Fundamental Growth Scoring Algorithm Description (v2 - Based on Code)

Objective: To evaluate the growth potential and quality of S&P 500 constituents by calculating a composite score and ranking the companies.

I. Core Indicators & Calculation Methods (Reflecting Code Implementation):

The algorithm assesses companies across several dimensions using the following indicators:

A. Growth Dimension

1. **Revenue_CAGR (Revenue Compound Annual Growth Rate):**
    - **Calculation:**
        1. Retrieves the last N+1 years (years_of_annual_data + 1) of annual revenue data from the database.
        2. Removes any NaN values from this N+1 year series.
        3. Calculates CAGR using the **actual number of available valid data points** (minimum 2 required). For example, 4 valid points yield a 3-year CAGR; 3 valid points yield a 2-year CAGR.
        4. Returns NaN if fewer than 2 valid points remain after removing NaNs.
        5. Logs a warning if the calculated period differs from the configured N years.
    - **Purpose:** Measures the average annualized historical revenue growth rate.
2. **EPS_CAGR (Earnings Per Share Compound Annual Growth Rate):**
    - **Calculation:**
        1. Retrieves the last N+1 years of annual eps data.
        2. Removes NaN values.
        3. Returns NaN if fewer than 2 valid points remain or if the starting EPS is <= min_eps_for_cagr (from config).
        4. Calculates CAGR using the **actual number of available valid data points**.
        5. Logs a warning if the calculated period differs from the configured N years.
    - **Purpose:** Measures the average annualized historical growth rate of profitability per share.
3. **Slope_Revenue (Revenue Quarterly Growth Momentum - Slope):**
    - **Calculation:**
        1. Calculates the quarter-over-quarter (QoQ) percentage change for revenue (SeqGrowth_Rev).
        2. Replaces any resulting inf or -inf values with NaN.

3. Takes the most recent 4 **valid (non-NaN)** QoQ growth rates.
4. Performs linear regression on these valid rates (requires at least 2 points). The result is the slope (b) of the regression line.
5. Returns NaN if fewer than 2 valid QoQ rates are available for regression.

- **Purpose:** Captures the recent acceleration or deceleration trend in revenue growth.

4. **Slope_EPS (EPS Quarterly Growth Momentum - Slope):**
   - **Calculation:**
     1. Calculates the QoQ percentage change for eps (SeqGrowth_EPS).
     2. Handles zero or negative prior-quarter EPS based on eps_qoq_denominator_handling config (defaults to NaN or uses eps_qoq_zero_value).
     3. Replaces any resulting inf or -inf values with NaN.
     4. Takes the most recent 4 **valid (non-NaN)** QoQ growth rates.
     5. Performs linear regression (requires at least 2 points). The result is the slope (b).
     6. Returns NaN if fewer than 2 valid QoQ rates are available.
   - **Purpose:** Captures the recent acceleration or deceleration trend in profitability growth.

B. Profitability & Efficiency Dimension

5. **TTM_OpMargin_Level (Trailing Twelve Month Operating Margin):**
   - **Calculation:**
     1. Calculates the sum of the last 4 quarters' op_income (TTM_OpIncome), requiring 4 valid (non-NaN) data points.
     2. Calculates the sum of the last 4 quarters' revenue (TTM_Revenue), requiring 4 valid data points.
     3. TTM_OpMargin_Level = TTM_OpIncome / TTM_Revenue. Returns NaN if either TTM value is NaN or if TTM_Revenue is zero.
   - **Purpose:** Measures the core business profitability over the last year.
6. **TTM_ROE_Level (Trailing Twelve Month Return on Equity):**
   - **Calculation:**
     1. Calculates the sum of the last 4 quarters' net_income (TTM_NetIncome), requiring 4 valid data points.
     2. Calculates the average of the last 5 quarters' ending equity (Avg_Equity_5Q), requiring 5 valid data points.
     3. TTM_ROE_Level = TTM_NetIncome / Avg_Equity_5Q. Returns NaN if either TTM_NetIncome or Avg_Equity_5Q is NaN, or if Avg_Equity_5Q is zero.
   - **Purpose:** Measures the efficiency of using shareholder equity to generate

profit over the last year.

