**Time Series Feature Explanations**

**1. Trend Strength (trend\_strength)**

* **Practical Usefulness:**
  + **Retail Sales:** A business analyst can use trend strength to assess if an observed increase in monthly sales is a statistically significant upward movement or just part of normal random fluctuations. A strong trend might justify increased inventory orders, while a weak trend might suggest caution.
  + **Stock Market Analysis:** An investor could use trend strength to determine if a stock price is in a strong, reliable uptrend or downtrend, influencing buy/sell decisions, rather than reacting to short-term volatility.
* **Calculation:**
  1. **STL Decomposition:** The time series (Yt) is decomposed into trend (Tt​), seasonal (St​), and remainder (Rt​) components, such that Yt​=Tt​+St​+Rt​ (for additive decomposition). This is done iteratively:
     1. An initial trend is estimated (using a moving average).
     2. The series is detrended (Yt​−Tt​).
     3. The seasonal component is estimated by averaging the detrended series over each seasonal period and then smoothing these seasonal sub-series (using Loess).
     4. The seasonal component is removed from the original series to get a seasonally adjusted series (Yt​−St​).
     5. A new trend component Tt​ is estimated by smoothing the seasonally adjusted series (using Loess).
     6. These steps are repeated a few times for robustness.
  2. **Deseasonalized Series:** The deseasonalized series is calculated as Dt​=Yt​−St​=Tt​+Rt​.
  3. **Variances:**
     1. The variance of the remainder component is calculated: Var(Rt​).
     2. The variance of the deseasonalized series is calculated: Var(Dt​).
  4. **Strength Calculation:** Trend strength is computed as: Strengthtrend​=max(0,1−Var(Dt​)Var(Rt​)​). The result is capped between 0 and 1 and returned.

**2. Median Crosses (median\_crosses)**

* **Practical Usefulness:**
  + **Process Control:** In manufacturing, if a quality metric frequently crosses its median, it might indicate process instability requiring investigation, even if the average remains acceptable.
  + **Environmental Monitoring:** Tracking how often a pollutant level crosses its long-term median can highlight periods of increased fluctuation or unusual activity.
* **Calculation:**
  1. The median value of the entire time series is calculated.
  2. Then, for each point Yt​ and its preceding point Yt−1​, if (Yt​>median and Yt−1​<median) or (Yt​<median and Yt−1​>median) a crossing is counted.
  3. The total count of crossings is returned.

**3. Trend Changes (trend\_changes)**

* **Practical Usefulness:**
  + **Economic Analysis:** Identifying when an economic indicator like GDP growth rate or unemployment changes its trend can signal shifts in the economic cycle, informing policy decisions.
  + **Marketing Campaign Analysis:** Detecting trend changes in website traffic or conversion rates after launching a marketing campaign can help assess its impact and identify when its effectiveness starts or wanes.
  + **Calculation:**
  1. **Pelt Algorithm (Pruned Exact Linear Time):**
     + F(t), the minimum cost for segmenting the series up to point t, defined as F(t)=min0≤s<t​(F(s)+c(ys..t−1​)+β), is calculated.
     + A pruning rule, essential to speed up the process, is defined in the following fashion: if at some point s<t, another point u<s exists such that F(u)+c(yu..s−1​)≥F(s)+c(ys..t−1​), then u can be pruned as a potential last changepoint before t when considering future points beyond t.
  2. The algorithm iteratively computes F(t) for t=1,...,N (series length), keeping track of optimal last changepoints.
  3. The value returned is the number of detected changepoints (breakpoints) found by backtracking through the optimal solutions.

