**1. Visualization Functions**

These functions help plot and analyze time series visually:

**plot\_ts**

* **Purpose:** Visualize a time series with optional moving average (MA) and confidence intervals.
* **Inputs:**
  + ts: The time series.
  + plot\_ma: Whether to plot the moving average.
  + plot\_intervals: Whether to include confidence intervals.
  + window: The rolling window size for MA.
* **Output:** Plots the time series, optionally with MA and intervals.

**test\_stationarity\_acf\_pacf**

* **Purpose:** Test stationarity using the Dickey-Fuller test and visualize autocorrelation (ACF) and partial autocorrelation (PACF).
* **Inputs:**
  + ts: The time series.
  + sample: Proportion of initial observations for sample mean and variance.
  + maxlag: Max lag for ACF/PACF.
* **Output:** A plot and stationarity conclusion (stationary/non-stationary).

**2. Trend and Stationarity**

Functions that help analyze and handle trends:

**fit\_trend**

* **Purpose:** Fit a polynomial trend line to the data.
* **Inputs:**
  + ts: The time series.
  + degree: Degree of the polynomial to fit.
* **Outputs:**
  + Dataframe with trend values.
  + Trend parameters.

**diff\_ts / undo\_diff**

* **Purpose:** Compute or reverse differencing for a time series.
* **Inputs:**
  + ts: Time series.
  + lag: Difference lag.
* **Output:** Differenced series (or original reconstructed series).

**3. Decomposition and Seasonality**

**decompose\_ts**

* **Purpose:** Decompose a time series into trend, seasonality, and residual components.
* **Inputs:**
  + ts: Time series.
  + s: Seasonal period (e.g., 7 for weekly data).
* **Output:** Dictionary of decomposed components (trend, seasonality, residuals).

**extract\_seasonality / apply\_seasonality**

* **Purpose:** Extract seasonality patterns from the data or apply them to adjust the series.
* **Inputs:**
  + freq: Frequency of observations (D, W, etc.).
* **Output:** Adjusted time series or seasonal components.

**4. Outliers**

**find\_outliers / remove\_outliers**

* **Purpose:** Detect and remove outliers using a one-class SVM.
* **Inputs:**
  + ts: Time series.
  + perc: Proportion of points treated as outliers.
* **Outputs:**
  + Dataframe marking outliers or cleaned series.

**5. Train-Test Splitting**

**split\_train\_test**

* **Purpose:** Split time series into training and testing sets.
* **Inputs:**
  + test: Test size or split point.
  + plot: Whether to plot the split.
* **Output:** Training and testing subsets.

**6. Statistical Tests**

**test\_2ts\_casuality**

* **Purpose:** Check Granger causality between two time series.
* **Inputs:**
  + ts1, ts2: Time series to test.
  + maxlag: Max lag to test.
* **Output:** Prints causality results.

**7. Forecasting Models**

**Random Walk:**

* simulate\_rw, forecast\_rw: Simulate and forecast using a random walk model.

**Exponential Smoothing:**

* tune\_expsmooth\_model, fit\_expsmooth: Tune and fit models like Holt-Winters for exponential smoothing.

**ARIMA:**

* tune\_arima\_model, fit\_sarimax: Grid search for optimal ARIMA parameters and fit ARIMA/SARIMAX models.

**GARCH:**

* fit\_garch: Fit GARCH models for volatility prediction.

**Neural Networks:**

* fit\_lstm, forecast\_lstm: Preprocess, train, and forecast using LSTM models.

**Prophet:**

* fit\_prophet, forecast\_prophet: Use the Facebook Prophet library for time series forecasting.

**8. Utilities**

**utils\_conf\_int**

* **Purpose:** Compute confidence intervals for forecasts.
* **Inputs:**
  + lst\_values: Forecast values.
  + error\_std: Standard deviation of residuals.
* **Output:** Confidence intervals.

**utils\_generate\_indexdate**

* **Purpose:** Generate a date index for forecasts.
* **Inputs:**
  + start, end: Start and end dates or number of periods.
* **Output:** Pandas date index.