C. Cash Flow Dimension

7. **Annual_FCF_Growth_Slope (Annual Free Cash Flow Growth Slope):**
    - **Calculation:**
        1. Calculates annual Free Cash Flow (FCF = ocf - capex). NaN OCF/CapEx are treated as 0.
        2. Retrieves the last N+1 years (years_of_annual_data + 1) of annual FCF data.
        3. Removes NaN values.
        4. Performs linear regression on the valid FCF sequence (requires at least 2 points). The result is the slope (b).
        5. Returns NaN if fewer than 2 valid FCF points are available.
    - **Purpose:** Measures the trend in the company's ability to generate discretionary cash flow.

D. Financial Health Dimension (Used for Screening)

8. **Debt_to_Equity_Ratio (Latest Quarter-End):**
    - **Calculation:** Latest Quarter Total Debt / Latest Quarter Equity. Returns NaN if Equity is zero or NaN. Treats NaN Total Debt as 0.
    - **Purpose:** Measures financial leverage.
9. **Interest_Coverage_Ratio (TTM):**
    - **Calculation:**
        1. Calculates TTM_EBIT (sum of last 4 quarters' ebit, requires 4 valid points).
        2. Calculates TTM_InterestExp (sum of last 4 quarters' interest_exp, requires 4 valid points).
        3. ICR = TTM_EBIT / TTM_InterestExp.
        4. Handles special cases: inf if TTM_InterestExp is 0 and TTM_EBIT >= 0; -inf if TTM_InterestExp is 0 and TTM_EBIT < 0; NaN if TTM_InterestExp < 0 or if either TTM value is NaN. Resulting infinities are later replaced by NaN.
    - **Purpose:** Measures the ability to cover interest payments with operating earnings.

II. Scoring and Ranking Process:

1. **Data Collection & Calculation:** Fetch S&P 500 list/industries, download/update annual and quarterly data into SQLite DB, then calculate all indicators above.
2. **Financial Health Screening (Optional):**
    - Based on [Screening] settings in config_finance.ini.
    - If enabled, mark companies as Screened_Out = True if:
        - Debt_to_Equity_Ratio > max_debt_to_equity (or is NaN).

        - **OR** Interest_Coverage_Ratio < min_interest_coverage (and is **not** NaN - current logic ignores NaN ICR for screening).
3. **Indicator Standardization:**
    - **Applies only to:** Companies not screened out (Screened_Out = False).
    - **Metrics:** Standardizes core performance indicators (Revenue_CAGR, EPS_CAGR, Slope_Revenue, Slope_EPS, TTM_OpMargin_Level, TTM_ROE_Level, Annual_FCF_Growth_Slope).
    - **Method:** Calculates the **percentile rank** (rank(pct=True)) for each metric among the screened-in companies, either 'overall' or grouped by 'industry' based on ranking_method config. Ranks are scaled to 0-100 (higher rank = higher score). NaN metric values result in NaN scores.
4. **Composite Score Calculation:**
    - **NaN Score Handling:** Before final aggregation, any resulting NaN values in the standardized score columns (Score_*) are **imputed with 50**.
    - **Fused Growth Scores:**
        - Score_Growth_Revenue = (w_cagr * Score_Revenue_CAGR) + (w_accel * Score_Revenue_Accel)
        - Score_Growth_EPS = (w_cagr * Score_EPS_CAGR) + (w_accel * Score_EPS_Accel)
        - (Weights w_cagr, w_accel from config)
    - **Final Overall Score (Overall_Score):**
        - Overall_Score = (W_GrowthRev * Score_Growth_Revenue) + (W_GrowthEPS * Score_Growth_EPS) + (W_Profitability * Imputed_Score_TTM_OpMargin) + (W_Efficiency * Imputed_Score_TTM_ROE) + (W_FCF * Imputed_Score_FCF_Slope)
        - (Dimension weights W_* from config; uses imputed scores)
    - **Screened-Out Handling:** For companies marked Screened_Out = True, all Score_* columns and the Overall_Score are **reset to NaN**.
5. **Ranking & Output:**
    - Companies are ranked (Rank) based on Overall_Score in descending order. NaN scores are ranked last.
    - Results (raw indicators, standardized scores, overall score, rank, screening status) are saved to an Excel file.

Note: The algorithm's output depends heavily on the quality and completeness of data from yfinance. The CAGR calculation uses actual available periods, potentially mixing timeframes. The final score is influenced by indicator results, NaN handling (imputation with 50), and configured weights.