**4. Linear Regression Slope (linear\_regression\_slope)**

* **Practical Usefulness:**
  + **Resource Depletion:** While analyzing the production data of an oil well, the linear regression slope can estimate the average rate of decline in production per month, helping to forecast its remaining lifespan.
  + **Agricultural Yields:** Farmers can use the slope from regressing crop yield against year to understand the average annual increase or decrease in productivity due to factors like soil changes or farming practices.
* **Calculation:**
  1. **Fit Linear Regression:** First, the linear regression model yi​=β0​+β1​xi​ is fitted, yielding predicted values y^​i​.
  2. **Ordinary Least Squares (OLS):** Then, the slope β1​ is estimated by minimizing the sum of squared residuals ∑(yi​−(β0​+β1​xi​))2. The value is computed and returned using the following formula: β1​=∑i=1N​(xi​−xˉ)2∑i=1N​(xi​−xˉ)(yi​−yˉ​)​ where xˉ is the mean of X and yˉ​ is the mean of Y.

**5. Linear Regression R² (linear\_regression\_r2)**

* **Practical Usefulness:**
  + **Software Performance:** When analyzing software response time over increasing user load, R² indicates how much of the performance degradation is linearly related to the load. A low R² might suggest other non-linear factors are at play.
  + **Educational Assessment:** If tracking student test scores over a semester, R² for a linear fit can show how much of the score improvement is explained by a steady learning trend versus other influences.
* **Calculation:**

1. **Fit Linear Regression:** First, the linear regression model yi​=β0​+β1​xi​ is fitted, yielding predicted values y^​i​.
2. **Total Sum of Squares (SST):** The total variance in the observed data is computed as Y: SST=∑i=1N​(yi​−yˉ​)2, where yˉ​ is the mean of Y.
3. **Sum of Squared Residuals (SSR) / Sum of Squared Errors (SSE):** The variance not explained by the model is computed as SSR=∑i=1N​(yi​−y^1)^2.
4. **R-squared Calculation:** The R^2 value, calculated as R2=1−SST/SSR,​ is returned.

**6. Forecastability (forecastability)**

* **Practical Usefulness:**
  + **Inventory Management:** For a product with high sales forecastability, a business can maintain lower safety stock. For a product with low forecastability (highly random sales), higher safety stock might be needed to avoid stockouts.
  + **Call Center Staffing:** If call arrival rates have high forecastability, staffing levels can be optimized more precisely. Low forecastability might require more flexible staffing or overstaffing to handle unpredictable peaks.
* **Calculation:**
  1. **Power Spectral Density (PSD):** The PSD of the time series is computed. This is done using the chosen model. The Welch's method (the default model) uses these formula:
     + The series is divided into (potentially overlapping) segments.
     + Each segment is windowed (with a Hann window).
     + The Fast Fourier Transform (FFT) is computed for each windowed segment.
     + The squared magnitude of the FFT gives the periodogram for that segment.
     + The PSD is calculated as the average of these periodograms.
  2. **Normalization:** The PSD is normalized so that it sums to 1, effectively treating it as a probability distribution of power across frequencies. Let this be Pk​ for frequency bin k.
  3. **Shannon Entropy:** The Shannon entropy of this normalized PSD is calculated as H=−∑k​Pk​log2​(Pk​).
  4. **Forecastability:** The entropy value is inverted (1/H) and returned.

**7. Entropy Pairs (entropy\_pairs)**

* **Practical Usefulness:**
  + **Medical Diagnostics:** Analyzing physiological signals (like EEG), specific motif distributions captured by this entropy might differentiate between healthy and pathological states, even if overall signal amplitude or frequency is similar.
  + **Anomaly Detection in Sensor Networks:** A sudden change in the feature value from the data of a sensor reading could indicate a novel system behavior or a sensor malfunction, suggesting a deviation from its typical complex pattern generation.

