |
| long_threshold    | Numeric threshold above which carry is treated as positive enough to go long.                   |
| short_threshold   | Numeric threshold below which carry is treated as negative enough to go short.                  |
| target_size       | Numeric absolute target exposure.                                                               |
| compute_features  | Logical; when 'TRUE', missing carry features are added to 'DT' in place.                        |
| debug             | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

strat\_fx\_carry\_trend\_action\_plan

*FX-Carry-Trend Action Plan*

---

**Description**

Applies the FX-carry-plus-trend rule to the latest row and translates the resulting target exposure into an executable action plan.

**Usage**

```
strat_fx_carry_trend_action_plan(
  DT,
  state,
  domestic_rate_col = "r_domestic",
  foreign_rate_col = "r_foreign",
  tenor_col = NULL,
  tenor_tag = "1m",
  trend_n = 50L,
  spot_col = "spot",
  long_threshold = 0,
  short_threshold = 0,
  target_size = 1,
  compute_features = TRUE,
  strat_id = 612L,
  tol_pos = 0.1,
  debug = FALSE
)
```

**Arguments**

|                   |                                                                                 |
|-------------------|---------------------------------------------------------------------------------|
| DT                | A 'data.table' containing FX carry inputs or a precomputed 'fx_carry_*' column. |
| state             | Named list describing the current trading state.                                |
| domestic_rate_col | Domestic annualized rate column name.                                           |
| foreign_rate_col  | Foreign annualized rate column name.                                            |
| tenor_col         | Optional tenor-in-years column name.                                            |
| tenor_tag         | Character label appended to the carry column name.                              |
| trend_n           | Integer EMA window used for the spot trend filter.                              |
| spot_col          | Spot-price column name.                                                         |

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| long_threshold   | Numeric threshold above which carry is treated as positive enough to go long.                   |
| short_threshold  | Numeric threshold below which carry is treated as negative enough to go short.                  |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing carry and EMA features are added to 'DT' in place.                |
| strat_id         | Integer strategy identifier recorded on generated actions.                                      |
| tol_pos          | Numeric tolerance passed to the action planner.                                                 |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

### Value

A list produced by 'gen\_action\_plan\_rcpp()'.

---

strat\_fx\_carry\_trend\_tgt\_pos  
*FX-Carry-Trend Target Positions*

---

### Description

Generates an FX target-position path that requires both carry and spot trend agreement. Positive carry only targets a long exposure when spot is above its EMA, while negative carry only targets a short exposure when spot is below its EMA.

### Usage

```
strat_fx_carry_trend_tgt_pos(
  DT,
  domestic_rate_col = "r_domestic",
  foreign_rate_col = "r_foreign",
  tenor_col = NULL,
  tenor_tag = "1m",
  trend_n = 50L,
  spot_col = "spot",
  long_threshold = 0,
  short_threshold = 0,
  target_size = 1,
  compute_features = TRUE,
  debug = FALSE
)
```

**Arguments**

|                   |                                                                                                 |
|-------------------|-------------------------------------------------------------------------------------------------|
| DT                | A 'data.table' containing FX carry inputs or a precomputed 'fx_carry_*' column.                 |
| domestic_rate_col | Domestic annualized rate column name.                                                           |
| foreign_rate_col  | Foreign annualized rate column name.                                                            |
| tenor_col         | Optional tenor-in-years column name.                                                            |
| tenor_tag         | Character label appended to the carry column name.                                              |
| trend_n           | Integer EMA window used for the spot trend filter.                                              |
| spot_col          | Spot-price column name.                                                                         |
| long_threshold    | Numeric threshold above which carry is treated as positive enough to go long.                   |
| short_threshold   | Numeric threshold below which carry is treated as negative enough to go short.                  |
| target_size       | Numeric absolute target exposure.                                                               |
| compute_features  | Logical; when 'TRUE', missing carry and EMA features are added to 'DT' in place.                |
| debug             | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

strat\_gamma\_scalp\_support\_action\_plan

*Gamma-Scalp-Support Action Plan*

---

**Description**

Applies the long-gamma support rule to the latest row, translates the resulting target exposure into an executable action plan, and optionally attaches a delta-hedge adjustment recommendation.

**Usage**

```
strat_gamma_scalp_support_action_plan(
  DT,
  state,
  rv_n = 20L,
  iv_col = "iv",
  annualization = 252,
  edge_threshold = 0,
  min_time_to_expiry = 5/252,
```

```

    max_time_to_expiry = 90/252,
    target_size = 1,
    compute_features = TRUE,
    strat_id = 708L,
    tol_pos = 0.1,
    current_delta = NULL,
    hedge_delta = NULL,
    target_delta = 0,
    debug = FALSE
)

```

### Arguments

|                    |                                                                                                       |
|--------------------|-------------------------------------------------------------------------------------------------------|
| DT                 | A 'data.table' containing 'close', time-to-expiry 'time_to_expiry', and an implied-volatility column. |
| state              | Named list describing the current trading state.                                                      |
| rv_n               | Integer realized-volatility window.                                                                   |
| iv_col             | Implied-volatility column name.                                                                       |
| annualization      | Numeric annualization factor passed to 'calc_realized_vol()'.<br>Default is 252.                      |
| edge_threshold     | Minimum realized-minus-implied volatility edge required to activate the long-gamma support position.  |
| min_time_to_expiry | Minimum time to expiry in years.                                                                      |
| max_time_to_expiry | Maximum time to expiry in years.                                                                      |
| target_size        | Numeric absolute target exposure.                                                                     |
| compute_features   | Logical; when 'TRUE', missing realized-volatility and gamma-edge features are added to 'DT' in place. |
| strat_id           | Integer strategy identifier recorded on generated actions.                                            |
| tol_pos            | Numeric tolerance passed to the action planner.                                                       |
| current_delta      | Optional current portfolio delta used for the hedge suggestion.                                       |
| hedge_delta        | Optional delta contribution per hedge unit used for the hedge suggestion.                             |
| target_delta       | Numeric target delta used for the hedge suggestion.                                                   |
| debug              | Logical; when 'TRUE', returns a list with the generated target vector and feature column names.       |

### Value

A list with an action plan and, when hedge inputs are supplied, a delta-hedge adjustment table.

---

strat\_gamma\_scalp\_support\_tgt\_pos

*Gamma-Scalp-Support Target Positions*


---

## Description

Generates a support target-position path for long-gamma structures. The strategy is active when realized volatility is sufficiently rich relative to implied volatility and option expiry remains within a tradable window.

## Usage

```
strat_gamma_scalp_support_tgt_pos(
  DT,
  rv_n = 20L,
  iv_col = "iv",
  annualization = 252,
  edge_threshold = 0,
  min_time_to_expiry = 5/252,
  max_time_to_expiry = 90/252,
  target_size = 1,
  compute_features = TRUE,
  debug = FALSE
)
```

## Arguments

|                    |                                                                                                       |
|--------------------|-------------------------------------------------------------------------------------------------------|
| DT                 | A 'data.table' containing 'close', time-to-expiry 'time_to_expiry', and an implied-volatility column. |
| rv_n               | Integer realized-volatility window.                                                                   |
| iv_col             | Implied-volatility column name.                                                                       |
| annualization      | Numeric annualization factor passed to 'calc_realized_vol()'.                                         |
| edge_threshold     | Minimum realized-minus-implied volatility edge required to activate the long-gamma support position.  |
| min_time_to_expiry | Minimum time to expiry in years.                                                                      |
| max_time_to_expiry | Maximum time to expiry in years.                                                                      |
| target_size        | Numeric absolute target exposure.                                                                     |
| compute_features   | Logical; when 'TRUE', missing realized-volatility and gamma-edge features are added to 'DT' in place. |
| debug              | Logical; when 'TRUE', returns a list with the generated target vector and feature column names.       |

Value

A numeric vector of target positions, or a list when ‘debug = TRUE’.

---

|                                           |
|-------------------------------------------|
| strat_iv_directional_overlay_action_plan  |
| <i>IV-Directional-Overlay Action Plan</i> |

---

Description

Applies the IV-skew directional-overlay rule to the latest summarized row and translates the resulting target exposure into an executable action plan.

Usage

```
strat_iv_directional_overlay_action_plan(  
  DT,  
  state,  
  date_col = "date",  
  expiry_col = "time_to_expiry",  
  type_col = "type",  
  moneyness_col = "option_log_forward_moneyness",  
  iv_col = "iv",  
  target_abs_moneyness = 0.1,  
  trend_col = "close",  
  trend_n = 50L,  
  skew_long_threshold = 0.02,  
  skew_short_threshold = -0.02,  
  overlay_mode = c("confirm", "flip"),  
  target_size = 1,  
  compute_features = TRUE,  
  strat_id = 710L,  
  tol_pos = 0.1,  
  debug = FALSE  
)
```

Arguments

|               |                                                                                                                                                                                                   |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DT            | An option-chain ‘data.table’ when ‘compute_features = TRUE’, or a summarized ‘data.table’ containing ‘iv_skew’, a trend-value column, and a ‘trend_ema_*’ column when ‘compute_features = FALSE’. |
| state         | Named list describing the current trading state.                                                                                                                                                  |
| date_col      | Date or timestamp column name.                                                                                                                                                                    |
| expiry_col    | Time-to-expiry column name.                                                                                                                                                                       |
| type_col      | Option-type column name.                                                                                                                                                                          |
| moneyness_col | Forward-moneyness feature column name.                                                                                                                                                            |

|                      |                                                                                         |
|----------------------|-----------------------------------------------------------------------------------------|
| iv_col               | Implied-volatility column name.                                                         |
| target_abs_moneyness | Numeric target absolute log-forward-moneyness used by 'calc_option_iv_skew()'.          |
| trend_col            | Underlying trend-value column name.                                                     |
| trend_n              | Integer EMA window used for the trend overlay.                                          |
| skew_long_threshold  | Numeric skew threshold above which the base signal is long.                             |
| skew_short_threshold | Numeric skew threshold below which the base signal is short.                            |
| overlay_mode         | Either "confirm" or "flip".                                                             |
| target_size          | Numeric absolute target exposure.                                                       |
| compute_features     | Logical; when 'TRUE', the chain is summarized before generating targets.                |
| strat_id             | Integer strategy identifier recorded on generated actions.                              |
| tol_pos              | Numeric tolerance passed to the action planner.                                         |
| debug                | Logical; when 'TRUE', returns a list with the generated target vector and summary data. |

**Value**

A list produced by 'gen\_action\_plan\_rcpp()'.

---

strat\_iv\_directional\_overlay\_tgt\_pos  
*IV-Directional-Overlay Target Positions*

---

**Description**

Generates a target-position path from option implied-volatility skew with a simple underlying-trend overlay. The base skew signal can either be confirmed by the underlying trend or flipped when the trend disagrees.

**Usage**

```
strat_iv_directional_overlay_tgt_pos(
  DT,
  date_col = "date",
  expiry_col = "time_to_expiry",
  type_col = "type",
  moneyness_col = "option_log_forward_moneyness",
  iv_col = "iv",
  target_abs_moneyness = 0.1,
  trend_col = "close",
  trend_n = 50L,
```

```

    skew_long_threshold = 0.02,
    skew_short_threshold = -0.02,
    overlay_mode = c("confirm", "flip"),
    target_size = 1,
    compute_features = TRUE,
    debug = FALSE
)

```

### Arguments

|                      |                                                                                                                                                                                                   |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DT                   | An option-chain 'data.table' when 'compute_features = TRUE', or a summarized 'data.table' containing 'iv_skew', a trend-value column, and a 'trend_ema_*' column when 'compute_features = FALSE'. |
| date_col             | Date or timestamp column name.                                                                                                                                                                    |
| expiry_col           | Time-to-expiry column name.                                                                                                                                                                       |
| type_col             | Option-type column name.                                                                                                                                                                          |
| moneyness_col        | Forward-moneyness feature column name.                                                                                                                                                            |
| iv_col               | Implied-volatility column name.                                                                                                                                                                   |
| target_abs_moneyness | Numeric target absolute log-forward-moneyness used by 'calc_option_iv_skew()'.                                                                                                                    |
| trend_col            | Underlying trend-value column name.                                                                                                                                                               |
| trend_n              | Integer EMA window used for the trend overlay.                                                                                                                                                    |
| skew_long_threshold  | Numeric skew threshold above which the base signal is long.                                                                                                                                       |
| skew_short_threshold | Numeric skew threshold below which the base signal is short.                                                                                                                                      |
| overlay_mode         | Either "confirm" or "flip".                                                                                                                                                                       |
| target_size          | Numeric absolute target exposure.                                                                                                                                                                 |
| compute_features     | Logical; when 'TRUE', the chain is summarized before generating targets.                                                                                                                          |
| debug                | Logical; when 'TRUE', returns a list with the generated target vector and summary data.                                                                                                           |

### Value

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

strat\_iv\_skew\_action\_plan

*IV-Skew Action Plan*


---

### Description

Applies the IV-skew rule to the latest summarized row and translates the resulting target exposure into an executable action plan.

### Usage

```
strat_iv_skew_action_plan(
  DT,
  state,
  date_col = "date",
  expiry_col = "time_to_expiry",
  type_col = "type",
  moneyness_col = "option_log_forward_moneyness",
  iv_col = "iv",
  target_abs_moneyness = 0.1,
  long_threshold = 0.02,
  short_threshold = -0.02,
  target_size = 1,
  compute_features = TRUE,
  strat_id = 701L,
  tol_pos = 0.1,
  debug = FALSE
)
```

### Arguments

|                      |                                                                                                                                                 |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| DT                   | An option-chain 'data.table' when 'compute_features = TRUE', or a summarized 'data.table' containing 'iv_skew' when 'compute_features = FALSE'. |
| state                | Named list describing the current trading state.                                                                                                |
| date_col             | Date or timestamp column name.                                                                                                                  |
| expiry_col           | Time-to-expiry column name.                                                                                                                     |
| type_col             | Option-type column name.                                                                                                                        |
| moneyness_col        | Forward-moneyness feature column name.                                                                                                          |
| iv_col               | Implied-volatility column name.                                                                                                                 |
| target_abs_moneyness | Numeric target absolute log-forward-moneyness used by 'calc_option_iv_skew()'.                                                                  |
| long_threshold       | Numeric threshold above which skew is treated as positive enough to go long.                                                                    |
| short_threshold      | Numeric threshold below which skew is treated as negative enough to go short.                                                                   |

|                  |                                                                                         |
|------------------|-----------------------------------------------------------------------------------------|
| target_size      | Numeric absolute target exposure.                                                       |
| compute_features | Logical; when 'TRUE', the chain is summarized before generating targets.                |
| strat_id         | Integer strategy identifier recorded on generated actions.                              |
| tol_pos          | Numeric tolerance passed to the action planner.                                         |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and summary data. |

**Value**

A list produced by 'gen\_action\_plan\_rcpp()'.

---

strat\_iv\_skew\_realized\_vol\_confirm\_action\_plan  
*IV-Skew-Realized-Vol-Confirm Action Plan*

---

**Description**

Applies the IV-skew with realized-volatility confirmation rule to the latest summarized row and translates the resulting target exposure into an executable action plan.

**Usage**

```
strat_iv_skew_realized_vol_confirm_action_plan(
  DT,
  state,
  date_col = "date",
  expiry_col = "time_to_expiry",
  type_col = "type",
  moneyness_col = "option_log_forward_moneyness",
  iv_col = "iv",
  target_abs_moneyness = 0.1,
  close_col = "close",
  rv_n = 20L,
  annualization = 252,
  long_threshold = 0.02,
  short_threshold = -0.02,
  iv_rv_confirm = 0,
  target_size = 1,
  compute_features = TRUE,
  strat_id = 709L,
  tol_pos = 0.1,
  debug = FALSE
)
```

**Arguments**

|                      |                                                                                                                                                                           |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DT                   | An option-chain 'data.table' when 'compute_features = TRUE', or a summarized 'data.table' containing 'iv_skew' and 'iv_rv_spread_<rv_n>' when 'compute_features = FALSE'. |
| state                | Named list describing the current trading state.                                                                                                                          |
| date_col             | Date or timestamp column name.                                                                                                                                            |
| expiry_col           | Time-to-expiry column name.                                                                                                                                               |
| type_col             | Option-type column name.                                                                                                                                                  |
| moneyness_col        | Forward-moneyness feature column name.                                                                                                                                    |
| iv_col               | Implied-volatility column name.                                                                                                                                           |
| target_abs_moneyness | Numeric target absolute log-forward-moneyness used by 'calc_option_iv_skew()'.                                                                                            |
| close_col            | Underlying close column name used to compute realized volatility.                                                                                                         |
| rv_n                 | Integer realized-volatility window.                                                                                                                                       |
| annualization        | Annualization factor passed to 'calc_realized_vol()'.                                                                                                                     |
| long_threshold       | Numeric threshold above which skew is treated as positive enough to go long.                                                                                              |
| short_threshold      | Numeric threshold below which skew is treated as negative enough to go short.                                                                                             |
| iv_rv_confirm        | Numeric minimum confirmation spread magnitude.                                                                                                                            |
| target_size          | Numeric absolute target exposure.                                                                                                                                         |
| compute_features     | Logical; when 'TRUE', the chain is summarized before generating targets.                                                                                                  |
| strat_id             | Integer strategy identifier recorded on generated actions.                                                                                                                |
| tol_pos              | Numeric tolerance passed to the action planner.                                                                                                                           |
| debug                | Logical; when 'TRUE', returns a list with the generated target vector and summary data.                                                                                   |

**Value**

A list produced by 'gen\_action\_plan\_rcpp()'.

---

strat\_iv\_skew\_realized\_vol\_confirm\_tgt\_pos

*IV-Skew-Realized-Vol-Confirm Target Positions*

---

**Description**

Generates a target-position path from option implied-volatility skew, gated by an implied-minus-realized volatility confirmation spread. Positive skew is actionable only when summarized implied volatility remains sufficiently above realized volatility; negative skew is actionable only when the confirmation spread is sufficiently negative.

