X13-ARIMA-SEATS

Rohit Jangid

July 2024

Contents

1 Introduction

The Ministry of Statistics and Programme Implementation (MoSPI) is responsible for assembling and sharing key country indexes such as the Gross Domestic Product (GDP) and the Consumer Price Index (CPI). GDP represents the overall economic activity, while CPI measures inflation trends. Both indicators provide critical insights into a country's economic performance. Seasonal variations, due to factors like holidays and agriculture, often obscure the true trends. This necessitates **seasonal adjustment**, a statistical technique to remove these effects and reveal the underlying patterns in time series data.

Currently, MoSPI does not apply seasonal adjustments to its indices. This project's goal is to address the need for seasonal adjustment techniques using the widely adopted X-13-ARIMA-SEATS model, developed by the U.S. Census Bureau. Our objective is to provide clarity on the technical details of this model and equip MoSPI with the tools for correct implementation.

2 Problem Summary

Economic indicators such as GDP and CPI are often influenced by seasonal factors like holidays, agricultural cycles, and other periodic events. These factors introduce systematic patterns in the data, which, if not accounted for, can mislead decision-makers about the actual trends in the economy. For instance, a rise in consumer spending around holidays may inflate the perception of economic growth, which is temporary.

The **X-13-ARIMA-SEATS** model is a powerful statistical tool designed to remove such seasonal effects. It is based on ARIMA (AutoRegressive Integrated Moving Average) modeling combined with signal extraction techniques. This model enables users to separate the trend, seasonal, and irregular components from time series data.

Our project aims to:

- Provide a detailed technical explanation of the X-13-ARIMA-SEATS model for MoSPI.
- Test the appropriateness of the model on Indian economic data.
- Deliver an R package to apply the model on Indian data for seasonal adjustment.

We have made significant progress in understanding the ARIMA framework and its integration with signal extraction for seasonal adjustments. We are currently testing the model on sample datasets and preparing the technical document for submission to MoSPI (See [**?**], page 233).

3 X-13ARIMA-SEATS

The X-13ARIMA-SEATS is a comprehensive seasonal adjustment software developed by the U.S. Census Bureau. It is widely used for time series analysis to remove seasonal variations from data and identify underlying trends. The software integrates two powerful algorithms: the X-11 method and SEATS (Signal Extraction in ARIMA Time Series). These techniques allow users to adjust for seasonal effects and extract trend and irregular components.

3.1 SEATS and X-13

SEATS is a model-based approach that uses ARIMA modeling to decompose a time series into its trend, seasonal, and irregular components. It was originally developed by Agustin Maravall and Victor Gómez at the Bank of Spain. SEATS provides a statistically rigorous way to account for seasonal patterns and produce smoother trends (See X-13ARIMA-SEATS Manual, page 2).

X-13, on the other hand, builds on the earlier X-11 and X-12 methods, which use filter-based techniques for seasonal adjustment. The X-13ARIMA-SEATS software combines the strengths of both approaches, giving users flexibility in choosing between model-based and filter-based seasonal adjustments (See X-13ARIMA-SEATS Manual, page 2).

3.2 regARIMA and TRAMO

The regARIMA model used in X-13ARIMA-SEATS is based on TRAMO (Time Series Regression with ARIMA Noise, Missing Observations, and Outliers). TRAMO handles preprocessing tasks such as outlier detection, modeling calendar effects, and managing missing observations. It fits a regression model with ARIMA errors to clean the data before applying seasonal adjustment. The software is able to extend time series data through forecasting and backcasting, improving the accuracy of seasonal adjustments near the boundaries of the data (See X-13ARIMA-SEATS Manual, page 5).

The X-13ARIMA-SEATS program operates in several stages to transform raw time series data into its seasonal, trend, and irregular components. The main steps involve reading the raw data, applying the regARIMA model for transformations, handling outliers and missing values, and then using either SEATS or X-11 to extract the seasonal and other components.

3.2.1 Workflow Explanation

The general workflow of the X-13ARIMA-SEATS program can be broken down as follows:

- **Raw Data Input**: The process begins with inputting the raw time series data, which is defined in the 'series' spec of the specification file.
- **regARIMA Model**: The regARIMA model is applied to the data. This involves:
  - **Transformations**: Automatically or manually defined transformations (e.g., logarithmic or differencing).
  - **Handling Outliers and Missing Values**: Identifies and adjusts outliers, and fills missing data where applicable.
  - **Model Fitting**: Fits an ARIMA model to the transformed data, accounting for these adjustments.