**Calculation:**

1. **Symbolization (Coarse Graining):** The continuous time series is converted into a discrete symbolic series yt of the same length, which is achieved using a "quantile" based method with an alphabet size of 3. Data points are mapped to symbols {1, 2, 3} based on whether they fall into the lower, middle, or upper third of the data's value distribution.
2. **Count Occurrences of Single Symbols:** For each symbol s in the alphabet {1, 2, 3}, the indices of all its occurrences in the symbolized series yt are found and their counts (sizes\_r1[s-1]) are stored.
3. **Calculate Frequencies of Symbol Pairs (Transitions):** For every possible pair of symbols (i, j) where i is the current symbol and j is the next symbol (both from {1, 2, 3}):
   * + The number of times the sequence "symbol i followed immediately by symbol j" appears in yt is counted.
     + Then, the normalized frequency for this pair is calculated as out2[i-1][j-1] = sizes\_r2[i-1][j-1] / (series\_length - 1).
4. **Calculate the Conditional Entropies:** For each current symbol i (from 1 to 3), the Shannon entropy of the distribution of the *next* symbols is computed, given that the current symbol is i. The entropy Hi​ for a given current symbol i is: Hi​=−∑j=13​pj∣i​⋅log(pj∣i​), where pj∣i​ is out2[i-1][j-1] (the normalized frequency of symbol j following symbol i). The logarithm base is typically 2 or e, and 0log0 is treated as 0.
5. **Calculate the Final Sum:** The final value is computed and returned as the sum of these conditional entropies: hh=∑i=13​Hi​.

**8. Fluctuation (fluctuation)**

* **Practical Usefulness:**
  + **Financial Market Volatility:** In analyzing stock price differences, a high value might indicate a "jumpy" market with frequent large price changes over short intervals, signaling higher risk or specific trading conditions.
  + **Wearable Health Monitoring:** For activity data from a wearable, a higher value could distinguish between smooth, consistent activity and erratic, stop-and-go movements.
* **Calculation:**
  1. **Z-Normalization:** The time series Yt​ is first z-normalized to Zt​=(Yt​−mean(Y))/std(Y).
  2. **Successive Differences:** The absolute differences between consecutive values of the z-normalized series are calculated, following the formula ΔZt​=∣Zt​−Zt−1​∣.
  3. **Thresholding and Counting:** Then the counting of the number of absolute differences ΔZt​ that are greater than a threshold of 0.04 is computed as N.
  4. **Proportion:** The final value, which is returned, is the proportion of such differences, computed as: pnn40=N>0.04/Total number of differences (i.e., series length - 1)​​

**9. Autocorrelation Relevance (ac\_relevance)**

* **Practical Usefulness:**
  + **Speech Processing:** In analyzing a speech signal, the first zero-crossing of the autocovariance can be related to the fundamental frequency (pitch) of voiced segments, helping in speech recognition or speaker identification.
  + **Climate Science:** For temperature data, this feature might indicate the dominant short-term cyclical component (related to diurnal cycles if data is high frequency, for example).
* **Calculation:**

1. **Autocovariance Function (ACVF):** The autocovariance function of the time series Yt​ is computed for various lags k. The autocovariance at a lag k is calculated as ACVF(k)=E[(Yt​−μ)(Yt+k​−μ)].
2. **First Zero-Crossing:** The feature value is computed and returned as the smallest positive lag k>1 for which ACVF(k) crosses the defined threshold 1/e.

**10. Seasonal Strength (seasonal\_strength)**

* **Practical Usefulness:**
  + **Retail Demand Planning:** A high seasonal strength for ice cream sales (peaking in summer) allows a company to confidently plan production and marketing efforts around these predictable peaks and troughs.
  + **Tourism Industry:** Hotels can use the seasonal strength of booking data to optimize pricing, staffing and promotions, anticipating high and low seasons.
* **Calculation:**
  1. **STL Decomposition:** The time series Yt​ is decomposed into trend (Tt​), seasonal (St​), and remainder (Rt​) components, using an STL decomposition.
  2. **Detrended Series:** The detrended series is calculated as Yt′​=Yt​−Tt​=St​+Rt​.
  3. **Variances Calculation:**
     1. The variance of the remainder component is calculated: Var(Rt​).
     2. The variance of the detrended series is calculated: Var(Yt′​).
  4. **Strength Calculation:** The value for seasonal strength is computed as Strengthseasonal​=max(0,1−Var(Yt′​)Var(Rt​)​) If Var(Yt′​) is very close to zero, special handling (as in the Python code) applies: if Var(Rt​) is also near zero, strength is 0, otherwise it's 1. This value is capped between 0 and 1 and returned.