**Usage**

```

strat_iv_skew_realized_vol_confirm_tgt_pos(
  DT,
  date_col = "date",
  expiry_col = "time_to_expiry",
  type_col = "type",
  moneyness_col = "option_log_forward_moneyness",
  iv_col = "iv",
  target_abs_moneyness = 0.1,
  close_col = "close",
  rv_n = 20L,
  annualization = 252,
  long_threshold = 0.02,
  short_threshold = -0.02,
  iv_rv_confirm = 0,
  target_size = 1,
  compute_features = TRUE,
  debug = FALSE
)

```

**Arguments**

|                      |                                                                                                                                                                           |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DT                   | An option-chain 'data.table' when 'compute_features = TRUE', or a summarized 'data.table' containing 'iv_skew' and 'iv_rv_spread_<rv_n>' when 'compute_features = FALSE'. |
| date_col             | Date or timestamp column name.                                                                                                                                            |
| expiry_col           | Time-to-expiry column name.                                                                                                                                               |
| type_col             | Option-type column name.                                                                                                                                                  |
| moneyness_col        | Forward-moneyness feature column name.                                                                                                                                    |
| iv_col               | Implied-volatility column name.                                                                                                                                           |
| target_abs_moneyness | Numeric target absolute log-forward-moneyness used by 'calc_option_iv_skew()'.                                                                                            |
| close_col            | Underlying close column name used to compute realized volatility.                                                                                                         |
| rv_n                 | Integer realized-volatility window.                                                                                                                                       |
| annualization        | Annualization factor passed to 'calc_realized_vol()'.                                                                                                                     |
| long_threshold       | Numeric threshold above which skew is treated as positive enough to go long.                                                                                              |
| short_threshold      | Numeric threshold below which skew is treated as negative enough to go short.                                                                                             |
| iv_rv_confirm        | Numeric minimum confirmation spread magnitude.                                                                                                                            |
| target_size          | Numeric absolute target exposure.                                                                                                                                         |
| compute_features     | Logical; when 'TRUE', the chain is summarized before generating targets.                                                                                                  |
| debug                | Logical; when 'TRUE', returns a list with the generated target vector and summary data.                                                                                   |

**Value**

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

strat\_iv\_skew\_tgt\_pos *IV-Skew Target Positions*

---

**Description**

Generates a simple target-position path from option implied-volatility skew. Positive skew above the long threshold targets a long structure exposure, and negative skew below the short threshold targets a short structure exposure.

**Usage**

```
strat_iv_skew_tgt_pos(
  DT,
  date_col = "date",
  expiry_col = "time_to_expiry",
  type_col = "type",
  moneyness_col = "option_log_forward_moneyness",
  iv_col = "iv",
  target_abs_moneyness = 0.1,
  long_threshold = 0.02,
  short_threshold = -0.02,
  target_size = 1,
  compute_features = TRUE,
  debug = FALSE
)
```

**Arguments**

|                      |                                                                                                                                                 |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| DT                   | An option-chain 'data.table' when 'compute_features = TRUE', or a summarized 'data.table' containing 'iv_skew' when 'compute_features = FALSE'. |
| date_col             | Date or timestamp column name.                                                                                                                  |
| expiry_col           | Time-to-expiry column name.                                                                                                                     |
| type_col             | Option-type column name.                                                                                                                        |
| moneyness_col        | Forward-moneyness feature column name.                                                                                                          |
| iv_col               | Implied-volatility column name.                                                                                                                 |
| target_abs_moneyness | Numeric target absolute log-forward-moneyness used by 'calc_option_iv_skew()'.                                                                  |
| long_threshold       | Numeric threshold above which skew is treated as positive enough to go long.                                                                    |
| short_threshold      | Numeric threshold below which skew is treated as negative enough to go short.                                                                   |
| target_size          | Numeric absolute target exposure.                                                                                                               |

|                  |                                                                                         |
|------------------|-----------------------------------------------------------------------------------------|
| compute_features | Logical; when 'TRUE', the chain is summarized before generating targets.                |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and summary data. |

**Value**

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

|                                    |
|------------------------------------|
| strat_iv_skew_zscore_action_plan   |
| <i>IV-Skew-Z-Score Action Plan</i> |

---

**Description**

Applies the IV-skew z-score rule to the latest summarized row and translates the resulting target exposure into an executable action plan.

**Usage**

```
strat_iv_skew_zscore_action_plan(  
  DT,  
  state,  
  date_col = "date",  
  expiry_col = "time_to_expiry",  
  type_col = "type",  
  moneyness_col = "option_log_forward_moneyness",  
  iv_col = "iv",  
  target_abs_moneyness = 0.1,  
  z_n = 60L,  
  long_z = 1,  
  short_z = -1,  
  mode = c("momentum", "reversion"),  
  target_size = 1,  
  compute_features = TRUE,  
  strat_id = 706L,  
  tol_pos = 0.1,  
  debug = FALSE  
)
```

**Arguments**

|          |                                                                                                                                                                        |
|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DT       | An option-chain 'data.table' when 'compute_features = TRUE', or a summarized 'data.table' containing 'iv_skew' and 'zscore_iv_skew_*' when 'compute_features = FALSE'. |
| state    | Named list describing the current trading state.                                                                                                                       |
| date_col | Date or timestamp column name.                                                                                                                                         |

|                      |                                                                                                                                                            |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| expiry_col           | Time-to-expiry column name.                                                                                                                                |
| type_col             | Option-type column name.                                                                                                                                   |
| moneyiness_col       | Forward-moneyness feature column name.                                                                                                                     |
| iv_col               | Implied-volatility column name.                                                                                                                            |
| target_abs_moneyness | Numeric target absolute log-forward-moneyness used by 'calc_option_iv_skew()'.<br>Logical; when 'TRUE', the chain is summarized before generating targets. |
| z_n                  | Integer rolling z-score window.                                                                                                                            |
| long_z               | Numeric upper z-score threshold.                                                                                                                           |
| short_z              | Numeric lower z-score threshold.                                                                                                                           |
| mode                 | Character strategy mode: 'momentum' or 'reversion'.                                                                                                        |
| target_size          | Numeric absolute target exposure.                                                                                                                          |
| compute_features     | Logical; when 'TRUE', the chain is summarized before generating targets.                                                                                   |
| strat_id             | Integer strategy identifier recorded on generated actions.                                                                                                 |
| tol_pos              | Numeric tolerance passed to the action planner.                                                                                                            |
| debug                | Logical; when 'TRUE', returns a list with the generated target vector and summary data.                                                                    |

**Value**

A list produced by 'gen\_action\_plan\_rcpp()'.

---

strat\_iv\_skew\_zscore\_tgt\_pos  
*IV-Skew-Z-Score Target Positions*

---

**Description**

Generates a target-position path from the rolling z-score of option implied-volatility skew, with either momentum or reversion interpretation.

**Usage**

```
strat_iv_skew_zscore_tgt_pos(
  DT,
  date_col = "date",
  expiry_col = "time_to_expiry",
  type_col = "type",
  moneyiness_col = "option_log_forward_moneyness",
  iv_col = "iv",
  target_abs_moneyness = 0.1,
  z_n = 60L,
  long_z = 1,
```

```

    short_z = -1,
    mode = c("momentum", "reversion"),
    target_size = 1,
    compute_features = TRUE,
    debug = FALSE
  )

```

### Arguments

|                      |                                                                                                                                                                        |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DT                   | An option-chain 'data.table' when 'compute_features = TRUE', or a summarized 'data.table' containing 'iv_skew' and 'zscore_iv_skew_*' when 'compute_features = FALSE'. |
| date_col             | Date or timestamp column name.                                                                                                                                         |
| expiry_col           | Time-to-expiry column name.                                                                                                                                            |
| type_col             | Option-type column name.                                                                                                                                               |
| moneyness_col        | Forward-moneyness feature column name.                                                                                                                                 |
| iv_col               | Implied-volatility column name.                                                                                                                                        |
| target_abs_moneyness | Numeric target absolute log-forward-moneyness used by 'calc_option_iv_skew()'.                                                                                         |
| z_n                  | Integer rolling z-score window.                                                                                                                                        |
| long_z               | Numeric upper z-score threshold.                                                                                                                                       |
| short_z              | Numeric lower z-score threshold.                                                                                                                                       |
| mode                 | Character strategy mode: 'momentum' or 'reversion'.                                                                                                                    |
| target_size          | Numeric absolute target exposure.                                                                                                                                      |
| compute_features     | Logical; when 'TRUE', the chain is summarized before generating targets.                                                                                               |
| debug                | Logical; when 'TRUE', returns a list with the generated target vector and summary data.                                                                                |

### Value

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

strat\_iv\_term\_structure\_action\_plan

*IV-Term-Structure Action Plan*

---

### Description

Applies the IV-term-structure rule to the latest summarized row and translates the resulting target exposure into an executable action plan.

**Usage**

```
strat_iv_term_structure_action_plan(
  DT,
  state,
  date_col = "date",
  expiry_col = "time_to_expiry",
  moneyness_col = "option_log_forward_moneyness",
  iv_col = "iv",
  long_threshold = 0.01,
  short_threshold = -0.01,
  target_size = 1,
  compute_features = TRUE,
  strat_id = 702L,
  tol_pos = 0.1,
  debug = FALSE
)
```

**Arguments**

|                  |                                                                                                                                                           |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| DT               | An option-chain ‘data.table’ when ‘compute_features = TRUE’, or a summarized ‘data.table’ containing ‘iv_term_structure’ when ‘compute_features = FALSE’. |
| state            | Named list describing the current trading state.                                                                                                          |
| date_col         | Date or timestamp column name.                                                                                                                            |
| expiry_col       | Time-to-expiry column name.                                                                                                                               |
| moneyness_col    | Forward-moneyness feature column name.                                                                                                                    |
| iv_col           | Implied-volatility column name.                                                                                                                           |
| long_threshold   | Numeric threshold above which term structure is treated as positive enough to go long.                                                                    |
| short_threshold  | Numeric threshold below which term structure is treated as negative enough to go short.                                                                   |
| target_size      | Numeric absolute target exposure.                                                                                                                         |
| compute_features | Logical; when ‘TRUE’, the chain is summarized before generating targets.                                                                                  |
| strat_id         | Integer strategy identifier recorded on generated actions.                                                                                                |
| tol_pos          | Numeric tolerance passed to the action planner.                                                                                                           |
| debug            | Logical; when ‘TRUE’, returns a list with the generated target vector and summary data.                                                                   |

**Value**

A list produced by ‘gen\_action\_plan\_rcpp()’.

---

strat\_iv\_term\_structure\_tgt\_pos

*IV-Term-Structure Target Positions*


---

## Description

Generates a simple target-position path from option ATM implied-volatility term structure. Positive term-structure slope above the long threshold targets a long structure exposure, and negative slope below the short threshold targets a short exposure.

## Usage

```
strat_iv_term_structure_tgt_pos(
  DT,
  date_col = "date",
  expiry_col = "time_to_expiry",
  moneyness_col = "option_log_forward_moneyness",
  iv_col = "iv",
  long_threshold = 0.01,
  short_threshold = -0.01,
  target_size = 1,
  compute_features = TRUE,
  debug = FALSE
)
```

## Arguments

|                  |                                                                                                                                                           |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| DT               | An option-chain ‘data.table’ when ‘compute_features = TRUE’, or a summarized ‘data.table’ containing ‘iv_term_structure’ when ‘compute_features = FALSE’. |
| date_col         | Date or timestamp column name.                                                                                                                            |
| expiry_col       | Time-to-expiry column name.                                                                                                                               |
| moneyness_col    | Forward-moneyness feature column name.                                                                                                                    |
| iv_col           | Implied-volatility column name.                                                                                                                           |
| long_threshold   | Numeric threshold above which term structure is treated as positive enough to go long.                                                                    |
| short_threshold  | Numeric threshold below which term structure is treated as negative enough to go short.                                                                   |
| target_size      | Numeric absolute target exposure.                                                                                                                         |
| compute_features | Logical; when ‘TRUE’, the chain is summarized before generating targets.                                                                                  |
| debug            | Logical; when ‘TRUE’, returns a list with the generated target vector and summary data.                                                                   |

Value

A numeric vector of target positions, or a list when ‘debug = TRUE’.

---

|                                  |
|----------------------------------|
| strat_ladder_bounce_action_plan  |
| <i>Ladder Bounce Action Plan</i> |

---

Description

Applies the naive ladder-bounce rule to the latest bar and translates the resulting target exposure into an executable action plan.

Usage

```
strat_ladder_bounce_action_plan(  
  DT,  
  state,  
  cycle_N = 180L,  
  span = 3L,  
  refined = TRUE,  
  min_swing = 0.05,  
  center_idx = 9L,  
  lower = 7L,  
  upper = 13L,  
  target_size = 1,  
  compute_ladder = TRUE,  
  detailed_report = FALSE,  
  strat_id = 201L,  
  tol_pos = 0.1,  
  debug = FALSE  
)
```

Arguments

|            |                                                                                                               |
|------------|---------------------------------------------------------------------------------------------------------------|
| DT         | A candle ‘data.table’. When ‘compute_ladder = TRUE’, it must contain ‘date-time’, ‘high’, ‘low’, and ‘close’. |
| state      | Named list describing the current trading state.                                                              |
| cycle_N    | Integer cycle width passed to ‘calc_ladder_index()’.                                                          |
| span       | Integer pivot span passed to ‘calc_ladder_index()’.                                                           |
| refined    | Logical pivot refinement flag.                                                                                |
| min_swing  | Minimum swing threshold passed to ‘calc_ladder_index()’.                                                      |
| center_idx | Zero-based native center index for ladder mapping.                                                            |
| lower      | Integer lower ladder threshold.                                                                               |
| upper      | Integer upper ladder threshold.                                                                               |

|                 |                                                                             |
|-----------------|-----------------------------------------------------------------------------|
| target_size     | Numeric absolute target exposure.                                           |
| compute_ladder  | Logical; compute the ladder index when missing.                             |
| detailed_report | Logical; forwarded to 'calc_ladder_index()'.                                |
| strat_id        | Integer strategy identifier recorded on generated actions.                  |
| tol_pos         | Numeric tolerance passed to the action planner.                             |
| debug           | Logical; when 'TRUE', returns a list with the generated target column name. |

**Value**

A list produced by 'gen\_action\_plan\_rcpp()'.

---

strat\_ladder\_bounce\_tgt\_pos

*Ladder Bounce Target Positions*

---

**Description**

Generates a naive mean-reversion target-position path from signed Fibonacci ladder indices. The rule is intentionally simple and public: touching the lower outer threshold targets a bounce upward, while touching the upper outer threshold targets a bounce downward.

**Usage**

```
strat_ladder_bounce_tgt_pos(
  DT,
  cycle_N = 180L,
  span = 3L,
  refined = TRUE,
  min_swing = 0.05,
  center_idx = 9L,
  lower = 7L,
  upper = 13L,
  target_size = 1,
  compute_ladder = TRUE,
  detailed_report = FALSE,
  debug = FALSE
)
```

**Arguments**

|         |                                                                                                               |
|---------|---------------------------------------------------------------------------------------------------------------|
| DT      | A candle 'data.table'. When 'compute_ladder = TRUE', it must contain 'date-time', 'high', 'low', and 'close'. |
| cycle_N | Integer cycle width passed to 'calc_ladder_index()'.                                                          |
| span    | Integer pivot span passed to 'calc_ladder_index()'.                                                           |

|                 |                                                                             |
|-----------------|-----------------------------------------------------------------------------|
| refined         | Logical pivot refinement flag.                                              |
| min_swing       | Minimum swing threshold passed to 'calc_ladder_index()'.                    |
| center_idx      | Zero-based native center index for ladder mapping.                          |
| lower           | Integer lower ladder threshold.                                             |
| upper           | Integer upper ladder threshold.                                             |
| target_size     | Numeric absolute target exposure.                                           |
| compute_ladder  | Logical; compute the ladder index when missing.                             |
| detailed_report | Logical; forwarded to 'calc_ladder_index()'.                                |
| debug           | Logical; when 'TRUE', returns a list with the generated target column name. |

**Value**

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

strat\_ladder\_breakout\_action\_plan

*Ladder Breakout Action Plan*

---

**Description**

Applies the naive ladder-breakout rule to the latest bar and translates the resulting target exposure into an executable action plan.

**Usage**

```
strat_ladder_breakout_action_plan(
  DT,
  state,
  cycle_N = 180L,
  span = 3L,
  refined = TRUE,
  min_swing = 0.05,
  center_idx = 9L,
  lower = 7L,
  upper = 13L,
  target_size = 1,
  compute_ladder = TRUE,
  detailed_report = FALSE,
  strat_id = 202L,
  tol_pos = 0.1,
  debug = FALSE
)
```

**Arguments**

|                 |                                                                                                               |
|-----------------|---------------------------------------------------------------------------------------------------------------|
| DT              | A candle 'data.table'. When 'compute_ladder = TRUE', it must contain 'date-time', 'high', 'low', and 'close'. |
| state           | Named list describing the current trading state.                                                              |
| cycle_N         | Integer cycle width passed to 'calc_ladder_index()'.                                                          |
| span            | Integer pivot span passed to 'calc_ladder_index()'.                                                           |
| refined         | Logical pivot refinement flag.                                                                                |
| min_swing       | Minimum swing threshold passed to 'calc_ladder_index()'.                                                      |
| center_idx      | Zero-based native center index for ladder mapping.                                                            |
| lower           | Integer lower ladder threshold.                                                                               |
| upper           | Integer upper ladder threshold.                                                                               |
| target_size     | Numeric absolute target exposure.                                                                             |
| compute_ladder  | Logical; compute the ladder index when missing.                                                               |
| detailed_report | Logical; forwarded to 'calc_ladder_index()'.                                                                  |
| strat_id        | Integer strategy identifier recorded on generated actions.                                                    |
| tol_pos         | Numeric tolerance passed to the action planner.                                                               |
| debug           | Logical; when 'TRUE', returns a list with the generated target column name.                                   |

**Value**

A list produced by 'gen\_action\_plan\_rcpp()'.

---

strat\_ladder\_breakout\_tgt\_pos

*Ladder Breakout Target Positions*

---

**Description**

Generates a naive continuation target-position path from signed Fibonacci ladder indices. The public rule is intentionally simple: positions beyond the lower threshold target further downside, and positions beyond the upper threshold target further upside.

**Usage**

```
strat_ladder_breakout_tgt_pos(
  DT,
  cycle_N = 180L,
  span = 3L,
  refined = TRUE,
  min_swing = 0.05,
  center_idx = 9L,
```

```
lower = 7L,  
upper = 13L,  
target_size = 1,  
compute_ladder = TRUE,  
detailed_report = FALSE,  
debug = FALSE  
)
```

Arguments

|                 |                                                                                                               |
|-----------------|---------------------------------------------------------------------------------------------------------------|
| DT              | A candle 'data.table'. When 'compute_ladder = TRUE', it must contain 'date-time', 'high', 'low', and 'close'. |
| cycle_N         | Integer cycle width passed to 'calc_ladder_index()'.                                                          |
| span            | Integer pivot span passed to 'calc_ladder_index()'.                                                           |
| refined         | Logical pivot refinement flag.                                                                                |
| min_swing       | Minimum swing threshold passed to 'calc_ladder_index()'.                                                      |
| center_idx      | Zero-based native center index for ladder mapping.                                                            |
| lower           | Integer lower ladder threshold.                                                                               |
| upper           | Integer upper ladder threshold.                                                                               |
| target_size     | Numeric absolute target exposure.                                                                             |
| compute_ladder  | Logical; compute the ladder index when missing.                                                               |
| detailed_report | Logical; forwarded to 'calc_ladder_index()'.                                                                  |
| debug           | Logical; when 'TRUE', returns a list with the generated target column name.                                   |

Value

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

|                                    |
|------------------------------------|
| strat_macd_contrarian_action_plan  |
| <i>MACD-Contrarian Action Plan</i> |

---

Description

Applies the MACD-contrarian rule to the latest bar and translates the resulting target exposure into an executable action plan.