  The output from the regARIMA model is the linearized series, which is free of seasonal effects, outliers, and missing values.

- **Seasonal Adjustment**: This linearized series is passed through either the SEATS or X-11 algorithm to extract the seasonal, trend, and irregular components.
  - **SEATS**: A model-based approach for decomposing the series.
  - **X-11**: A filter-based approach for seasonal decomposition.
- **Final Output**: The program provides the seasonal component, trend component, and irregular component of the time series.

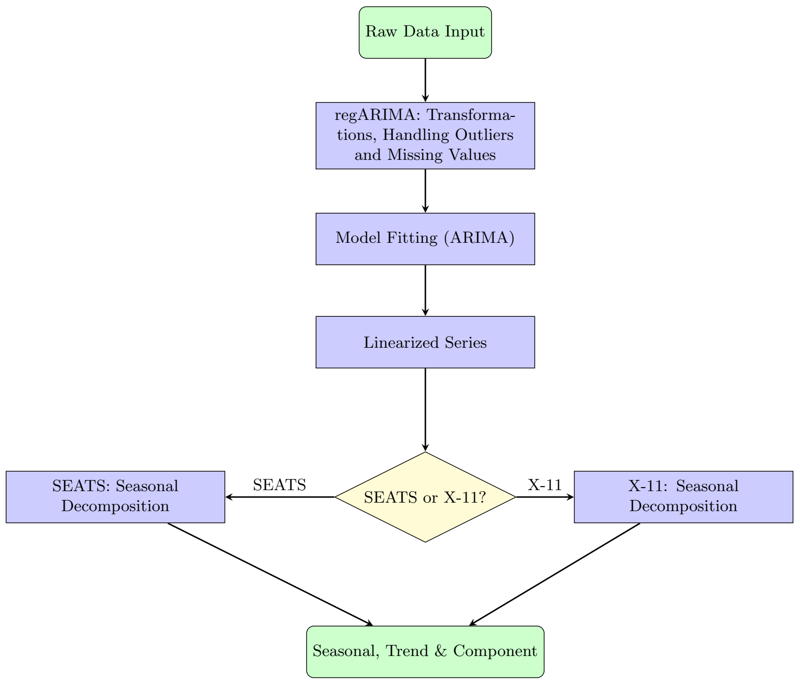

Figure 1: Flowchart of the Seasonal Adjustment Process

3.3 Using X-13ARIMA-SEATS

The software can be executed using the command line by specifying the path to the input specification (spec) file. A generic command to run X-13ARIMA-SEATS is:

```
path\x13as path\filename
```

The spec file contains the necessary instructions for running the software, including details about the time series data, ARIMA models, and output preferences. It is a simple text file with a '.spc' extension.

3.3.1 Spec File and Key Specifications

A spec file consists of various specifications (specs) that control the flow of execution. Here is an example spec file for demonstration purpose.

```
series{
  title = "Consumer Food Price Index - All India Combined"
  start = 2013.01
  span = (2013.01, 2024.08)
  data = (
    105.4 106.4 106.5 107.5 109.1 112.4 115.2 117.3 119.0 121.1 123.9 118.7
    115.6 114.8 115.7 117.4 118.8 120.5 125.4 127.5 126.4 125.8 125.3 123.4
    122.7 122.7 122.8 123.4 124.5 127.1 128.1 130.3 131.3 132.4 132.9 131.3
    131.1 129.2 129.2 131.3 133.8 137.0 138.8 138.0 136.5 136.8 135.6 133.1
    131.9 131.8 131.8 132.1 132.4 134.1 138.3 140.1 138.2 139.4 141.5 139.7
    138.1 136.1 135.5 135.8 136.5 138.0 140.1 140.5 138.9 138.2 137.8 136.0
    135.0 135.1 135.9 137.3 139.0 141.1 143.4 144.7 146.0 149.1 151.6 155.3
    153.4 149.7 147.8 153.4 151.8 153.4 156.7 157.8 161.6 165.5 166.0 160.6
    156.4 155.5 155.0 156.4 159.4 161.3 162.9 162.7 162.7 166.9 169.1 167.1
    164.9 164.6 166.9 169.4 172.1 173.8 173.8 175.1 176.7 178.6 177.0 174.1
    174.8 174.4 174.9 175.9 177.2 181.7 193.8 192.5 188.4 190.4 192.4 190.7
    189.3 189.5 189.8 191.2 192.6 198.7 204.3 203.4
  )
}

transform{
  function = auto
}

automdl{maxorder = (3, ) }

outlier{types = (ls ao)}

estimate{
    save = residuals
}

regression {
  variables = (const, td)
  user = (diwali)
  start = 2013.01
  data = (
   0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0405 -0.0405  0.0
   0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.3405 -0.3405  0.0
   0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0 -0.6595  0.6595  0.0
   0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.3405 -0.3405  0.0
   0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.3405 -0.3405  0.0
```

```
   0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0 -0.3595  0.3595  0.0
   0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.3405 -0.3405  0.0
   0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0 -0.6595  0.6595  0.0
   0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0 -0.0595  0.0595  0.0
   0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.3405 -0.3405  0.0
   0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0 -0.6595  0.6595  0.0
   0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.3405 -0.3405  0.0
   0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.3405 -0.3405  0.0)
print = rmx
}

seats{}
```