**11. Window Fluctuation (window\_fluctuation)**

* **Practical Usefulness:**
  + **Financial Time Series:** Can help distinguish between periods of persistent trending (Hurst exponent > 0.5) and anti-persistent, mean-reverting behavior (Hurst < 0.5) in asset prices, informing trading strategies.
  + **Geophysical Data Analysis:** In analyzing earthquake data or river flow levels, this feature can characterize the long-range dependence or memory effects, helping to understand the underlying physical processes.
* **Calculation:**
  1. **Define Window Sizes (τ):** Approximately 50 window sizes (τ) are generated, logarithmically spaced from a minimum defined size up to half the series length. Duplicates are removed and if fewer than 12 unique τ values remain, the feature returns 0.
  2. **Cumulative Sum:** The series is then transformed into its cumulative sum, yCS.
  3. **Calculate Fluctuation F(τ) for each Window Size:** For each unique window size τ:
* The cumulative sum series yCS is divided into non-overlapping segments of length τ.
* For each segment:
  + - 1. A linear trend is fitted to the segment data points (using xReg=[1,2,...,τ] as the independent variable) and subtracted to get a detrended segment.
      2. For the "rsrangefit" method: The squared range (max - min)2 of this detrended segment is calculated.
      3. For the "dfa" method: The sum of squares of the values in the detrended segment is calculated.
* The values for all segment are then aggregated, using one of the following formulas:

1. F(τ)=(∑(range of detrended segment)2)/number of segments​.
2. F(τ)=(∑(sum of squares of detrended segment))/(number of segments×τ)​.
   1. **Log-Log Analysis and Breakpoint Detection:**
      * The logarithms of the window sizes (log(τ)) and their corresponding fluctuation values (log(F(τ))) are computed.
      * The algorithm searches for an optimal breakpoint in the log(F(τ)) vs log(τ) plot. This is done by iterating through possible split points. For each split point i:
        1. Fit a linear regression to the first i points (log(τ)[0..i−1] vs log(F)[0..i−1]).
        2. Fit another linear regression to the remaining points (log(τ)[i−1..end] vs log(F)[i−1..end]).
        3. Calculate the sum of squared errors (residuals) for both fits. The total sum of squared errors for this split point i is stored.
      * The split point i that yields the minimum total sum of squared errors is chosen as the optimal breakpoint. A minimum number of points is required for each regression.
   2. **Result:** The final computed and returned value is the proportion of window sizes that fall into the first linear segment (before the optimal breakpoint): (firstMinInd+1)/ntt, where ntt is the number of unique τ values.

**12. Short-Term Variation (st\_variation)**

* **Practical Usefulness:**
  + **Quality Control:** In a manufacturing process, if a product dimension shows low short-term variation, it suggests stability. An increase in reversals might indicate an emerging issue.
  + **Algorithmic Trading:** A high number of local upward movements might suggest short-term momentum that a trading algorithm could try to exploit.
* **Calculation:**
  1. Iterate through the time series Yt​ from the second point to the penultimate point (t=1 to N−1).
  2. For each point Yt​, a comparison with its previous point Yt−1​ is computed.
  3. The number of times Yt​>Yt−1​ is counted and returned.

**13. Autocorrelation (ac)**

* **Practical Usefulness:**
  + **Demand Forecasting:** If daily sales data has a high positive lag-1 autocorrelation, it means high sales one day are likely followed by high sales the next, useful for short-term inventory adjustments.
  + **Energy Load Prediction:** Electricity load often shows strong positive lag-1 autocorrelation, as load at one hour is very similar to the previous hour, crucial for grid balancing.
* **Calculation:**

1. Given a time series Yt​ of length N with mean Yˉ, the autocovariance at lag 1 (c1​) is calculated as c1​=N1​∑t=1N−1​(Yt​−Yˉ)(Yt+1​−Yˉ) and the variance (autocovariance at lag 0, c0​) is calculated as: c0​=N1​∑t=1N​(Yt​−Yˉ)2
2. The autocorrelation value that is computed and returned is defined as: ρ1​=c0​c1​​.