Usage

```
strat_macd_contrarian_action_plan(  
  DT,  
  state,  
  fast = 12L,  
  slow = 26L,  
  signal = 9L,  
  target_size = 1,  
  compute_features = TRUE,  
  strat_id = 305L,  
  tol_pos = 0.1,  
  debug = FALSE  
)
```

Arguments

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle 'data.table'.                                                                          |
| state            | Named list describing the current trading state.                                                |
| fast             | Integer fast EMA window.                                                                        |
| slow             | Integer slow EMA window.                                                                        |
| signal           | Integer MACD signal EMA window.                                                                 |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing MACD features are added to 'DT' in place.                         |
| strat_id         | Integer strategy identifier recorded on generated actions.                                      |
| tol_pos          | Numeric tolerance passed to the action planner.                                                 |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

Value

A list produced by 'gen\_action\_plan\_rcpp()'.

---

|                                         |
|-----------------------------------------|
| strat_macd_contrarian_tgt_pos           |
| <i>MACD-Contrarian Target Positions</i> |

---

Description

Generates a target-position path that is the exact directional inverse of 'strat\_macd\_cross\_tgt\_pos()'. Positive MACD spread targets a short exposure, and negative MACD spread targets a long exposure.

Usage

```
strat_macd_contrarian_tgt_pos(  
  DT,  
  fast = 12L,  
  slow = 26L,  
  signal = 9L,  
  target_size = 1,  
  compute_features = TRUE,  
  debug = FALSE  
)
```

Arguments

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle 'data.table'.                                                                          |
| fast             | Integer fast EMA window.                                                                        |
| slow             | Integer slow EMA window.                                                                        |
| signal           | Integer MACD signal EMA window.                                                                 |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing MACD features are added to 'DT' in place.                         |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

Value

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

|                               |
|-------------------------------|
| strat_macd_cross_action_plan  |
| <i>MACD-Cross Action Plan</i> |

---

Description

Applies the MACD crossover rule to the latest bar and translates the resulting target exposure into an executable action plan.

Usage

```
strat_macd_cross_action_plan(  
  DT,  
  state,  
  fast = 12L,  
  slow = 26L,  
  signal = 9L,  
  target_size = 1,
```

```
compute_features = TRUE,  
strat_id = 304L,  
tol_pos = 0.1,  
debug = FALSE  
)
```

**Arguments**

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle 'data.table'.                                                                          |
| state            | Named list describing the current trading state.                                                |
| fast             | Integer fast EMA window.                                                                        |
| slow             | Integer slow EMA window.                                                                        |
| signal           | Integer MACD signal EMA window.                                                                 |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing MACD features are added to 'DT' in place.                         |
| strat_id         | Integer strategy identifier recorded on generated actions.                                      |
| tol_pos          | Numeric tolerance passed to the action planner.                                                 |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A list produced by 'gen\_action\_plan\_rcpp()'.

---

|                                    |
|------------------------------------|
| strat_macd_cross_tgt_pos           |
| <i>MACD-Cross Target Positions</i> |

---

**Description**

Generates a simple target-position path from MACD and MACD-signal line crossovers. Positive MACD spread targets a long exposure, and negative MACD spread targets a short exposure.

**Usage**

```
strat_macd_cross_tgt_pos(  
  DT,  
  fast = 12L,  
  slow = 26L,  
  signal = 9L,  
  target_size = 1,  
  compute_features = TRUE,  
  debug = FALSE  
)
```

**Arguments**

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle 'data.table'.                                                                          |
| fast             | Integer fast EMA window.                                                                        |
| slow             | Integer slow EMA window.                                                                        |
| signal           | Integer MACD signal EMA window.                                                                 |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing MACD features are added to 'DT' in place.                         |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

strat\_macd\_histogram\_momentum\_action\_plan  
*MACD-Histogram-Momentum Action Plan*

---

**Description**

Applies the histogram-acceleration MACD rule to the latest bar and translates the resulting target exposure into an executable action plan.

**Usage**

```
strat_macd_histogram_momentum_action_plan(
  DT,
  state,
  fast = 12L,
  slow = 26L,
  signal = 9L,
  accel_lag = 1L,
  target_size = 1,
  compute_features = TRUE,
  strat_id = 314L,
  tol_pos = 0.1,
  debug = FALSE
)
```

**Arguments**

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle 'data.table'.                                                                          |
| state            | Named list describing the current trading state.                                                |
| fast             | Integer fast EMA window.                                                                        |
| slow             | Integer slow EMA window.                                                                        |
| signal           | Integer MACD signal EMA window.                                                                 |
| accel_lag        | Integer lag used to measure histogram acceleration.                                             |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing MACD features are added to 'DT' in place.                         |
| strat_id         | Integer strategy identifier recorded on generated actions.                                      |
| tol_pos          | Numeric tolerance passed to the action planner.                                                 |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A list produced by 'gen\_action\_plan\_rcpp()'.

---

strat\_macd\_histogram\_momentum\_tgt\_pos

*MACD-Histogram-Momentum Target Positions*

---

**Description**

Generates a target-position path from MACD histogram direction and acceleration. Long exposure is targeted when the histogram is positive and still rising. Short exposure is targeted when the histogram is negative and still falling.

**Usage**

```
strat_macd_histogram_momentum_tgt_pos(
  DT,
  fast = 12L,
  slow = 26L,
  signal = 9L,
  accel_lag = 1L,
  target_size = 1,
  compute_features = TRUE,
  debug = FALSE
)
```

**Arguments**

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle 'data.table'.                                                                          |
| fast             | Integer fast EMA window.                                                                        |
| slow             | Integer slow EMA window.                                                                        |
| signal           | Integer MACD signal EMA window.                                                                 |
| accel_lag        | Integer lag used to measure histogram acceleration.                                             |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing MACD features are added to 'DT' in place.                         |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

strat\_macd\_zero\_line\_action\_plan  
*MACD-Zero-Line Action Plan*

---

**Description**

Applies the MACD-plus-zero-line rule to the latest bar and translates the resulting target exposure into an executable action plan.

**Usage**

```
strat_macd_zero_line_action_plan(
  DT,
  state,
  fast = 12L,
  slow = 26L,
  signal = 9L,
  target_size = 1,
  compute_features = TRUE,
  strat_id = 310L,
  tol_pos = 0.1,
  debug = FALSE
)
```

**Arguments**

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle 'data.table'.                                                                          |
| state            | Named list describing the current trading state.                                                |
| fast             | Integer fast EMA window.                                                                        |
| slow             | Integer slow EMA window.                                                                        |
| signal           | Integer MACD signal EMA window.                                                                 |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing MACD features are added to 'DT' in place.                         |
| strat_id         | Integer strategy identifier recorded on generated actions.                                      |
| tol_pos          | Numeric tolerance passed to the action planner.                                                 |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A list produced by 'gen\_action\_plan\_rcpp()'.

---

strat\_macd\_zero\_line\_tgt\_pos

*MACD-Zero-Line Target Positions*

---

**Description**

Generates a target-position path from MACD crossovers filtered by the zero-line. Long targets require bullish crossover with MACD above zero, while short targets require bearish crossover with MACD below zero.

**Usage**

```
strat_macd_zero_line_tgt_pos(
  DT,
  fast = 12L,
  slow = 26L,
  signal = 9L,
  target_size = 1,
  compute_features = TRUE,
  debug = FALSE
)
```

**Arguments**

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle 'data.table'.                                                                          |
| fast             | Integer fast EMA window.                                                                        |
| slow             | Integer slow EMA window.                                                                        |
| signal           | Integer MACD signal EMA window.                                                                 |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing MACD features are added to 'DT' in place.                         |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

strat\_pair\_spread\_bollinger\_action\_plan  
*Pair-Spread-Bollinger Action Plan*

---

**Description**

Applies the pair-spread Bollinger reversion rule to the latest bar and translates the resulting target exposure into an executable action plan.

**Usage**

```
strat_pair_spread_bollinger_action_plan(
  DT,
  state,
  x_col = "close",
  y_col = "benchmark_close",
  n = 20L,
  k = 2,
  target_size = 1,
  compute_features = TRUE,
  strat_id = 505L,
  tol_pos = 0.1,
  debug = FALSE
)
```

**Arguments**

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A 'data.table' containing the traded and benchmark price columns.                               |
| state            | Named list describing the current trading state.                                                |
| x_col            | Traded asset price column.                                                                      |
| y_col            | Benchmark or paired asset price column.                                                         |
| n                | Integer Bollinger window.                                                                       |
| k                | Numeric Bollinger width multiplier.                                                             |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing spread and spread-band features are added to 'DT' in place.       |
| strat_id         | Integer strategy identifier recorded on generated actions.                                      |
| tol_pos          | Numeric tolerance passed to the action planner.                                                 |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A list produced by 'gen\_action\_plan\_rcpp()'.

---

strat\_pair\_spread\_bollinger\_tgt\_pos  
*Pair-Spread-Bollinger Target Positions*

---

**Description**

Generates a pair-spread mean-reversion target-position path using Bollinger bands on the arithmetic spread between the traded asset and benchmark.

**Usage**

```
strat_pair_spread_bollinger_tgt_pos(
  DT,
  x_col = "close",
  y_col = "benchmark_close",
  n = 20L,
  k = 2,
  target_size = 1,
  compute_features = TRUE,
  debug = FALSE
)
```

**Arguments**

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A 'data.table' containing the traded and benchmark price columns.                               |
| x_col            | Traded asset price column.                                                                      |
| y_col            | Benchmark or paired asset price column.                                                         |
| n                | Integer Bollinger window.                                                                       |
| k                | Numeric Bollinger width multiplier.                                                             |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing spread and spread-band features are added to 'DT' in place.       |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

strat\_pair\_spread\_half\_life\_revert\_action\_plan  
*Pair-Spread-Half-Life-Reversion Action Plan*

---

**Description**

Applies the pair-spread half-life mean-reversion rule to the latest bar and translates the resulting target exposure for the traded asset into an executable action plan.

**Usage**

```
strat_pair_spread_half_life_revert_action_plan(
  DT,
  state,
  x_col = "close",
  y_col = "benchmark_close",
  z_n = 20L,
  hl_n = 60L,
  max_half_life = 20,
  long_z = -1,
  short_z = 1,
  exit_z = 0,
  target_size = 1,
  compute_features = TRUE,
  sample = TRUE,
  strat_id = 510L,
  tol_pos = 0.1,
  debug = FALSE
)
```

**Arguments**

|                  |                                                                                                   |
|------------------|---------------------------------------------------------------------------------------------------|
| DT               | A 'data.table' containing the traded and benchmark price columns.                                 |
| state            | Named list describing the current trading state.                                                  |
| x_col            | Traded asset price column.                                                                        |
| y_col            | Benchmark or paired asset price column.                                                           |
| z_n              | Integer rolling window used for spread z-scores.                                                  |
| hl_n             | Integer rolling window used for the half-life proxy.                                              |
| max_half_life    | Numeric maximum half-life allowed for entries.                                                    |
| long_z           | Numeric lower z-score threshold used for long entries.                                            |
| short_z          | Numeric upper z-score threshold used for short entries.                                           |
| exit_z           | Numeric z-score threshold used for exits.                                                         |
| target_size      | Numeric absolute target exposure.                                                                 |
| compute_features | Logical; when 'TRUE', missing spread, z-score, and half-life features are added to 'DT' in place. |
| sample           | Logical; if 'TRUE', rolling z-score uses sample standard deviation.                               |
| strat_id         | Integer strategy identifier recorded on generated actions.                                        |
| tol_pos          | Numeric tolerance passed to the action planner.                                                   |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names.   |

**Value**

A list produced by 'gen\_action\_plan\_rcpp()'.

---

strat\_pair\_spread\_half\_life\_revert\_tgt\_pos

*Pair-Spread-Half-Life-Reversion Target Positions*

---

**Description**

Generates a mean-reversion target-position path for a traded asset series relative to a benchmark series, but only when a rolling half-life proxy suggests the spread is mean-reverting quickly enough.

**Usage**

```
strat_pair_spread_half_life_revert_tgt_pos(
  DT,
  x_col = "close",
  y_col = "benchmark_close",
  z_n = 20L,
  hl_n = 60L,
```

```
max_half_life = 20,  
long_z = -1,  
short_z = 1,  
exit_z = 0,  
target_size = 1,  
compute_features = TRUE,  
sample = TRUE,  
debug = FALSE  
)
```

Arguments

|                  |                                                                                                   |
|------------------|---------------------------------------------------------------------------------------------------|
| DT               | A ‘data.table’ containing the traded and benchmark price columns.                                 |
| x_col            | Traded asset price column.                                                                        |
| y_col            | Benchmark or paired asset price column.                                                           |
| z_n              | Integer rolling window used for spread z-scores.                                                  |
| hl_n             | Integer rolling window used for the half-life proxy.                                              |
| max_half_life    | Numeric maximum half-life allowed for entries.                                                    |
| long_z           | Numeric lower z-score threshold used for long entries.                                            |
| short_z          | Numeric upper z-score threshold used for short entries.                                           |
| exit_z           | Numeric z-score threshold used for exits.                                                         |
| target_size      | Numeric absolute target exposure.                                                                 |
| compute_features | Logical; when ‘TRUE’, missing spread, z-score, and half-life features are added to ‘DT’ in place. |
| sample           | Logical; if ‘TRUE’, rolling z-score uses sample standard deviation.                               |
| debug            | Logical; when ‘TRUE’, returns a list with the generated target vector and feature column names.   |

Value

A numeric vector of target positions, or a list when ‘debug = TRUE’.

---

|                                          |
|------------------------------------------|
| strat_pair_spread_revert_action_plan     |
| <i>Pair-Spread-Reversion Action Plan</i> |

---

Description

Applies the pair-spread mean-reversion rule to the latest bar and translates the resulting target exposure for the traded asset into an executable action plan.

**Usage**

```
strat_pair_spread_revert_action_plan(
  DT,
  state,
  x_col = "close",
  y_col = "benchmark_close",
  z_n = 20L,
  entry_z = 2,
  exit_z = 0.5,
  target_size = 1,
  compute_features = TRUE,
  sample = TRUE,
  strat_id = 501L,
  tol_pos = 0.1,
  debug = FALSE
)
```

**Arguments**

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A 'data.table' containing the traded and benchmark price columns.                               |
| state            | Named list describing the current trading state.                                                |
| x_col            | Traded asset price column.                                                                      |
| y_col            | Benchmark or paired asset price column.                                                         |
| z_n              | Integer rolling window used for spread z-scores.                                                |
| entry_z          | Numeric absolute z-score threshold used for entries.                                            |
| exit_z           | Numeric absolute z-score threshold used for exits.                                              |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing spread and z-score features are added to 'DT' in place.           |
| sample           | Logical; if 'TRUE', rolling z-score uses sample standard deviation.                             |
| strat_id         | Integer strategy identifier recorded on generated actions.                                      |
| tol_pos          | Numeric tolerance passed to the action planner.                                                 |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A list produced by 'gen\_action\_plan\_rcpp()'.

---

strat\_pair\_spread\_revert\_tgt\_pos

*Pair-Spread-Reversion Target Positions*


---

## Description

Generates a simple mean-reversion target-position path for a traded asset series relative to a benchmark series. The arithmetic spread is standardized with a rolling z-score and extreme positive spreads target a short exposure, while extreme negative spreads target a long exposure.

## Usage

```
strat_pair_spread_revert_tgt_pos(
  DT,
  x_col = "close",
  y_col = "benchmark_close",
  z_n = 20L,
  entry_z = 2,
  exit_z = 0.5,
  target_size = 1,
  compute_features = TRUE,
  sample = TRUE,
  debug = FALSE
)
```

## Arguments

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A 'data.table' containing the traded and benchmark price columns.                               |
| x_col            | Traded asset price column.                                                                      |
| y_col            | Benchmark or paired asset price column.                                                         |
| z_n              | Integer rolling window used for spread z-scores.                                                |
| entry_z          | Numeric absolute z-score threshold used for entries.                                            |
| exit_z           | Numeric absolute z-score threshold used for exits.                                              |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing spread and z-score features are added to 'DT' in place.           |
| sample           | Logical; if 'TRUE', rolling z-score uses sample standard deviation.                             |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

## Value

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

strat\_ratio\_revert\_action\_plan

*Ratio-Reversion Action Plan*


---

## Description

Applies the ratio mean-reversion rule to the latest bar and translates the resulting target exposure for the traded asset into an executable action plan.

## Usage

```
strat_ratio_revert_action_plan(
  DT,
  state,
  x_col = "close",
  y_col = "benchmark_close",
  z_n = 20L,
  entry_z = 2,
  exit_z = 0.5,
  target_size = 1,
  compute_features = TRUE,
  sample = TRUE,
  strat_id = 502L,
  tol_pos = 0.1,
  debug = FALSE
)
```

## Arguments

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A 'data.table' containing the traded and benchmark price columns.                               |
| state            | Named list describing the current trading state.                                                |
| x_col            | Traded asset price column.                                                                      |
| y_col            | Benchmark or paired asset price column.                                                         |
| z_n              | Integer rolling window used for spread z-scores.                                                |
| entry_z          | Numeric absolute z-score threshold used for entries.                                            |
| exit_z           | Numeric absolute z-score threshold used for exits.                                              |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing spread and z-score features are added to 'DT' in place.           |
| sample           | Logical; if 'TRUE', rolling z-score uses sample standard deviation.                             |
| strat_id         | Integer strategy identifier recorded on generated actions.                                      |
| tol_pos          | Numeric tolerance passed to the action planner.                                                 |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A list produced by ‘gen\_action\_plan\_rcpp()’.

---

|                                         |
|-----------------------------------------|
| strat_ratio_revert_tgt_pos              |
| <i>Ratio-Reversion Target Positions</i> |

---

**Description**

Generates a simple mean-reversion target-position path for a traded asset series relative to a benchmark series using the rolling z-score of the price ratio.

**Usage**

```
strat_ratio_revert_tgt_pos(  
  DT,  
  x_col = "close",  
  y_col = "benchmark_close",  
  z_n = 20L,  
  entry_z = 2,  
  exit_z = 0.5,  
  target_size = 1,  
  compute_features = TRUE,  
  sample = TRUE,  
  debug = FALSE  
)
```

**Arguments**

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A ‘data.table’ containing the traded and benchmark price columns.                               |
| x_col            | Traded asset price column.                                                                      |
| y_col            | Benchmark or paired asset price column.                                                         |
| z_n              | Integer rolling window used for spread z-scores.                                                |
| entry_z          | Numeric absolute z-score threshold used for entries.                                            |
| exit_z           | Numeric absolute z-score threshold used for exits.                                              |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when ‘TRUE’, missing spread and z-score features are added to ‘DT’ in place.           |
| sample           | Logical; if ‘TRUE’, rolling z-score uses sample standard deviation.                             |
| debug            | Logical; when ‘TRUE’, returns a list with the generated target vector and feature column names. |

**Value**

A numeric vector of target positions, or a list when ‘debug = TRUE’.

---

strat\_regime\_switch\_action\_plan

*Regime-Switching Action Plan*


---

## Description

Applies the regime-switching rule to the latest row and translates the resulting target exposure into an executable action plan.

## Usage

```
strat_regime_switch_action_plan(
  DT,
  state,
  fast = 20L,
  slow = 50L,
  adx_n = 14L,
  rv_n = 20L,
  bb_n = 20L,
  bb_k = 2,
  trend_adx_threshold = 25,
  revert_adx_threshold = 18,
  high_vol_threshold = 0.4,
  breadth_col = NULL,
  breadth_long_threshold = -Inf,
  breadth_short_threshold = Inf,
  annualization = 252,
  target_size = 1,
  compute_features = TRUE,
  strat_id = 105L,
  tol_pos = 0.1,
  debug = FALSE
)
```

## Arguments

|       |                                                              |
|-------|--------------------------------------------------------------|
| DT    | A candle 'data.table'.                                       |
| state | Named list describing the current trading state.             |
| fast  | Integer fast EMA window for the trend regime.                |
| slow  | Integer slow EMA window for the trend regime.                |
| adx_n | Integer ADX window.                                          |
| rv_n  | Integer realized-volatility window.                          |
| bb_n  | Integer Bollinger window for the reversion regime.           |
| bb_k  | Numeric Bollinger width multiplier for the reversion regime. |

|                         |                                                                                                 |
|-------------------------|-------------------------------------------------------------------------------------------------|
| trend_adx_threshold     | Numeric ADX threshold above which the trend regime is allowed.                                  |
| revert_adx_threshold    | Numeric ADX threshold below which the mean-reversion regime is allowed.                         |
| high_vol_threshold      | Maximum realized volatility allowed for active regimes.                                         |
| breadth_col             | Optional breadth filter column.                                                                 |
| breadth_long_threshold  | Minimum breadth value required for long trend states.                                           |
| breadth_short_threshold | Maximum breadth value required for short trend states.                                          |
| annualization           | Numeric annualization factor passed to 'calc_realized_vol()'.                                   |
| target_size             | Numeric absolute target exposure.                                                               |
| compute_features        | Logical; when 'TRUE', missing features are added to 'DT' in place.                              |
| strat_id                | Integer strategy identifier recorded on generated actions.                                      |
| tol_pos                 | Numeric tolerance passed to the action planner.                                                 |
| debug                   | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A list produced by 'gen\_action\_plan\_rcpp()'.

---

strat\_regime\_switch\_tgt\_pos

*Regime-Switching Target Positions*

---

**Description**

Generates a simple meta-strategy target-position path that switches among a trend rule, a mean-reversion rule, and flat exposure depending on ADX, realized volatility, and optionally a breadth filter.

**Usage**

```
strat_regime_switch_tgt_pos(
  DT,
  fast = 20L,
  slow = 50L,
  adx_n = 14L,
  rv_n = 20L,
  bb_n = 20L,
  bb_k = 2,
```

```

trend_adx_threshold = 25,
revert_adx_threshold = 18,
high_vol_threshold = 0.4,
breadth_col = NULL,
breadth_long_threshold = -Inf,
breadth_short_threshold = Inf,
annualization = 252,
target_size = 1,
compute_features = TRUE,
debug = FALSE
)

```

### Arguments

|                         |                                                                                                 |
|-------------------------|-------------------------------------------------------------------------------------------------|
| DT                      | A candle 'data.table'.                                                                          |
| fast                    | Integer fast EMA window for the trend regime.                                                   |
| slow                    | Integer slow EMA window for the trend regime.                                                   |
| adx_n                   | Integer ADX window.                                                                             |
| rv_n                    | Integer realized-volatility window.                                                             |
| bb_n                    | Integer Bollinger window for the reversion regime.                                              |
| bb_k                    | Numeric Bollinger width multiplier for the reversion regime.                                    |
| trend_adx_threshold     | Numeric ADX threshold above which the trend regime is allowed.                                  |
| revert_adx_threshold    | Numeric ADX threshold below which the mean-reversion regime is allowed.                         |
| high_vol_threshold      | Maximum realized volatility allowed for active regimes.                                         |
| breadth_col             | Optional breadth filter column.                                                                 |
| breadth_long_threshold  | Minimum breadth value required for long trend states.                                           |
| breadth_short_threshold | Maximum breadth value required for short trend states.                                          |
| annualization           | Numeric annualization factor passed to 'calc_realized_vol()'.                                   |
| target_size             | Numeric absolute target exposure.                                                               |
| compute_features        | Logical; when 'TRUE', missing features are added to 'DT' in place.                              |
| debug                   | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

### Value

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

strat\_relative\_strength\_action\_plan  
*Relative-Strength Action Plan*

---

## Description

Applies the relative-strength rule to the latest bar and translates the resulting target exposure for the traded asset into an executable action plan.

## Usage

```
strat_relative_strength_action_plan(
  DT,
  state,
  x_col = "close",
  y_col = "benchmark_close",
  n = 20L,
  long_threshold = 1.02,
  short_threshold = 0.98,
  target_size = 1,
  compute_features = TRUE,
  use_log = TRUE,
  strat_id = 503L,
  tol_pos = 0.1,
  debug = FALSE
)
```

## Arguments

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A 'data.table' containing the traded and benchmark price columns.                               |
| state            | Named list describing the current trading state.                                                |
| x_col            | Traded asset price column.                                                                      |
| y_col            | Benchmark price column.                                                                         |
| n                | Integer relative-strength lookback window.                                                      |
| long_threshold   | Numeric threshold above which the traded asset is considered relatively strong.                 |
| short_threshold  | Numeric threshold below which the traded asset is considered relatively weak.                   |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing relative-strength features are added to 'DT' in place.            |
| use_log          | Logical; if 'TRUE', relative strength is based on log-return aggregation.                       |
| strat_id         | Integer strategy identifier recorded on generated actions.                                      |
| tol_pos          | Numeric tolerance passed to the action planner.                                                 |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

Value

A list produced by 'gen\_action\_plan\_rcpp()'.

---

|                                                    |
|----------------------------------------------------|
| strat_relative_strength_dual_momentum_action_plan  |
| <i>Relative-Strength-Dual-Momentum Action Plan</i> |

---

Description

Applies the dual-momentum relative-strength rule to the latest bar and translates the resulting target exposure into an executable action plan.

Usage

```
strat_relative_strength_dual_momentum_action_plan(  
  DT,  
  state,  
  x_col = "close",  
  y_col = "benchmark_close",  
  rs_n = 20L,  
  mom_n = 60L,  
  rs_long_threshold = 1,  
  rs_short_threshold = 1,  
  mom_long_threshold = 0,  
  mom_short_threshold = 0,  
  allow_short = TRUE,  
  target_size = 1,  
  compute_features = TRUE,  
  use_log = TRUE,  
  strat_id = 504L,  
  tol_pos = 0.1,  
  debug = FALSE  
)
```

Arguments

|                   |                                                                                                   |
|-------------------|---------------------------------------------------------------------------------------------------|
| DT                | A 'data.table' containing the traded and benchmark price columns.                                 |
| state             | Named list describing the current trading state.                                                  |
| x_col             | Traded asset price column.                                                                        |
| y_col             | Benchmark price column.                                                                           |
| rs_n              | Integer relative-strength lookback window.                                                        |
| mom_n             | Integer absolute-momentum lookback window.                                                        |
| rs_long_threshold | Numeric threshold above which the traded asset is considered relatively strong enough to go long. |

|                     |                                                                                                  |
|---------------------|--------------------------------------------------------------------------------------------------|
| rs_short_threshold  | Numeric threshold below which the traded asset is considered relatively weak enough to go short. |
| mom_long_threshold  | Numeric momentum threshold required for long exposure.                                           |
| mom_short_threshold | Numeric momentum threshold required for short exposure.                                          |
| allow_short         | Logical; if 'TRUE', allows short targets.                                                        |
| target_size         | Numeric absolute target exposure.                                                                |
| compute_features    | Logical; when 'TRUE', missing features are added to 'DT' in place.                               |
| use_log             | Logical; if 'TRUE', relative strength and momentum are based on log-return aggregation.          |
| strat_id            | Integer strategy identifier recorded on generated actions.                                       |
| tol_pos             | Numeric tolerance passed to the action planner.                                                  |
| debug               | Logical; when 'TRUE', returns a list with the generated target vector and feature column names.  |

**Value**

A list produced by 'gen\_action\_plan\_rcpp()'.

---

strat\_relative\_strength\_dual\_momentum\_tgt\_pos

*Relative-Strength-Dual-Momentum Target Positions*

---

**Description**

Generates a benchmark-relative target-position path that requires both positive relative strength and positive absolute momentum for long exposure, and optionally allows short exposure when both are negative.

**Usage**

```
strat_relative_strength_dual_momentum_tgt_pos(
  DT,
  x_col = "close",
  y_col = "benchmark_close",
  rs_n = 20L,
  mom_n = 60L,
  rs_long_threshold = 1,
  rs_short_threshold = 1,
  mom_long_threshold = 0,
  mom_short_threshold = 0,
  allow_short = TRUE,
```

```

    target_size = 1,
    compute_features = TRUE,
    use_log = TRUE,
    debug = FALSE
  )

```

### Arguments

|                     |                                                                                                   |
|---------------------|---------------------------------------------------------------------------------------------------|
| DT                  | A 'data.table' containing the traded and benchmark price columns.                                 |
| x_col               | Traded asset price column.                                                                        |
| y_col               | Benchmark price column.                                                                           |
| rs_n                | Integer relative-strength lookback window.                                                        |
| mom_n               | Integer absolute-momentum lookback window.                                                        |
| rs_long_threshold   | Numeric threshold above which the traded asset is considered relatively strong enough to go long. |
| rs_short_threshold  | Numeric threshold below which the traded asset is considered relatively weak enough to go short.  |
| mom_long_threshold  | Numeric momentum threshold required for long exposure.                                            |
| mom_short_threshold | Numeric momentum threshold required for short exposure.                                           |
| allow_short         | Logical; if 'TRUE', allows short targets.                                                         |
| target_size         | Numeric absolute target exposure.                                                                 |
| compute_features    | Logical; when 'TRUE', missing features are added to 'DT' in place.                                |
| use_log             | Logical; if 'TRUE', relative strength and momentum are based on log-return aggregation.           |
| debug               | Logical; when 'TRUE', returns a list with the generated target vector and feature column names.   |

### Value

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

strat\_relative\_strength\_persistence\_action\_plan

*Relative-Strength-Persistence Action Plan*

---

### Description

Applies the persistent relative-strength rule to the latest bar and translates the resulting target exposure into an executable action plan.

**Usage**

```
strat_relative_strength_persistence_action_plan(
  DT,
  state,
  x_col = "close",
  y_col = "benchmark_close",
  n = 20L,
  long_threshold = 1.02,
  short_threshold = 0.98,
  persist_n = 3L,
  target_size = 1,
  compute_features = TRUE,
  use_log = TRUE,
  strat_id = 508L,
  tol_pos = 0.1,
  debug = FALSE
)
```

**Arguments**

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A ‘data.table’ containing the traded and benchmark price columns.                               |
| state            | Named list describing the current trading state.                                                |
| x_col            | Traded asset price column.                                                                      |
| y_col            | Benchmark or paired asset price column.                                                         |
| n                | Integer relative-strength lookback window.                                                      |
| long_threshold   | Numeric threshold above which relative strength is treated as long-positive.                    |
| short_threshold  | Numeric threshold below which relative strength is treated as short-negative.                   |
| persist_n        | Integer number of consecutive bars required for activation.                                     |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when ‘TRUE’, missing relative-strength features are added to ‘DT’ in place.            |
| use_log          | Logical; when ‘TRUE’, relative strength uses log returns.                                       |
| strat_id         | Integer strategy identifier recorded on generated actions.                                      |
| tol_pos          | Numeric tolerance passed to the action planner.                                                 |
| debug            | Logical; when ‘TRUE’, returns a list with the generated target vector and feature column names. |

**Value**

A list produced by ‘gen\_action\_plan\_rcpp()’.

---

strat\_relative\_strength\_persistence\_tgt\_pos

*Relative-Strength-Persistence Target Positions*


---

## Description

Generates a target-position path from relative-strength signals that must persist for multiple bars before turning on. Persistent relative outperformance targets a long exposure and persistent under-performance targets a short exposure.

## Usage

```
strat_relative_strength_persistence_tgt_pos(
  DT,
  x_col = "close",
  y_col = "benchmark_close",
  n = 20L,
  long_threshold = 1.02,
  short_threshold = 0.98,
  persist_n = 3L,
  target_size = 1,
  compute_features = TRUE,
  use_log = TRUE,
  debug = FALSE
)
```

## Arguments

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A 'data.table' containing the traded and benchmark price columns.                               |
| x_col            | Traded asset price column.                                                                      |
| y_col            | Benchmark or paired asset price column.                                                         |
| n                | Integer relative-strength lookback window.                                                      |
| long_threshold   | Numeric threshold above which relative strength is treated as long-positive.                    |
| short_threshold  | Numeric threshold below which relative strength is treated as short-negative.                   |
| persist_n        | Integer number of consecutive bars required for activation.                                     |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing relative-strength features are added to 'DT' in place.            |
| use_log          | Logical; when 'TRUE', relative strength uses log returns.                                       |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

Value

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

|                                           |
|-------------------------------------------|
| strat_relative_strength_tgt_pos           |
| <i>Relative-Strength Target Positions</i> |

---

Description

Generates a simple benchmark-relative target-position path from rolling relative-strength values. Strong relative performance targets a long exposure, while weak relative performance targets a short exposure.

Usage

```
strat_relative_strength_tgt_pos(  
  DT,  
  x_col = "close",  
  y_col = "benchmark_close",  
  n = 20L,  
  long_threshold = 1.02,  
  short_threshold = 0.98,  
  target_size = 1,  
  compute_features = TRUE,  
  use_log = TRUE,  
  debug = FALSE  
)
```

Arguments

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A 'data.table' containing the traded and benchmark price columns.                               |
| x_col            | Traded asset price column.                                                                      |
| y_col            | Benchmark price column.                                                                         |
| n                | Integer relative-strength lookback window.                                                      |
| long_threshold   | Numeric threshold above which the traded asset is considered relatively strong.                 |
| short_threshold  | Numeric threshold below which the traded asset is considered relatively weak.                   |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing relative-strength features are added to 'DT' in place.            |
| use_log          | Logical; if 'TRUE', relative strength is based on log-return aggregation.                       |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A numeric vector of target positions, or a list when ‘debug = TRUE’.

---

|                               |
|-------------------------------|
| strat_roll_yield_action_plan  |
| <i>Roll-Yield Action Plan</i> |

---

**Description**

Applies the roll-yield rule to the latest summarized row and translates the resulting target exposure into an executable action plan.

**Usage**

```
strat_roll_yield_action_plan(  
  DT,  
  state,  
  date_col = "date",  
  rank_col = "contract_rank",  
  price_col = "close",  
  time_to_expiry_col = "time_to_expiry",  
  rank_front = 1L,  
  rank_deferred = 2L,  
  long_threshold = 0,  
  short_threshold = 0,  
  target_size = 1,  
  compute_features = TRUE,  
  strat_id = 604L,  
  tol_pos = 0.1,  
  debug = FALSE  
)
```

**Arguments**

|                    |                                                                                                                                                                                         |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DT                 | A futures curve ‘data.table’ panel when ‘compute_features = TRUE’, or a summarized per-date ‘data.table’ containing the relevant ‘roll_yield_*’ column when ‘compute_features = FALSE’. |
| state              | Named list describing the current trading state.                                                                                                                                        |
| date_col           | Date or timestamp column name for the curve panel.                                                                                                                                      |
| rank_col           | Contract-rank column name.                                                                                                                                                              |
| price_col          | Price column name.                                                                                                                                                                      |
| time_to_expiry_col | Time-to-expiry column name expressed in years.                                                                                                                                          |
| rank_front         | Integer front-contract rank.                                                                                                                                                            |
| rank_deferred      | Integer deferred-contract rank.                                                                                                                                                         |

|                  |                                                                                                                |
|------------------|----------------------------------------------------------------------------------------------------------------|
| long_threshold   | Numeric threshold above which roll yield is treated as positive enough to go long.                             |
| short_threshold  | Numeric threshold below which roll yield is treated as negative enough to go short.                            |
| target_size      | Numeric absolute target exposure.                                                                              |
| compute_features | Logical; when 'TRUE', the input panel is summarized to per-date roll-yield features before generating targets. |
| strat_id         | Integer strategy identifier recorded on generated actions.                                                     |
| tol_pos          | Numeric tolerance passed to the action planner.                                                                |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and summary data.                        |

**Value**

A list produced by 'gen\_action\_plan\_rcpp()'.

---

strat\_roll\_yield\_cross\_sectional\_action\_plan  
*Roll-Yield-Cross-Sectional Action Plan*

---

**Description**

Applies the latest cross-sectional roll-yield target weights to a portfolio state and converts them into a portfolio-adjustment table plus order intents.