**14. Differenced Series Autocorrelation (diff\_series)**

* **Practical Usefulness:**
  + **Financial Returns Analysis:** Stock prices are often non-stationary (have a trend/random walk). Analyzing the autocorrelation of their differences (returns) helps identify if there's any remaining predictability after removing the primary random walk component.
  + **Process Improvement:** If a process output shows a trend, differencing can make it stationary. This feature can then reveal if there are lingering systematic patterns in the rate of change that could be addressed.
* **Calculation:**
  1. **First Differencing:** A new time series DYt​ is created by taking the first differences of the original series, Yt​: DYt​=Yt+1​−Yt​ for t=1,...,N−1.
  2. **Autocorrelation of Differenced Series:** Then the first 10 autocorrelation coefficients (ρ1​,ρ2​,...,ρ10​) of the differenced series Dyt are calculated​, using the same method as for the ac feature (lag-1 ACF), but applied to DYt​ and for lags 1 through 10.
  3. **Sum of Squares:** The returned value is calculated as the sum of the squares of these first 10 autocorrelation coefficients: diff1a​cf10=∑k=110​(ρk​(DYt​))2

**15. Series Complexity (complexity)**

* **Practical Usefulness:**
  + **Signal Processing:** When comparing different sensor readings that measure the same phenomenon, a much higher complexity in one signal might indicate noise or interference, rather than true signal variation.
  + **Machine Condition Monitoring:** An increase in the complexity of vibration data from a machine over time could indicate developing faults or wear and tear, as the vibrations become less regular.
* **Calculation:**
  1. **Z-Normalization:** The mean (μ) and standard deviation (σ) of the series is computed. Then, the series is z-normalized following the formula Zt​=(Yt​−μ)/σ ,for each point t.
  2. **Raw Complexity Estimate:** If the length of Z is less than 2, the returned value for complexity is 0. Otherwise, the complexity of the series is computed in the following fashion:
     + The first differences of the normalized series are computed: ΔZt​=Zt+1​−Zt​.
     + The complexity estimate is computed and returned as the square root of the sum of the squares of these differences: CE=∑t=1N−1​(ΔZt​)2​

**16. Records Concentration (rec\_concentration)**

* **Practical Usefulness:**
  + **Customer Segmentation:** While analyzing customer purchase frequency data, this feature might reveal common purchasing patterns (most customers buy 2-3 times a month, for example), helping to segment customers.
  + **Sensor Data Validation:** If a sensor typically outputs values concentrated in a specific range, a shift in this feature could indicate a calibration issue or a real change in the measured environment.
* **Calculation:**
  1. **Data Range:** The minimum and maximum values of the time series are determined.
  2. **Binning:** The calculated range is divided into 10 equal-width bins.
  3. **Histogram:** The number of data points from the time series that fall into each of the 10 bins are counted.
  4. **Mode Identification:** The bin with the highest count (the modal bin) is identified.
  5. **Feature Value:** The returned value is the midpoint (center value) of this modal bin.

**17. Series Centroid (centroid)**

* **Practical Usefulness:**
  + **Audio Analysis:** In music information retrieval, the spectral centroid distinguishes between "bright" sounds (high centroid, like cymbals) and "dull" sounds (low centroid, like a bass drum), aiding in instrument recognition or genre classification.
  + **Vibration Analysis:** For machine diagnostics, a shift in the spectral centroid of vibration signals can indicate changes in operational speed or the emergence of specific fault frequencies (bearing defects often have characteristic high-frequency components, for example).
* **Calculation:**
  1. **Power Spectral Density (PSD):** The PSD of the time series, P(f), is computed, which shows the power of the signal at each frequency f.
  2. **Centroid Calculation:** The spectral centroid C, which is returned, is computed as the power-weighted average of the frequencies: C=∑k​Pk​∑k​fk​⋅Pk​​ The sum is over all frequency bins k.