**Usage**

```
strat_roll_yield_cross_sectional_action_plan(
  DT,
  portfolio_state,
  equity,
  date_col = "date",
  asset_col = "asset",
  signal_col = "roll_yield",
  long_n = 1L,
  short_n = 0L,
  gross_exposure = 1,
  strat_id = 509L,
  pricing_method = "market",
  debug = FALSE
)
```

**Arguments**

|                 |                                                                                                 |
|-----------------|-------------------------------------------------------------------------------------------------|
| DT              | A panel 'data.table' already summarized to one row per asset-date.                              |
| portfolio_state | A 'data.table' containing current per-asset portfolio state.                                    |
| equity          | Numeric total portfolio equity used to scale target weights into target notionals.              |
| date_col        | Date column used for cross-sectional grouping.                                                  |
| asset_col       | Asset identifier column.                                                                        |
| signal_col      | Roll-yield signal column used for ranking.                                                      |
| long_n          | Integer number of assets to allocate long each date.                                            |
| short_n         | Integer number of assets to allocate short each date.                                           |
| gross_exposure  | Numeric gross exposure allocated across each side.                                              |
| strat_id        | Integer strategy identifier recorded on generated order intents.                                |
| pricing_method  | Pricing method passed to 'build_order_intents()'.                                               |
| debug           | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A list with 'adjustment\_plan', 'order\_intents', and the latest target weights.

---

strat\_roll\_yield\_cross\_sectional\_tgt\_pos

*Roll-Yield-Cross-Sectional Target Positions*

---

**Description**

Generates per-row target weights by ranking a roll-yield signal across assets within each date. Top-ranked assets receive positive weights and optionally bottom-ranked assets receive negative weights.

**Usage**

```
strat_roll_yield_cross_sectional_tgt_pos(
  DT,
  date_col = "date",
  asset_col = "asset",
  signal_col = "roll_yield",
  long_n = 1L,
  short_n = 0L,
  gross_exposure = 1,
  debug = FALSE
)
```

**Arguments**

|                |                                                                                                 |
|----------------|-------------------------------------------------------------------------------------------------|
| DT             | A panel 'data.table' already summarized to one row per asset-date.                              |
| date_col       | Date column used for cross-sectional grouping.                                                  |
| asset_col      | Asset identifier column.                                                                        |
| signal_col     | Roll-yield signal column used for ranking.                                                      |
| long_n         | Integer number of assets to allocate long each date.                                            |
| short_n        | Integer number of assets to allocate short each date.                                           |
| gross_exposure | Numeric gross exposure allocated across each side.                                              |
| debug          | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

strat\_roll\_yield\_mean\_revert\_action\_plan  
*Roll-Yield-Mean-Revert Action Plan*

---

**Description**

Applies the roll-yield mean-reversion rule to the latest summarized row and translates the resulting target exposure into an executable action plan.

**Usage**

```
strat_roll_yield_mean_revert_action_plan(
  DT,
  state,
  date_col = "date",
  rank_col = "contract_rank",
  price_col = "close",
  time_to_expiry_col = "time_to_expiry",
  rank_front = 1L,
  rank_deferred = 2L,
  z_n = 60L,
  long_z = -1,
  short_z = 1,
  target_size = 1,
  compute_features = TRUE,
  strat_id = 614L,
  tol_pos = 0.1,
  debug = FALSE
)
```

**Arguments**

|                    |                                                                                                                                                                                                |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DT                 | A futures curve 'data.table' panel when 'compute_features = TRUE', or a summarized per-date 'data.table' containing the relevant 'zscore_roll_yield_*' column when 'compute_features = FALSE'. |
| state              | Named list describing the current trading state.                                                                                                                                               |
| date_col           | Date or timestamp column name for the curve panel.                                                                                                                                             |
| rank_col           | Contract-rank column name.                                                                                                                                                                     |
| price_col          | Price column name.                                                                                                                                                                             |
| time_to_expiry_col | Time-to-expiry column name expressed in years.                                                                                                                                                 |
| rank_front         | Integer front-contract rank.                                                                                                                                                                   |
| rank_deferred      | Integer deferred-contract rank.                                                                                                                                                                |
| z_n                | Integer rolling window used to standardize roll yield.                                                                                                                                         |
| long_z             | Numeric lower z-score threshold used for long entries.                                                                                                                                         |
| short_z            | Numeric upper z-score threshold used for short entries.                                                                                                                                        |
| target_size        | Numeric absolute target exposure.                                                                                                                                                              |
| compute_features   | Logical; when 'TRUE', the input panel is summarized to per-date roll-yield features before generating targets.                                                                                 |
| strat_id           | Integer strategy identifier recorded on generated actions.                                                                                                                                     |
| tol_pos            | Numeric tolerance passed to the action planner.                                                                                                                                                |
| debug              | Logical; when 'TRUE', returns a list with the generated target vector and summary data.                                                                                                        |

**Value**

A list produced by 'gen\_action\_plan\_rcpp()'.

---

strat\_roll\_yield\_mean\_revert\_tgt\_pos

*Roll-Yield-Mean-Revert Target Positions*

---

**Description**

Generates a simple futures target-position path by fading extreme roll-yield z-scores. Unusually low roll yield targets a long exposure and unusually high roll yield targets a short exposure.

**Usage**

```
strat_roll_yield_mean_revert_tgt_pos(
  DT,
  date_col = "date",
  rank_col = "contract_rank",
  price_col = "close",
  time_to_expiry_col = "time_to_expiry",
  rank_front = 1L,
  rank_deferred = 2L,
  z_n = 60L,
  long_z = -1,
  short_z = 1,
  target_size = 1,
  compute_features = TRUE,
  debug = FALSE
)
```

**Arguments**

|                    |                                                                                                                                                                                                |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DT                 | A futures curve 'data.table' panel when 'compute_features = TRUE', or a summarized per-date 'data.table' containing the relevant 'zscore_roll_yield_*' column when 'compute_features = FALSE'. |
| date_col           | Date or timestamp column name for the curve panel.                                                                                                                                             |
| rank_col           | Contract-rank column name.                                                                                                                                                                     |
| price_col          | Price column name.                                                                                                                                                                             |
| time_to_expiry_col | Time-to-expiry column name expressed in years.                                                                                                                                                 |
| rank_front         | Integer front-contract rank.                                                                                                                                                                   |
| rank_deferred      | Integer deferred-contract rank.                                                                                                                                                                |
| z_n                | Integer rolling window used to standardize roll yield.                                                                                                                                         |
| long_z             | Numeric lower z-score threshold used for long entries.                                                                                                                                         |
| short_z            | Numeric upper z-score threshold used for short entries.                                                                                                                                        |
| target_size        | Numeric absolute target exposure.                                                                                                                                                              |
| compute_features   | Logical; when 'TRUE', the input panel is summarized to per-date roll-yield features before generating targets.                                                                                 |
| debug              | Logical; when 'TRUE', returns a list with the generated target vector and summary data.                                                                                                        |

**Value**

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

strat\_roll\_yield\_momentum\_action\_plan

*Roll-Yield-Momentum Action Plan*


---

## Description

Applies the roll-yield-plus-momentum rule to the latest summarized row and translates the resulting target exposure into an executable action plan.

## Usage

```
strat_roll_yield_momentum_action_plan(
    DT,
    state,
    date_col = "date",
    rank_col = "contract_rank",
    price_col = "close",
    time_to_expiry_col = "time_to_expiry",
    rank_front = 1L,
    rank_deferred = 2L,
    mom_n = 20L,
    long_roll_threshold = 0,
    short_roll_threshold = 0,
    long_mom_threshold = 0,
    short_mom_threshold = 0,
    target_size = 1,
    compute_features = TRUE,
    strat_id = 607L,
    tol_pos = 0.1,
    debug = FALSE
)
```

## Arguments

|                    |                                                                                                                                                                                                         |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DT                 | A futures curve 'data.table' panel when 'compute_features = TRUE', or a summarized per-date 'data.table' containing the relevant roll-yield and front-momentum columns when 'compute_features = FALSE'. |
| state              | Named list describing the current trading state.                                                                                                                                                        |
| date_col           | Date or timestamp column name for the curve panel.                                                                                                                                                      |
| rank_col           | Contract-rank column name.                                                                                                                                                                              |
| price_col          | Price column name.                                                                                                                                                                                      |
| time_to_expiry_col | Time-to-expiry column name expressed in years.                                                                                                                                                          |
| rank_front         | Integer front-contract rank.                                                                                                                                                                            |
| rank_deferred      | Integer deferred-contract rank.                                                                                                                                                                         |

|                      |                                                                                         |
|----------------------|-----------------------------------------------------------------------------------------|
| mom_n                | Integer momentum lookback window applied to the summarized front-contract price.        |
| long_roll_threshold  | Numeric roll-yield threshold required for long exposure.                                |
| short_roll_threshold | Numeric roll-yield threshold required for short exposure.                               |
| long_mom_threshold   | Numeric momentum threshold required for long exposure.                                  |
| short_mom_threshold  | Numeric momentum threshold required for short exposure.                                 |
| target_size          | Numeric absolute target exposure.                                                       |
| compute_features     | Logical; when 'TRUE', the input panel is summarized before generating targets.          |
| strat_id             | Integer strategy identifier recorded on generated actions.                              |
| tol_pos              | Numeric tolerance passed to the action planner.                                         |
| debug                | Logical; when 'TRUE', returns a list with the generated target vector and summary data. |

**Value**

A list produced by 'gen\_action\_plan\_rcpp()'.

---

strat\_roll\_yield\_momentum\_tgt\_pos

*Roll-Yield-Momentum Target Positions*

---

**Description**

Generates a futures target-position path from annualized roll yield with a front-contract momentum confirmation filter.

**Usage**

```
strat_roll_yield_momentum_tgt_pos(
  DT,
  date_col = "date",
  rank_col = "contract_rank",
  price_col = "close",
  time_to_expiry_col = "time_to_expiry",
  rank_front = 1L,
  rank_deferred = 2L,
  mom_n = 20L,
  long_roll_threshold = 0,
  short_roll_threshold = 0,
  long_mom_threshold = 0,
```

```

    short_mom_threshold = 0,
    target_size = 1,
    compute_features = TRUE,
    debug = FALSE
)
```

### Arguments

|                      |                                                                                                                                                                                                         |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DT                   | A futures curve 'data.table' panel when 'compute_features = TRUE', or a summarized per-date 'data.table' containing the relevant roll-yield and front-momentum columns when 'compute_features = FALSE'. |
| date_col             | Date or timestamp column name for the curve panel.                                                                                                                                                      |
| rank_col             | Contract-rank column name.                                                                                                                                                                              |
| price_col            | Price column name.                                                                                                                                                                                      |
| time_to_expiry_col   | Time-to-expiry column name expressed in years.                                                                                                                                                          |
| rank_front           | Integer front-contract rank.                                                                                                                                                                            |
| rank_deferred        | Integer deferred-contract rank.                                                                                                                                                                         |
| mom_n                | Integer momentum lookback window applied to the summarized front-contract price.                                                                                                                        |
| long_roll_threshold  | Numeric roll-yield threshold required for long exposure.                                                                                                                                                |
| short_roll_threshold | Numeric roll-yield threshold required for short exposure.                                                                                                                                               |
| long_mom_threshold   | Numeric momentum threshold required for long exposure.                                                                                                                                                  |
| short_mom_threshold  | Numeric momentum threshold required for short exposure.                                                                                                                                                 |
| target_size          | Numeric absolute target exposure.                                                                                                                                                                       |
| compute_features     | Logical; when 'TRUE', the input panel is summarized before generating targets.                                                                                                                          |
| debug                | Logical; when 'TRUE', returns a list with the generated target vector and summary data.                                                                                                                 |

### Value

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

strat\_roll\_yield\_tgt\_pos

*Roll-Yield Target Positions*


---

## Description

Generates a simple futures target-position path from annualized roll yield. Positive roll yield targets a long exposure and negative roll yield targets a short exposure.

## Usage

```
strat_roll_yield_tgt_pos(
  DT,
  date_col = "date",
  rank_col = "contract_rank",
  price_col = "close",
  time_to_expiry_col = "time_to_expiry",
  rank_front = 1L,
  rank_deferred = 2L,
  long_threshold = 0,
  short_threshold = 0,
  target_size = 1,
  compute_features = TRUE,
  debug = FALSE
)
```

## Arguments

|                    |                                                                                                                                                                                         |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DT                 | A futures curve ‘data.table’ panel when ‘compute_features = TRUE’, or a summarized per-date ‘data.table’ containing the relevant ‘roll_yield_*’ column when ‘compute_features = FALSE’. |
| date_col           | Date or timestamp column name for the curve panel.                                                                                                                                      |
| rank_col           | Contract-rank column name.                                                                                                                                                              |
| price_col          | Price column name.                                                                                                                                                                      |
| time_to_expiry_col | Time-to-expiry column name expressed in years.                                                                                                                                          |
| rank_front         | Integer front-contract rank.                                                                                                                                                            |
| rank_deferred      | Integer deferred-contract rank.                                                                                                                                                         |
| long_threshold     | Numeric threshold above which roll yield is treated as positive enough to go long.                                                                                                      |
| short_threshold    | Numeric threshold below which roll yield is treated as negative enough to go short.                                                                                                     |
| target_size        | Numeric absolute target exposure.                                                                                                                                                       |

|                  |                                                                                                                |
|------------------|----------------------------------------------------------------------------------------------------------------|
| compute_features | Logical; when 'TRUE', the input panel is summarized to per-date roll-yield features before generating targets. |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and summary data.                        |

**Value**

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

|                                   |
|-----------------------------------|
| strat_rsi_divergence_action_plan  |
| <i>RSI-Divergence Action Plan</i> |

---

**Description**

Applies the RSI-divergence rule to the latest bar and translates the resulting target exposure into an executable action plan.

**Usage**

```
strat_rsi_divergence_action_plan(  
  DT,  
  state,  
  rsi_n = 14L,  
  pivot_left = 2L,  
  pivot_right = 2L,  
  exit_level = 50,  
  target_size = 1,  
  compute_features = TRUE,  
  strat_id = 316L,  
  tol_pos = 0.1,  
  debug = FALSE  
)
```

**Arguments**

|             |                                                             |
|-------------|-------------------------------------------------------------|
| DT          | A candle 'data.table'.                                      |
| state       | Named list describing the current trading state.            |
| rsi_n       | Integer RSI window.                                         |
| pivot_left  | Integer number of bars on the left used to confirm pivots.  |
| pivot_right | Integer number of bars on the right used to confirm pivots. |
| exit_level  | Numeric neutral RSI level used to close open targets.       |
| target_size | Numeric absolute target exposure.                           |

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| compute_features | Logical; when 'TRUE', missing RSI features are added to 'DT' in place.                          |
| strat_id         | Integer strategy identifier recorded on generated actions.                                      |
| tol_pos          | Numeric tolerance passed to the action planner.                                                 |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

### Value

A list produced by 'gen\_action\_plan\_rcpp()'.

---

strat\_rsi\_divergence\_tgt\_pos  
*RSI-Divergence Target Positions*

---

### Description

Generates a mean-reversion target-position path from simple confirmed RSI divergences. Bullish divergence requires a lower confirmed price pivot low paired with a higher confirmed RSI pivot low. Bearish divergence requires a higher confirmed price pivot high paired with a lower confirmed RSI pivot high.

### Usage

```
strat_rsi_divergence_tgt_pos(
  DT,
  rsi_n = 14L,
  pivot_left = 2L,
  pivot_right = 2L,
  exit_level = 50,
  target_size = 1,
  compute_features = TRUE,
  debug = FALSE
)
```

### Arguments

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle 'data.table'.                                                                          |
| rsi_n            | Integer RSI window.                                                                             |
| pivot_left       | Integer number of bars on the left used to confirm pivots.                                      |
| pivot_right      | Integer number of bars on the right used to confirm pivots.                                     |
| exit_level       | Numeric neutral RSI level used to close open targets.                                           |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing RSI features are added to 'DT' in place.                          |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

strat\_rsi\_dynamic\_threshold\_revert\_action\_plan

*RSI-Dynamic-Threshold-Reversion Action Plan*


---

**Description**

Applies the dynamic-threshold RSI mean-reversion rule to the latest bar and translates the resulting target exposure into an executable action plan.

**Usage**

```
strat_rsi_dynamic_threshold_revert_action_plan(
  DT,
  state,
  rsi_n = 14L,
  threshold_n = 252L,
  lower_q = 0.1,
  upper_q = 0.9,
  exit_level = 0.5,
  target_size = 1,
  compute_features = TRUE,
  strat_id = 315L,
  tol_pos = 0.1,
  debug = FALSE
)
```

**Arguments**

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle 'data.table'.                                                                          |
| state            | Named list describing the current trading state.                                                |
| rsi_n            | Integer RSI window.                                                                             |
| threshold_n      | Integer rolling window used to estimate dynamic thresholds.                                     |
| lower_q          | Numeric lower quantile used for long entries.                                                   |
| upper_q          | Numeric upper quantile used for short entries.                                                  |
| exit_level       | Numeric rolling quantile used as the neutral exit level.                                        |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing RSI and rolling quantile features are added to 'DT' in place.     |
| strat_id         | Integer strategy identifier recorded on generated actions.                                      |
| tol_pos          | Numeric tolerance passed to the action planner.                                                 |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

Value

A list produced by 'gen\_action\_plan\_rcpp()'.

---

|                                                         |
|---------------------------------------------------------|
| strat_rsi_dynamic_threshold_revert_tgt_pos              |
| <i>RSI-Dynamic-Threshold-Reversion Target Positions</i> |

---

Description

Generates a mean-reversion target-position path from RSI with rolling quantile thresholds. Long exposure is opened when RSI falls into the rolling lower tail, short exposure is opened when RSI rises into the rolling upper tail, and positions are closed once RSI reverts toward a rolling neutral quantile.

Usage

```
strat_rsi_dynamic_threshold_revert_tgt_pos(  
  DT,  
  rsi_n = 14L,  
  threshold_n = 252L,  
  lower_q = 0.1,  
  upper_q = 0.9,  
  exit_level = 0.5,  
  target_size = 1,  
  compute_features = TRUE,  
  debug = FALSE  
)
```

Arguments

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle 'data.table'.                                                                          |
| rsi_n            | Integer RSI window.                                                                             |
| threshold_n      | Integer rolling window used to estimate dynamic thresholds.                                     |
| lower_q          | Numeric lower quantile used for long entries.                                                   |
| upper_q          | Numeric upper quantile used for short entries.                                                  |
| exit_level       | Numeric rolling quantile used as the neutral exit level.                                        |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing RSI and rolling quantile features are added to 'DT' in place.     |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

Value

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

strat\_rsi\_logr\_revert\_action\_plan

*Log-Return-RSI-Reversion Action Plan*


---

## Description

Applies the log-return RSI mean-reversion rule to the latest bar and translates the resulting target exposure into an executable action plan.

## Usage

```
strat_rsi_logr_revert_action_plan(
  DT,
  state,
  h = 12,
  oversold = 40,
  overbought = 60,
  exit_level = 50,
  target_size = 1,
  compute_features = TRUE,
  strat_id = 306L,
  tol_pos = 0.1,
  debug = FALSE
)
```

## Arguments

|                  |                                                                                                                                     |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| DT               | A candle ‘data.table’.                                                                                                              |
| state            | Named list describing the current trading state.                                                                                    |
| h                | Numeric half-life for the log-return RSI smoother.                                                                                  |
| oversold         | Numeric oversold threshold. The log-return RSI default is intentionally closer to the center than classic ‘30/70’ RSI thresholds.   |
| overbought       | Numeric overbought threshold. The log-return RSI default is intentionally closer to the center than classic ‘30/70’ RSI thresholds. |
| exit_level       | Numeric neutral RSI level used to close open targets.                                                                               |
| target_size      | Numeric absolute target exposure.                                                                                                   |
| compute_features | Logical; when ‘TRUE’, missing log-return RSI features are added to ‘DT’ in place.                                                   |
| strat_id         | Integer strategy identifier recorded on generated actions.                                                                          |
| tol_pos          | Numeric tolerance passed to the action planner.                                                                                     |
| debug            | Logical; when ‘TRUE’, returns a list with the generated target vector and feature column names.                                     |

**Value**

A list produced by ‘gen\_action\_plan\_rcpp()’.

---

|                                                  |
|--------------------------------------------------|
| strat_rsi_logr_revert_tgt_pos                    |
| <i>Log-Return-RSI-Reversion Target Positions</i> |

---

**Description**

Generates a simple mean-reversion target-position path from strategy’s log-return RSI feature. Oversold log-return RSI opens a long target, overbought log-return RSI opens a short target, and open targets are closed once the indicator mean-reverts to the exit level.

**Usage**

```
strat_rsi_logr_revert_tgt_pos(  
  DT,  
  h = 12,  
  oversold = 40,  
  overbought = 60,  
  exit_level = 50,  
  target_size = 1,  
  compute_features = TRUE,  
  debug = FALSE  
)
```

**Arguments**

|                  |                                                                                                                                     |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| DT               | A candle ‘data.table’.                                                                                                              |
| h                | Numeric half-life for the log-return RSI smoother.                                                                                  |
| oversold         | Numeric oversold threshold. The log-return RSI default is intentionally closer to the center than classic ‘30/70’ RSI thresholds.   |
| overbought       | Numeric overbought threshold. The log-return RSI default is intentionally closer to the center than classic ‘30/70’ RSI thresholds. |
| exit_level       | Numeric neutral RSI level used to close open targets.                                                                               |
| target_size      | Numeric absolute target exposure.                                                                                                   |
| compute_features | Logical; when ‘TRUE’, missing log-return RSI features are added to ‘DT’ in place.                                                   |
| debug            | Logical; when ‘TRUE’, returns a list with the generated target vector and feature column names.                                     |

**Value**

A numeric vector of target positions, or a list when ‘debug = TRUE’.

---

```
strat_rsi_revert_action_plan
```

*RSI-Reversion Action Plan*

---

## Description

Applies the RSI mean-reversion rule to the latest bar and translates the resulting target exposure into an executable action plan.

## Usage

```
strat_rsi_revert_action_plan(
  DT,
  state,
  n = 14L,
  oversold = 30,
  overbought = 70,
  exit_level = 50,
  target_size = 1,
  compute_features = TRUE,
  strat_id = 303L,
  tol_pos = 0.1,
  debug = FALSE
)
```

## Arguments

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle 'data.table'.                                                                          |
| state            | Named list describing the current trading state.                                                |
| n                | Integer RSI window.                                                                             |
| oversold         | Numeric oversold threshold.                                                                     |
| overbought       | Numeric overbought threshold.                                                                   |
| exit_level       | Numeric neutral RSI level used to close open targets.                                           |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing RSI features are added to 'DT' in place.                          |
| strat_id         | Integer strategy identifier recorded on generated actions.                                      |
| tol_pos          | Numeric tolerance passed to the action planner.                                                 |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

## Value

A list produced by 'gen\_action\_plan\_rcpp()'.

---

strat\_rsi\_revert\_tgt\_pos

*RSI-Reversion Target Positions*


---

## Description

Generates a simple mean-reversion target-position path from classic RSI levels. Oversold RSI opens a long target, overbought RSI opens a short target, and open targets are closed once RSI mean-reverts to the exit level.

## Usage

```
strat_rsi_revert_tgt_pos(
  DT,
  n = 14L,
  oversold = 30,
  overbought = 70,
  exit_level = 50,
  target_size = 1,
  compute_features = TRUE,
  debug = FALSE
)
```

## Arguments

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle 'data.table'.                                                                          |
| n                | Integer RSI window.                                                                             |
| oversold         | Numeric oversold threshold.                                                                     |
| overbought       | Numeric overbought threshold.                                                                   |
| exit_level       | Numeric neutral RSI level used to close open targets.                                           |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing RSI features are added to 'DT' in place.                          |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

## Value

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

|                                              |
|----------------------------------------------|
| strat_rsi_trend_aware_revert_action_plan     |
| <i>RSI-Trend-Aware-Reversion Action Plan</i> |

---

**Description**

Applies the trend-aware RSI-reversion rule to the latest bar and translates the resulting target exposure into an executable action plan.

**Usage**

```
strat_rsi_trend_aware_revert_action_plan(  
    DT,  
    state,  
    rsi_n = 14L,  
    trend_n = 50L,  
    oversold = 30,  
    overbought = 70,  
    exit_level = 50,  
    target_size = 1,  
    compute_features = TRUE,  
    strat_id = 311L,  
    tol_pos = 0.1,  
    debug = FALSE  
)
```

**Arguments**

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle ‘data.table’.                                                                          |
| state            | Named list describing the current trading state.                                                |
| rsi_n            | Integer RSI window.                                                                             |
| trend_n          | Integer EMA window used as the trend filter.                                                    |
| oversold         | Numeric RSI threshold used for long entries.                                                    |
| overbought       | Numeric RSI threshold used for short entries.                                                   |
| exit_level       | Numeric RSI level used to exit open positions.                                                  |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when ‘TRUE’, missing RSI and EMA features are added to ‘DT’ in place.                  |
| strat_id         | Integer strategy identifier recorded on generated actions.                                      |
| tol_pos          | Numeric tolerance passed to the action planner.                                                 |
| debug            | Logical; when ‘TRUE’, returns a list with the generated target vector and feature column names. |

**Value**

A list produced by ‘gen\_action\_plan\_rcpp()’.

---

|                                                   |
|---------------------------------------------------|
| strat_rsi_trend_aware_revert_tgt_pos              |
| <i>RSI-Trend-Aware-Reversion Target Positions</i> |

---

**Description**

Generates a target-position path from RSI mean-reversion signals that are conditioned on trend direction. Oversold longs are only allowed in uptrends, while overbought shorts are only allowed in downtrends.

**Usage**

```
strat_rsi_trend_aware_revert_tgt_pos(  
  DT,  
  rsi_n = 14L,  
  trend_n = 50L,  
  oversold = 30,  
  overbought = 70,  
  exit_level = 50,  
  target_size = 1,  
  compute_features = TRUE,  
  debug = FALSE  
)
```

**Arguments**

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle ‘data.table’.                                                                          |
| rsi_n            | Integer RSI window.                                                                             |
| trend_n          | Integer EMA window used as the trend filter.                                                    |
| oversold         | Numeric RSI threshold used for long entries.                                                    |
| overbought       | Numeric RSI threshold used for short entries.                                                   |
| exit_level       | Numeric RSI level used to exit open positions.                                                  |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when ‘TRUE’, missing RSI and EMA features are added to ‘DT’ in place.                  |
| debug            | Logical; when ‘TRUE’, returns a list with the generated target vector and feature column names. |

**Value**

A numeric vector of target positions, or a list when ‘debug = TRUE’.

---

strat\_straddle\_action\_plan

*Straddle Action Plan*


---

## Description

Applies the straddle-proxy rule to the latest summarized row and translates the resulting target exposure into an executable action plan.

## Usage

```
strat_straddle_action_plan(
  DT,
  state,
  date_col = "date",
  expiry_col = "time_to_expiry",
  moneyness_col = "option_log_forward_moneyness",
  iv_col = "iv",
  long_iv_threshold = 0.2,
  short_iv_threshold = 0.4,
  target_size = 1,
  compute_features = TRUE,
  strat_id = 703L,
  tol_pos = 0.1,
  debug = FALSE
)
```

## Arguments

|                    |                                                                                                                                                      |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| DT                 | An option-chain 'data.table' when 'compute_features = TRUE', or a summarized 'data.table' containing 'iv_atm_front' when 'compute_features = FALSE'. |
| state              | Named list describing the current trading state.                                                                                                     |
| date_col           | Date or timestamp column name.                                                                                                                       |
| expiry_col         | Time-to-expiry column name.                                                                                                                          |
| moneyness_col      | Forward-moneyness feature column name.                                                                                                               |
| iv_col             | Implied-volatility column name.                                                                                                                      |
| long_iv_threshold  | Numeric threshold at or below which ATM IV is treated as cheap enough to go long.                                                                    |
| short_iv_threshold | Numeric threshold at or above which ATM IV is treated as rich enough to go short.                                                                    |
| target_size        | Numeric absolute target exposure.                                                                                                                    |
| compute_features   | Logical; when 'TRUE', the chain is summarized before generating targets.                                                                             |

|          |                                                                                         |
|----------|-----------------------------------------------------------------------------------------|
| strat_id | Integer strategy identifier recorded on generated actions.                              |
| tol_pos  | Numeric tolerance passed to the action planner.                                         |
| debug    | Logical; when 'TRUE', returns a list with the generated target vector and summary data. |

**Value**

A list produced by 'gen\_action\_plan\_rcpp()'.

---

strat\_straddle\_tgt\_pos

*Straddle Target Positions*

---

**Description**

Generates a simple straddle-proxy target-position path from front-expiry ATM implied volatility. Low front ATM IV targets a long straddle proxy exposure, and high front ATM IV targets a short straddle proxy exposure.

**Usage**

```
strat_straddle_tgt_pos(
  DT,
  date_col = "date",
  expiry_col = "time_to_expiry",
  moneyness_col = "option_log_forward_moneyness",
  iv_col = "iv",
  long_iv_threshold = 0.2,
  short_iv_threshold = 0.4,
  target_size = 1,
  compute_features = TRUE,
  debug = FALSE
)
```

**Arguments**

|                   |                                                                                                                                                      |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| DT                | An option-chain 'data.table' when 'compute_features = TRUE', or a summarized 'data.table' containing 'iv_atm_front' when 'compute_features = FALSE'. |
| date_col          | Date or timestamp column name.                                                                                                                       |
| expiry_col        | Time-to-expiry column name.                                                                                                                          |
| moneyness_col     | Forward-moneyness feature column name.                                                                                                               |
| iv_col            | Implied-volatility column name.                                                                                                                      |
| long_iv_threshold | Numeric threshold at or below which ATM IV is treated as cheap enough to go long.                                                                    |

|                    |                                                                                         |
|--------------------|-----------------------------------------------------------------------------------------|
| short_iv_threshold | Numeric threshold at or above which ATM IV is treated as rich enough to go short.       |
| target_size        | Numeric absolute target exposure.                                                       |
| compute_features   | Logical; when 'TRUE', the chain is summarized before generating targets.                |
| debug              | Logical; when 'TRUE', returns a list with the generated target vector and summary data. |

**Value**

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

strat\_strangle\_action\_plan  
*Strangle Action Plan*

---

**Description**

Applies the strangle-proxy rule to the latest summarized row and translates the resulting target exposure into an executable action plan.

**Usage**

```
strat_strangle_action_plan(
  DT,
  state,
  date_col = "date",
  expiry_col = "time_to_expiry",
  type_col = "type",
  moneyness_col = "option_log_forward_moneyness",
  iv_col = "iv",
  target_abs_moneyness = 0.1,
  long_iv_threshold = 0.2,
  short_iv_threshold = 0.4,
  target_size = 1,
  compute_features = TRUE,
  strat_id = 704L,
  tol_pos = 0.1,
  debug = FALSE
)
```

**Arguments**

DT                    An option-chain 'data.table' when 'compute\_features = TRUE', or a summarized 'data.table' containing 'iv\_otm\_avg' when 'compute\_features = FALSE'.

|                      |                                                                                         |
|----------------------|-----------------------------------------------------------------------------------------|
| state                | Named list describing the current trading state.                                        |
| date_col             | Date or timestamp column name.                                                          |
| expiry_col           | Time-to-expiry column name.                                                             |
| type_col             | Option-type column name.                                                                |
| moneyiness_col       | Forward-moneyness feature column name.                                                  |
| iv_col               | Implied-volatility column name.                                                         |
| target_abs_moneyness | Numeric target absolute log-forward-moneyness used by 'calc_option_iv_skew()'.          |
| long_iv_threshold    | Numeric threshold at or below which OTM IV is treated as cheap enough to go long.       |
| short_iv_threshold   | Numeric threshold at or above which OTM IV is treated as rich enough to go short.       |
| target_size          | Numeric absolute target exposure.                                                       |
| compute_features     | Logical; when 'TRUE', the chain is summarized before generating targets.                |
| strat_id             | Integer strategy identifier recorded on generated actions.                              |
| tol_pos              | Numeric tolerance passed to the action planner.                                         |
| debug                | Logical; when 'TRUE', returns a list with the generated target vector and summary data. |

**Value**

A list produced by 'gen\_action\_plan\_rcpp()'.

---

strat\_strangle\_tgt\_pos

*Strangle Target Positions*

---

**Description**

Generates a simple strangle-proxy target-position path from the average front-expiry OTM implied volatility. Low OTM IV targets a long strangle proxy exposure, and high OTM IV targets a short strangle proxy exposure.

**Usage**

```
strat_strangle_tgt_pos(
  DT,
  date_col = "date",
  expiry_col = "time_to_expiry",
  type_col = "type",
```

```

    moneyness_col = "option_log_forward_moneyness",
    iv_col = "iv",
    target_abs_moneyness = 0.1,
    long_iv_threshold = 0.2,
    short_iv_threshold = 0.4,
    target_size = 1,
    compute_features = TRUE,
    debug = FALSE
)

```

### Arguments

|                      |                                                                                                                                                    |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| DT                   | An option-chain 'data.table' when 'compute_features = TRUE', or a summarized 'data.table' containing 'iv_otm_avg' when 'compute_features = FALSE'. |
| date_col             | Date or timestamp column name.                                                                                                                     |
| expiry_col           | Time-to-expiry column name.                                                                                                                        |
| type_col             | Option-type column name.                                                                                                                           |
| moneyness_col        | Forward-moneyness feature column name.                                                                                                             |
| iv_col               | Implied-volatility column name.                                                                                                                    |
| target_abs_moneyness | Numeric target absolute log-forward-moneyness used by 'calc_option_iv_skew()'.                                                                     |
| long_iv_threshold    | Numeric threshold at or below which OTM IV is treated as cheap enough to go long.                                                                  |
| short_iv_threshold   | Numeric threshold at or above which OTM IV is treated as rich enough to go short.                                                                  |
| target_size          | Numeric absolute target exposure.                                                                                                                  |
| compute_features     | Logical; when 'TRUE', the chain is summarized before generating targets.                                                                           |
| debug                | Logical; when 'TRUE', returns a list with the generated target vector and summary data.                                                            |

### Value

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

strat\_trend\_pullback\_action\_plan

*Trend-Pullback Action Plan*

---

### Description

Applies the trend-pullback rule to the latest bar and translates the resulting target exposure into an executable action plan.

**Usage**

```

strat_trend_pullback_action_plan(
  DT,
  state,
  trend_n = 20L,
  rsi_n = 14L,
  pullback_long = 40,
  pullback_short = 60,
  exit_rsi = 50,
  target_size = 1,
  compute_features = TRUE,
  strat_id = 403L,
  tol_pos = 0.1,
  debug = FALSE
)

```

**Arguments**

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle ‘data.table’.                                                                          |
| state            | Named list describing the current trading state.                                                |
| trend_n          | Integer EMA window used for the trend filter.                                                   |
| rsi_n            | Integer RSI window.                                                                             |
| pullback_long    | Numeric RSI threshold used to enter long pullbacks in an uptrend.                               |
| pullback_short   | Numeric RSI threshold used to enter short pullbacks in a downtrend.                             |
| exit_rsi         | Numeric RSI level used to exit pullback trades.                                                 |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when ‘TRUE’, missing EMA and RSI features are added to ‘DT’ in place.                  |
| strat_id         | Integer strategy identifier recorded on generated actions.                                      |
| tol_pos          | Numeric tolerance passed to the action planner.                                                 |
| debug            | Logical; when ‘TRUE’, returns a list with the generated target vector and feature column names. |

**Value**

A list produced by ‘gen\_action\_plan\_rcpp()’.

---

strat\_trend\_pullback\_atr\_action\_plan

*Trend-Pullback-ATR Action Plan*


---

## Description

Applies the trend-pullback-plus-ATR-zone rule to the latest bar and translates the resulting target exposure into an executable action plan.

## Usage

```
strat_trend_pullback_atr_action_plan(
  DT,
  state,
  trend_n = 20L,
  rsi_n = 14L,
  atr_n = 14L,
  pullback_long = 40,
  pullback_short = 60,
  exit_rsi = 50,
  min_atr_pullback = 0.5,
  max_atr_pullback = 3,
  target_size = 1,
  compute_features = TRUE,
  strat_id = 404L,
  tol_pos = 0.1,
  debug = FALSE
)
```

## Arguments

|                  |                                                                     |
|------------------|---------------------------------------------------------------------|
| DT               | A candle 'data.table'.                                              |
| state            | Named list describing the current trading state.                    |
| trend_n          | Integer EMA window used for the trend filter.                       |
| rsi_n            | Integer RSI window.                                                 |
| atr_n            | Integer ATR window.                                                 |
| pullback_long    | Numeric RSI threshold used to enter long pullbacks in an uptrend.   |
| pullback_short   | Numeric RSI threshold used to enter short pullbacks in a downtrend. |
| exit_rsi         | Numeric RSI level used to exit pullback trades.                     |
| min_atr_pullback | Numeric minimum ATR pullback distance.                              |
| max_atr_pullback | Numeric maximum ATR pullback distance.                              |
| target_size      | Numeric absolute target exposure.                                   |

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| compute_features | Logical; when ‘TRUE’, missing EMA, RSI, and ATR features are added to ‘DT’ in place.            |
| strat_id         | Integer strategy identifier recorded on generated actions.                                      |
| tol_pos          | Numeric tolerance passed to the action planner.                                                 |
| debug            | Logical; when ‘TRUE’, returns a list with the generated target vector and feature column names. |

**Value**

A list produced by ‘gen\_action\_plan\_rcpp()’.

---

|                                            |
|--------------------------------------------|
| strat_trend_pullback_atr_tgt_pos           |
| <i>Trend-Pullback-ATR Target Positions</i> |

---

**Description**

Generates a trend-following pullback target-position path that requires the pullback to reach a minimum ATR distance from the EMA trend anchor.

**Usage**

```
strat_trend_pullback_atr_tgt_pos(  
  DT,  
  trend_n = 20L,  
  rsi_n = 14L,  
  atr_n = 14L,  
  pullback_long = 40,  
  pullback_short = 60,  
  exit_rsi = 50,  
  min_atr_pullback = 0.5,  
  max_atr_pullback = 3,  
  target_size = 1,  
  compute_features = TRUE,  
  debug = FALSE  
)
```

**Arguments**

|               |                                                                   |
|---------------|-------------------------------------------------------------------|
| DT            | A candle ‘data.table’.                                            |
| trend_n       | Integer EMA window used for the trend filter.                     |
| rsi_n         | Integer RSI window.                                               |
| atr_n         | Integer ATR window.                                               |
| pullback_long | Numeric RSI threshold used to enter long pullbacks in an uptrend. |

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| pullback_short   | Numeric RSI threshold used to enter short pullbacks in a downtrend.                             |
| exit_rsi         | Numeric RSI level used to exit pullback trades.                                                 |
| min_atr_pullback | Numeric minimum ATR pullback distance.                                                          |
| max_atr_pullback | Numeric maximum ATR pullback distance.                                                          |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing EMA, RSI, and ATR features are added to 'DT' in place.            |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

strat\_trend\_pullback\_tgt\_pos

*Trend-Pullback Target Positions*

---

**Description**

Generates a simple trend-following pullback target-position path. The trend filter is price relative to an EMA, and entries occur when RSI pulls back within that trend direction.

**Usage**

```
strat_trend_pullback_tgt_pos(
  DT,
  trend_n = 20L,
  rsi_n = 14L,
  pullback_long = 40,
  pullback_short = 60,
  exit_rsi = 50,
  target_size = 1,
  compute_features = TRUE,
  debug = FALSE
)
```

**Arguments**

|         |                                               |
|---------|-----------------------------------------------|
| DT      | A candle 'data.table'.                        |
| trend_n | Integer EMA window used for the trend filter. |
| rsi_n   | Integer RSI window.                           |

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| pullback_long    | Numeric RSI threshold used to enter long pullbacks in an uptrend.                               |
| pullback_short   | Numeric RSI threshold used to enter short pullbacks in a downtrend.                             |
| exit_rsi         | Numeric RSI level used to exit pullback trades.                                                 |
| target_size      | Numeric absolute target exposure.                                                               |
| compute_features | Logical; when 'TRUE', missing EMA and RSI features are added to 'DT' in place.                  |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

### Value

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

strat\_vertical\_spread\_action\_plan  
*Vertical-Spread Action Plan*

---

### Description

Applies the vertical-spread proxy rule to the latest summarized row and translates the resulting target exposure into an executable action plan.

### Usage

```
strat_vertical_spread_action_plan(
  DT,
  state,
  date_col = "date",
  expiry_col = "time_to_expiry",
  type_col = "type",
  moneyness_col = "option_log_forward_moneyness",
  iv_col = "iv",
  long_threshold = 0.02,
  short_threshold = -0.02,
  target_size = 1,
  compute_features = TRUE,
  strat_id = 705L,
  tol_pos = 0.1,
  debug = FALSE
)
```

**Arguments**

|                  |                                                                                                                                                            |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DT               | An option-chain 'data.table' when 'compute_features = TRUE', or a summarized 'data.table' containing 'iv_put_call_spread' when 'compute_features = FALSE'. |
| state            | Named list describing the current trading state.                                                                                                           |
| date_col         | Date or timestamp column name.                                                                                                                             |
| expiry_col       | Time-to-expiry column name.                                                                                                                                |
| type_col         | Option-type column name.                                                                                                                                   |
| moneyness_col    | Forward-moneyness feature column name.                                                                                                                     |
| iv_col           | Implied-volatility column name.                                                                                                                            |
| long_threshold   | Numeric threshold above which put-call IV spread is treated as bearish enough to go short.                                                                 |
| short_threshold  | Numeric threshold below which put-call IV spread is treated as bullish enough to go long.                                                                  |
| target_size      | Numeric absolute target exposure.                                                                                                                          |
| compute_features | Logical; when 'TRUE', the chain is summarized before generating targets.                                                                                   |
| strat_id         | Integer strategy identifier recorded on generated actions.                                                                                                 |
| tol_pos          | Numeric tolerance passed to the action planner.                                                                                                            |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and summary data.                                                                    |

**Value**

A list produced by 'gen\_action\_plan\_rcpp()'.

---

strat\_vertical\_spread\_tgt\_pos  
*Vertical-Spread Target Positions*

---

**Description**

Generates a simple vertical-spread proxy target-position path from the ATM put-minus-call implied-volatility spread. Positive put-call IV spread above the long threshold targets a bearish vertical-spread proxy exposure, and negative spread below the short threshold targets a bullish exposure.

**Usage**

```
strat_vertical_spread_tgt_pos(
  DT,
  date_col = "date",
  expiry_col = "time_to_expiry",
  type_col = "type",
  moneyness_col = "option_log_forward_moneyness",
  iv_col = "iv",
  long_threshold = 0.02,
  short_threshold = -0.02,
  target_size = 1,
  compute_features = TRUE,
  debug = FALSE
)
```

**Arguments**

|                  |                                                                                                                                                            |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DT               | An option-chain ‘data.table’ when ‘compute_features = TRUE’, or a summarized ‘data.table’ containing ‘iv_put_call_spread’ when ‘compute_features = FALSE’. |
| date_col         | Date or timestamp column name.                                                                                                                             |
| expiry_col       | Time-to-expiry column name.                                                                                                                                |
| type_col         | Option-type column name.                                                                                                                                   |
| moneyness_col    | Forward-moneyness feature column name.                                                                                                                     |
| iv_col           | Implied-volatility column name.                                                                                                                            |
| long_threshold   | Numeric threshold above which put-call IV spread is treated as bearish enough to go short.                                                                 |
| short_threshold  | Numeric threshold below which put-call IV spread is treated as bullish enough to go long.                                                                  |
| target_size      | Numeric absolute target exposure.                                                                                                                          |
| compute_features | Logical; when ‘TRUE’, the chain is summarized before generating targets.                                                                                   |
| debug            | Logical; when ‘TRUE’, returns a list with the generated target vector and summary data.                                                                    |

**Value**

A numeric vector of target positions, or a list when ‘debug = TRUE’.

---

```
strat_vol_carry_action_plan
```

*Volatility-Carry Action Plan*

---

### Description

Applies the volatility-carry rule to the latest row and translates the resulting target exposure into an executable action plan.

### Usage

```
strat_vol_carry_action_plan(
  DT,
  state,
  iv_col = "iv",
  rv_n = 20L,
  annualization = 252,
  long_threshold = 0,
  short_threshold = 0,
  target_size = 1,
  compute_features = TRUE,
  strat_id = 707L,
  tol_pos = 0.1,
  debug = FALSE
)
```

### Arguments

|                  |                                                                                                        |
|------------------|--------------------------------------------------------------------------------------------------------|
| DT               | A 'data.table' containing 'close' and an implied-volatility column.                                    |
| state            | Named list describing the current trading state.                                                       |
| iv_col           | Implied-volatility column name.                                                                        |
| rv_n             | Integer realized-volatility window.                                                                    |
| annualization    | Numeric annualization factor passed to 'calc_realized_vol()'.                                          |
| long_threshold   | Numeric threshold above which IV minus RV is treated as rich enough to carry a short-vol exposure.     |
| short_threshold  | Numeric threshold below which IV minus RV is treated as cheap enough to go long volatility.            |
| target_size      | Numeric absolute target exposure.                                                                      |
| compute_features | Logical; when 'TRUE', missing realized-volatility and IV-minus-RV features are added to 'DT' in place. |
| strat_id         | Integer strategy identifier recorded on generated actions.                                             |
| tol_pos          | Numeric tolerance passed to the action planner.                                                        |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names.        |

**Value**

A list produced by ‘gen\_action\_plan\_rcpp()’.

---

|                                          |
|------------------------------------------|
| strat_vol_carry_tgt_pos                  |
| <i>Volatility-Carry Target Positions</i> |

---

**Description**

Generates a target-position path from the gap between implied and realized volatility. Positive target values represent short-vol carry exposure when implied volatility is rich relative to realized volatility, while negative targets represent long-vol exposure when implied volatility is cheap.

**Usage**

```
strat_vol_carry_tgt_pos(  
  DT,  
  iv_col = "iv",  
  rv_n = 20L,  
  annualization = 252,  
  long_threshold = 0,  
  short_threshold = 0,  
  target_size = 1,  
  compute_features = TRUE,  
  debug = FALSE  
)
```

**Arguments**

|                  |                                                                                                        |
|------------------|--------------------------------------------------------------------------------------------------------|
| DT               | A ‘data.table’ containing ‘close’ and an implied-volatility column.                                    |
| iv_col           | Implied-volatility column name.                                                                        |
| rv_n             | Integer realized-volatility window.                                                                    |
| annualization    | Numeric annualization factor passed to ‘calc_realized_vol()’.                                          |
| long_threshold   | Numeric threshold above which IV minus RV is treated as rich enough to carry a short-vol exposure.     |
| short_threshold  | Numeric threshold below which IV minus RV is treated as cheap enough to go long volatility.            |
| target_size      | Numeric absolute target exposure.                                                                      |
| compute_features | Logical; when ‘TRUE’, missing realized-volatility and IV-minus-RV features are added to ‘DT’ in place. |
| debug            | Logical; when ‘TRUE’, returns a list with the generated target vector and feature column names.        |

**Value**

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

|                                 |
|---------------------------------|
| strat_vol_target_action_plan    |
| <i>Vol-Targeted Action Plan</i> |

---

**Description**

Applies the volatility-targeted rule to the latest bar and translates the resulting target exposure into an executable action plan.

**Usage**

```
strat_vol_target_action_plan(  
  DT,  
  state,  
  trend_n = 20L,  
  rv_n = 20L,  
  vol_target = 0.2,  
  max_leverage = 1,  
  annualization = 252,  
  compute_features = TRUE,  
  strat_id = 402L,  
  tol_pos = 0.1,  
  debug = FALSE  
)
```

**Arguments**

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle 'data.table'.                                                                          |
| state            | Named list describing the current trading state.                                                |
| trend_n          | Integer EMA window used for the directional trend filter.                                       |
| rv_n             | Integer realized-volatility window.                                                             |
| vol_target       | Numeric annualized target volatility.                                                           |
| max_leverage     | Numeric cap on absolute target exposure.                                                        |
| annualization    | Numeric annualization factor passed to 'calc_realized_vol()'.                                   |
| compute_features | Logical; when 'TRUE', missing EMA and realized-vol features are added to 'DT' in place.         |
| strat_id         | Integer strategy identifier recorded on generated actions.                                      |
| tol_pos          | Numeric tolerance passed to the action planner.                                                 |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A list produced by 'gen\_action\_plan\_rcpp()'.

---

```
strat_vol_target_regime_floor_action_plan
```

*Vol-Target-Regime-Floor Action Plan*

---

**Description**

Applies the volatility-targeted rule with a volatility ceiling to the latest bar and translates the resulting target exposure into an executable action plan.

**Usage**

```
strat_vol_target_regime_floor_action_plan(
  DT,
  state,
  trend_n = 20L,
  rv_n = 20L,
  vol_target = 0.2,
  max_leverage = 1,
  rv_ceiling = 0.4,
  annualization = 252,
  compute_features = TRUE,
  strat_id = 406L,
  tol_pos = 0.1,
  debug = FALSE
)
```

**Arguments**

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle 'data.table'.                                                                          |
| state            | Named list describing the current trading state.                                                |
| trend_n          | Integer EMA window used for the directional trend filter.                                       |
| rv_n             | Integer realized-volatility window.                                                             |
| vol_target       | Numeric annualized target volatility.                                                           |
| max_leverage     | Numeric cap on absolute target exposure.                                                        |
| rv_ceiling       | Numeric maximum realized volatility allowed for active positions.                               |
| annualization    | Numeric annualization factor passed to 'calc_realized_vol()'.                                   |
| compute_features | Logical; when 'TRUE', missing EMA and realized-vol features are added to 'DT' in place.         |
| strat_id         | Integer strategy identifier recorded on generated actions.                                      |
| tol_pos          | Numeric tolerance passed to the action planner.                                                 |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

**Value**

A list produced by ‘gen\_action\_plan\_rcpp()’.

---

```
strat_vol_target_regime_floor_tgt_pos
```

*Vol-Target-Regime-Floor Target Positions*

---

**Description**

Generates a volatility-targeted target-position path that goes flat when realized volatility exceeds a ceiling. Direction comes from price relative to a trend EMA and size is otherwise scaled by target volatility over realized volatility.

**Usage**

```
strat_vol_target_regime_floor_tgt_pos(  
  DT,  
  trend_n = 20L,  
  rv_n = 20L,  
  vol_target = 0.2,  
  max_leverage = 1,  
  rv_ceiling = 0.4,  
  annualization = 252,  
  compute_features = TRUE,  
  debug = FALSE  
)
```

**Arguments**

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle ‘data.table’.                                                                          |
| trend_n          | Integer EMA window used for the directional trend filter.                                       |
| rv_n             | Integer realized-volatility window.                                                             |
| vol_target       | Numeric annualized target volatility.                                                           |
| max_leverage     | Numeric cap on absolute target exposure.                                                        |
| rv_ceiling       | Numeric maximum realized volatility allowed for active positions.                               |
| annualization    | Numeric annualization factor passed to ‘calc_realized_vol()’.                                   |
| compute_features | Logical; when ‘TRUE’, missing EMA and realized-vol features are added to ‘DT’ in place.         |
| debug            | Logical; when ‘TRUE’, returns a list with the generated target vector and feature column names. |

**Value**

A numeric vector of target positions, or a list when ‘debug = TRUE’.

---

strat\_vol\_target\_tgt\_pos

*Vol-Targeted Target Positions*


---

## Description

Generates a simple volatility-targeted target-position path. Direction is determined by price relative to a trend EMA, and exposure size is scaled by the ratio of target volatility to realized volatility.

## Usage

```
strat_vol_target_tgt_pos(
  DT,
  trend_n = 20L,
  rv_n = 20L,
  vol_target = 0.2,
  max_leverage = 1,
  annualization = 252,
  compute_features = TRUE,
  debug = FALSE
)
```

## Arguments

|                  |                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------|
| DT               | A candle 'data.table'.                                                                          |
| trend_n          | Integer EMA window used for the directional trend filter.                                       |
| rv_n             | Integer realized-volatility window.                                                             |
| vol_target       | Numeric annualized target volatility.                                                           |
| max_leverage     | Numeric cap on absolute target exposure.                                                        |
| annualization    | Numeric annualization factor passed to 'calc_realized_vol()'.                                   |
| compute_features | Logical; when 'TRUE', missing EMA and realized-vol features are added to 'DT' in place.         |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names. |

## Value

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

```
strat_vwap_revert_action_plan
      VWAP-Reversion Action Plan
```

---

### Description

Applies the VWAP-reversion rule to the latest row and translates the resulting target exposure into an executable action plan.

### Usage

```
strat_vwap_revert_action_plan(
  DT,
  state,
  vwap_n = 20L,
  entry_dev = 0.01,
  exit_dev = 0.0025,
  min_long_imbalance = -0.2,
  max_short_imbalance = 0.2,
  max_rel_spread = NULL,
  bid_col = "bid",
  ask_col = "ask",
  bid_size_col = "bid_size",
  ask_size_col = "ask_size",
  target_size = 1,
  compute_features = TRUE,
  strat_id = 506L,
  tol_pos = 0.1,
  debug = FALSE
)
```

### Arguments

|                     |                                                                                                            |
|---------------------|------------------------------------------------------------------------------------------------------------|
| DT                  | A 'data.table' containing at least 'close' and 'volume'.                                                   |
| state               | Named list describing the current trading state.                                                           |
| vwap_n              | Integer rolling VWAP window.                                                                               |
| entry_dev           | Numeric absolute VWAP-deviation threshold used for entries.                                                |
| exit_dev            | Numeric absolute VWAP-deviation threshold used for exits.                                                  |
| min_long_imbalance  | Minimum acceptable order imbalance for long entries. More negative values allow stronger selling pressure. |
| max_short_imbalance | Maximum acceptable order imbalance for short entries. More positive values allow stronger buying pressure. |
| max_rel_spread      | Optional maximum relative bid-ask spread filter.                                                           |

|                  |                                                                                                     |
|------------------|-----------------------------------------------------------------------------------------------------|
| bid_col          | Bid-price column name used when bid-ask features are missing.                                       |
| ask_col          | Ask-price column name used when bid-ask features are missing.                                       |
| bid_size_col     | Bid-size column name used when imbalance features are missing.                                      |
| ask_size_col     | Ask-size column name used when imbalance features are missing.                                      |
| target_size      | Numeric absolute target exposure.                                                                   |
| compute_features | Logical; when 'TRUE', missing VWAP and optional microstructure features are added to 'DT' in place. |
| strat_id         | Integer strategy identifier recorded on generated actions.                                          |
| tol_pos          | Numeric tolerance passed to the action planner.                                                     |
| debug            | Logical; when 'TRUE', returns a list with the generated target vector and feature column names.     |

**Value**

A list produced by 'gen\_action\_plan\_rcpp()'.

---

strat\_vwap\_revert\_tgt\_pos  
*VWAP-Reversion Target Positions*

---

**Description**

Generates a fair-value reversion target-position path from deviations between price and rolling VWAP. Large positive or negative deviations open a contrarian target, optionally filtered by queue imbalance and relative bid-ask spread.

**Usage**

```
strat_vwap_revert_tgt_pos(
  DT,
  vwap_n = 20L,
  entry_dev = 0.01,
  exit_dev = 0.0025,
  min_long_imbalance = -0.2,
  max_short_imbalance = 0.2,
  max_rel_spread = NULL,
  bid_col = "bid",
  ask_col = "ask",
  bid_size_col = "bid_size",
  ask_size_col = "ask_size",
  target_size = 1,
  compute_features = TRUE,
  debug = FALSE
)
```

**Arguments**

|                     |                                                                                                            |
|---------------------|------------------------------------------------------------------------------------------------------------|
| DT                  | A 'data.table' containing at least 'close' and 'volume'.                                                   |
| vwap_n              | Integer rolling VWAP window.                                                                               |
| entry_dev           | Numeric absolute VWAP-deviation threshold used for entries.                                                |
| exit_dev            | Numeric absolute VWAP-deviation threshold used for exits.                                                  |
| min_long_imbalance  | Minimum acceptable order imbalance for long entries. More negative values allow stronger selling pressure. |
| max_short_imbalance | Maximum acceptable order imbalance for short entries. More positive values allow stronger buying pressure. |
| max_rel_spread      | Optional maximum relative bid-ask spread filter.                                                           |
| bid_col             | Bid-price column name used when bid-ask features are missing.                                              |
| ask_col             | Ask-price column name used when bid-ask features are missing.                                              |
| bid_size_col        | Bid-size column name used when imbalance features are missing.                                             |
| ask_size_col        | Ask-size column name used when imbalance features are missing.                                             |
| target_size         | Numeric absolute target exposure.                                                                          |
| compute_features    | Logical; when 'TRUE', missing VWAP and optional microstructure features are added to 'DT' in place.        |
| debug               | Logical; when 'TRUE', returns a list with the generated target vector and feature column names.            |

**Value**

A numeric vector of target positions, or a list when 'debug = TRUE'.

---

summarize\_walk\_forward\_results

*Summarize Walk-Forward Mining Results*


---

**Description**

Aggregates out-of-sample walk-forward test results into a compact diagnostic table. The summary focuses on test-window performance, train/test score decay, and warmup quality.

**Usage**

```
summarize_walk_forward_results(x, score_col = "sortino", group_cols = NULL)
```

**Arguments**

|            |                                                                                                                 |
|------------|-----------------------------------------------------------------------------------------------------------------|
| x          | Result list returned by 'mine_strategy_walk_forward()', or a 'data.table' containing walk-forward test results. |
| score_col  | Metric used as the main score column.                                                                           |
| group_cols | Optional grouping columns, such as parameter columns. When 'NULL', all test rows are summarized together.       |

**Value**

A 'data.table' with window counts, average performance metrics, positive-window rate, average train/test score decay, and warmup counts.

# Index

backtest\_portfolio\_weights, 9  
backtest\_rcpp, 10  
build\_order\_intents, 11

calc\_ADX, 11  
calc\_aroon, 12  
calc\_ATR, 12  
calc\_ATR\_quantile, 13  
calc\_backtest\_performance, 13  
calc\_bid\_ask\_spread, 14  
calc\_BollingerBands, 15  
calc\_bond\_accrued\_interest, 15  
calc\_bond\_carry, 16  
calc\_bond\_carry\_roll\_decomp, 17  
calc\_bond\_cashflows, 17  
calc\_bond\_clean\_price, 18  
calc\_bond\_convexity, 18  
calc\_bond\_current\_yield, 19  
calc\_bond\_dirty\_price, 20  
calc\_bond\_duration, 21  
calc\_bond\_dv01, 21  
calc\_bond\_econvexity, 22  
calc\_bond\_eduration, 23  
calc\_bond\_holding\_period\_return, 24  
calc\_bond\_key\_rate\_duration, 25  
calc\_bond\_key\_rate\_duration\_tenor, 26  
calc\_bond\_mduration, 27  
calc\_bond\_next\_coupon, 27  
calc\_bond\_nominal\_spread, 28  
calc\_bond\_npv, 28  
calc\_bond\_prev\_coupon, 29  
calc\_bond\_price\_change\_approx, 30  
calc\_bond\_pv01, 31  
calc\_bond\_rate\_return\_approx, 31  
calc\_bond\_risk\_state, 32  
calc\_bond\_roll\_down\_return, 33  
calc\_bond\_spread\_duration, 34  
calc\_bond\_spread\_return\_approx, 35  
calc\_bond\_total\_return\_decomp, 35  
calc\_bond\_yield, 36

calc\_bond\_zspread, 37  
calc\_bond\_zspread\_return\_approx, 38  
calc\_breadth\_ad, 38  
calc\_breadth\_adl, 39  
calc\_breadth\_high\_low, 39  
calc\_breadth\_ratio, 40  
calc\_breadth\_trin, 41  
calc\_CCI, 41  
calc\_chaikinAD, 42  
calc\_chaikinVolatility, 42  
calc\_CMF, 43  
calc\_CMO, 43  
calc\_contango\_backwardation, 44  
calc\_coupon\_schedule, 44  
calc\_credit\_spread, 45  
calc\_curve\_butterfly, 46  
calc\_curve\_shock, 46  
calc\_curve\_slope, 47  
calc\_curve\_zero\_rate, 47  
calc\_day\_count\_frac, 48  
calc\_DEMA, 48  
calc\_DonchianChannels, 49  
calc\_EMA, 49  
calc\_EMA\_cross, 50  
calc\_EMV, 51  
calc\_excess\_spread, 51  
calc\_front\_next\_spread, 52  
calc\_futures\_curve\_butterfly, 53  
calc\_futures\_curve\_slope, 53  
calc\_fx\_basis, 54  
calc\_fx\_carry, 55  
calc\_fx\_forward\_points, 55  
calc\_fx\_realized\_carry, 56  
calc\_HMA, 57  
calc\_KeltnerChannels, 57  
calc\_KST, 58  
calc\_kurtosis, 59  
calc\_ladder\_index, 59  
calc\_log\_spread, 60

[calc\\_MACD, 61](#)  
[calc\\_MFI, 61](#)  
[calc\\_microprice, 62](#)  
[calc\\_mid\\_price, 62](#)  
[calc\\_normalize, 63](#)  
[calc\\_OBV, 63](#)  
[calc\\_option\\_delta, 64](#)  
[calc\\_option\\_forward\\_moneyness, 64](#)  
[calc\\_option\\_gamma, 65](#)  
[calc\\_option\\_greeks, 66](#)  
[calc\\_option\\_iv, 67](#)  
[calc\\_option\\_iv\\_skew, 68](#)  
[calc\\_option\\_iv\\_term\\_structure, 69](#)  
[calc\\_option\\_moneyness, 69](#)  
[calc\\_option\\_put\\_call\\_iv\\_spread, 70](#)  
[calc\\_option\\_rho, 71](#)  
[calc\\_option\\_risk\\_state, 71](#)  
[calc\\_option\\_smile\\_slope, 72](#)  
[calc\\_option\\_theta, 73](#)  
[calc\\_option\\_vega, 74](#)  
[calc\\_order\\_imbalance, 74](#)  
[calc\\_percent\\_rank, 75](#)  
[calc\\_period\\_rate, 75](#)  
[calc\\_position\\_greeks, 76](#)  
[calc\\_price\\_impact\\_proxy, 76](#)  
[calc\\_ratio, 77](#)  
[calc\\_realized\\_vol, 78](#)  
[calc\\_relative\\_strength, 78](#)  
[calc\\_ROC, 79](#)  
[calc\\_roll\\_yield, 81](#)  
[calc\\_rolling\\_beta, 79](#)  
[calc\\_rolling\\_corr, 80](#)  
[calc\\_rolling\\_quantile, 81](#)  
[calc\\_RSI, 82](#)  
[calc\\_SAR, 83](#)  
[calc\\_skewness, 83](#)  
[calc\\_slippage\\_proxy, 84](#)  
[calc\\_SMA, 84](#)  
[calc\\_SMI, 85](#)  
[calc\\_spread, 85](#)  
[calc\\_spread\\_curve\\_butterfly, 86](#)  
[calc\\_spread\\_curve\\_slope, 86](#)  
[calc\\_StochasticOscillator, 87](#)  
[calc\\_strategy\\_performance\\_summary, 88](#)  
[calc\\_term\\_structure\\_carry, 89](#)  
[calc\\_tracking\\_error, 90](#)  
[calc\\_TRIX, 90](#)  
[calc\\_turnover, 91](#)

[calc\\_ultimateOscillator, 92](#)  
[calc\\_VWAP, 92](#)  
[calc\\_VWMA, 93](#)  
[calc\\_WMA, 93](#)  
[calc\\_WPR, 94](#)  
[calc\\_ZLEMA, 94](#)  
[calc\\_zscore, 95](#)  
[compare\\_strategy\\_backtests, 95](#)  
  
[filter\\_walk\\_forward\\_results, 96](#)  
  
[gen\\_action\\_plan\\_rcpp, 97](#)  
  
[mine\\_strategy\\_asset\\_years, 100](#)  
[mine\\_strategy\\_assets, 98](#)  
[mine\\_strategy\\_params, 102](#)  
[mine\\_strategy\\_walk\\_forward, 103](#)  
  
[plan\\_curve\\_trade\\_adjustment, 105](#)  
[plan\\_delta\\_neutral\\_adjustment, 105](#)  
[plan\\_duration\\_neutral\\_adjustment, 106](#)  
[plan\\_portfolio\\_adjustment, 106](#)  
[plan\\_vega\\_target\\_adjustment, 107](#)  
  
[rank\\_strategy\\_results, 108](#)  
  
[select\\_strategy\\_params, 108](#)  
[strat\\_atr\\_breakout\\_action\\_plan, 110](#)  
[strat\\_atr\\_breakout\\_tgt\\_pos, 111](#)  
[strat\\_atr\\_breakout\\_trailing\\_stop\\_action\\_plan, 112](#)  
[strat\\_atr\\_breakout\\_trailing\\_stop\\_tgt\\_pos, 113](#)  
[strat\\_bollinger\\_low\\_adx\\_revert\\_action\\_plan, 114](#)  
[strat\\_bollinger\\_low\\_adx\\_revert\\_tgt\\_pos, 115](#)  
[strat\\_bollinger\\_revert\\_action\\_plan, 116](#)  
[strat\\_bollinger\\_revert\\_rsi\\_action\\_plan, 117](#)  
[strat\\_bollinger\\_revert\\_rsi\\_tgt\\_pos, 118](#)  
[strat\\_bollinger\\_revert\\_tgt\\_pos, 119](#)  
[strat\\_bollinger\\_squeeze\\_breakout\\_action\\_plan, 120](#)  
[strat\\_bollinger\\_squeeze\\_breakout\\_tgt\\_pos, 121](#)  
[strat\\_bond\\_carry\\_roll\\_action\\_plan, 122](#)

- strat\_bond\_carry\_roll\_duration\_cap\_action\_plan, 123
- strat\_bond\_carry\_roll\_duration\_cap\_tgt\_pos, 125
- strat\_bond\_carry\_roll\_spread\_filter\_action\_plan, 126
- strat\_bond\_carry\_roll\_spread\_filter\_tgt\_pos, 128
- strat\_bond\_carry\_roll\_tgt\_pos, 129
- strat\_buy\_and\_hold\_action\_plan, 131
- strat\_buy\_and\_hold\_tgt\_pos, 131
- strat\_credit\_spread\_momentum\_action\_plan, 132
- strat\_credit\_spread\_momentum\_tgt\_pos, 133
- strat\_credit\_spread\_revert\_action\_plan, 134
- strat\_credit\_spread\_revert\_tgt\_pos, 136
- strat\_cross\_sectional\_rank\_allocator\_action\_plan, 137
- strat\_cross\_sectional\_rank\_allocator\_tgt\_pos, 138
- strat\_curve\_butterfly\_action\_plan, 139
- strat\_curve\_butterfly\_tgt\_pos, 140
- strat\_curve\_steepener\_action\_plan, 141
- strat\_curve\_steepener\_tgt\_pos, 142
- strat\_curve\_steepener\_zscore\_action\_plan, 143
- strat\_curve\_steepener\_zscore\_tgt\_pos, 144
- strat\_donchian\_breakout\_action\_plan, 145
- strat\_donchian\_breakout\_tgt\_pos, 146
- strat\_donchian\_retest\_breakout\_action\_plan, 147
- strat\_donchian\_retest\_breakout\_tgt\_pos, 148
- strat\_donchian\_turtle\_action\_plan, 149
- strat\_donchian\_turtle\_tgt\_pos, 150
- strat\_ema\_cross\_action\_plan, 150
- strat\_ema\_cross\_adx\_action\_plan, 152
- strat\_ema\_cross\_adx\_tgt\_pos, 153
- strat\_ema\_cross\_slope\_confirm\_action\_plan, 154
- strat\_ema\_cross\_slope\_confirm\_tgt\_pos, 155
- strat\_ema\_cross\_tgt\_pos, 156
- strat\_ema\_triple\_trend\_action\_plan, 157
- strat\_ema\_triple\_trend\_tgt\_pos, 158
- strat\_funding\_basis\_convergence\_action\_plan, 159
- strat\_funding\_basis\_convergence\_tgt\_pos, 160
- strat\_fx\_carry\_action\_plan, 162
- strat\_fx\_carry\_basket\_rank\_action\_plan, 163
- strat\_fx\_carry\_basket\_rank\_tgt\_pos, 164
- strat\_fx\_carry\_tgt\_pos, 165
- strat\_fx\_carry\_trend\_action\_plan, 166
- strat\_fx\_carry\_trend\_tgt\_pos, 167
- strat\_gamma\_scalp\_support\_action\_plan, 168
- strat\_gamma\_scalp\_support\_tgt\_pos, 170
- strat\_iv\_directional\_overlay\_action\_plan, 171
- strat\_iv\_directional\_overlay\_tgt\_pos, 172
- strat\_iv\_skew\_action\_plan, 174
- strat\_iv\_skew\_realized\_vol\_confirm\_action\_plan, 175
- strat\_iv\_skew\_realized\_vol\_confirm\_tgt\_pos, 176
- strat\_iv\_skew\_tgt\_pos, 178
- strat\_iv\_skew\_zscore\_action\_plan, 179
- strat\_iv\_skew\_zscore\_tgt\_pos, 180
- strat\_iv\_term\_structure\_action\_plan, 181
- strat\_iv\_term\_structure\_tgt\_pos, 183
- strat\_ladder\_bounce\_action\_plan, 184
- strat\_ladder\_bounce\_tgt\_pos, 185
- strat\_ladder\_breakout\_action\_plan, 186
- strat\_ladder\_breakout\_tgt\_pos, 187
- strat\_macd\_contrarian\_action\_plan, 188
- strat\_macd\_contrarian\_tgt\_pos, 189
- strat\_macd\_cross\_action\_plan, 190
- strat\_macd\_cross\_tgt\_pos, 191
- strat\_macd\_histogram\_momentum\_action\_plan, 192
- strat\_macd\_histogram\_momentum\_tgt\_pos, 193
- strat\_macd\_zero\_line\_action\_plan, 194
- strat\_macd\_zero\_line\_tgt\_pos, 195
- strat\_pair\_spread\_bollinger\_action\_plan,

[196](#)  
 strat\_pair\_spread\_bollinger\_tgt\_pos, [197](#)  
 strat\_pair\_spread\_half\_life\_revert\_action\_plan, [198](#)  
 strat\_pair\_spread\_half\_life\_revert\_tgt\_pos, [199](#)  
 strat\_pair\_spread\_revert\_action\_plan, [200](#)  
 strat\_pair\_spread\_revert\_tgt\_pos, [202](#)  
 strat\_ratio\_revert\_action\_plan, [203](#)  
 strat\_ratio\_revert\_tgt\_pos, [204](#)  
 strat\_regime\_switch\_action\_plan, [205](#)  
 strat\_regime\_switch\_tgt\_pos, [206](#)  
 strat\_relative\_strength\_action\_plan, [208](#)  
 strat\_relative\_strength\_dual\_momentum\_action\_plan, [209](#)  
 strat\_relative\_strength\_dual\_momentum\_tgt\_pos, [210](#)  
 strat\_relative\_strength\_persistence\_action\_plan, [211](#)  
 strat\_relative\_strength\_persistence\_tgt\_pos, [213](#)  
 strat\_relative\_strength\_tgt\_pos, [214](#)  
 strat\_roll\_yield\_action\_plan, [215](#)  
 strat\_roll\_yield\_cross\_sectional\_action\_plan, [216](#)  
 strat\_roll\_yield\_cross\_sectional\_tgt\_pos, [217](#)  
 strat\_roll\_yield\_mean\_revert\_action\_plan, [218](#)  
 strat\_roll\_yield\_mean\_revert\_tgt\_pos, [219](#)  
 strat\_roll\_yield\_momentum\_action\_plan, [221](#)  
 strat\_roll\_yield\_momentum\_tgt\_pos, [222](#)  
 strat\_roll\_yield\_tgt\_pos, [224](#)  
 strat\_rsi\_divergence\_action\_plan, [225](#)  
 strat\_rsi\_divergence\_tgt\_pos, [226](#)  
 strat\_rsi\_dynamic\_threshold\_revert\_action\_plan, [227](#)  
 strat\_rsi\_dynamic\_threshold\_revert\_tgt\_pos, [228](#)  
 strat\_rsi\_logr\_revert\_action\_plan, [229](#)  
 strat\_rsi\_logr\_revert\_tgt\_pos, [230](#)  
 strat\_rsi\_revert\_action\_plan, [231](#)  
 strat\_rsi\_revert\_tgt\_pos, [232](#)  
 strat\_rsi\_trend\_aware\_revert\_action\_plan, [233](#)  
 strat\_rsi\_trend\_aware\_revert\_tgt\_pos, [234](#)  
 strat\_straddle\_action\_plan, [235](#)  
 strat\_straddle\_tgt\_pos, [236](#)  
 strat\_strangle\_action\_plan, [237](#)  
 strat\_strangle\_tgt\_pos, [238](#)  
 strat\_trend\_pullback\_action\_plan, [239](#)  
 strat\_trend\_pullback\_atr\_action\_plan, [241](#)  
 strat\_trend\_pullback\_atr\_tgt\_pos, [242](#)  
 strat\_trend\_pullback\_tgt\_pos, [243](#)  
 strat\_vertical\_spread\_action\_plan, [244](#)  
 strat\_vertical\_spread\_tgt\_pos, [245](#)  
 strat\_vol\_carry\_action\_plan, [247](#)  
 strat\_vol\_carry\_tgt\_pos, [248](#)  
 strat\_vol\_target\_action\_plan, [249](#)  
 strat\_vol\_target\_regime\_floor\_action\_plan, [250](#)  
 strat\_vol\_target\_regime\_floor\_tgt\_pos, [251](#)  
 strat\_vol\_target\_tgt\_pos, [252](#)  
 strat\_vwap\_revert\_action\_plan, [253](#)  
 strat\_vwap\_revert\_tgt\_pos, [254](#)  
 strategy\_monitor\_definition, [109](#)  
 strategy\_public\_definition, [110](#)  
 summarize\_walk\_forward\_results, [255](#)