3.3.2 Commonly used specs in `X13-ARIMA-SEATS`

1. **SERIES** The SERIES spec defines the input time series data for analysis. It includes details such as the starting and ending dates, frequency, and data type. This spec is crucial as it tells the software what time series to use for seasonal adjustment and other analyses. The SERIES spec also allows users to set the precision of the data and handle missing values appropriately (See X-13ARIMA-SEATS Manual, page 181).

2. **AUTOMODEL** The AUTOMODEL spec automates the selection of the best-fitting ARIMA model by comparing several candidate models and selecting the one with the lowest AIC (Akaike Information Criterion). This is particularly useful when the user is unsure of the correct model structure. The procedure is closely based on TRAMO, a method developed by Gómez and Maravall (2000) (See X-13ARIMA-SEATS Manual, page 50).

   The method is detailed in Gómez and Maravall's paper, "Automatic Modeling Methods for Univariate Series" in *A Course in Time Series*, edited by D. Pena, G. C. Tiao, and R. S. Tsay, New York: J. Wiley and Sons, 2000.

3. **Estimate** This spec controls the estimation method used for fitting the ARIMA model. Maximum likelihood estimation (MLE) is typically used, but users can modify various estimation settings such as the number of iterations and convergence criteria (See X-13ARIMA-SEATS Manual, page 100).

4. **Outlier** The Outlier spec detects and adjusts for outliers in the data. Three types of outliers are commonly detected: Additive Outliers (AO), Level Shifts (LS), and Transitory Changes (TC). The software automatically identifies these outliers and includes them as regressors in the model (See X-13ARIMA-SEATS Manual, page 133).

5. **Regression** The Regression spec allows users to specify the regression variables used in the regARIMA model. These variables can include predefined effects, such as trading day and holiday adjustments, or user-defined regressors (See X-13ARIMA-SEATS Manual, page 144).

6. **Transform** The Transform spec handles transformations of the data, such as logarithmic transformations, to stabilize the variance or make the series more stationary. This is particularly useful when dealing with time series that exhibit non-constant variance (See X-13ARIMA-SEATS Manual, page 212).

7. SEATS The SEATS spec controls the use of the SEATS algorithm for seasonal adjustment. This includes options for model selection, output settings, and diagnostics for the decomposition of the time series into its components (See X-13ARIMA-SEATS Manual, page 169).

8. **X-11** The X-11 spec is used when applying the X-11 seasonal adjustment method. This spec provides options for controlling seasonal filters, diagnostics, and the handling of trading day effects (See X-13ARIMA-SEATS Manual, page 223).

4 Background Mathematical Details

Definition 4.1. *A time series* $\{X_t\}$ *is an ordered collection (indexed by time) of random variables.*

Definition 4.2. *A time series is called* ***stationary*** *if the joint distribution of any finite collection of data points is time-independent. Formally, for all positive integers* $h, n_1, n_2, \ldots, n_k$*,*

$$X_{n_1}, X_{n_2}, \ldots, X_{n_k} \stackrel{d}{=} X_{n_1+h}, X_{n_2+h}, \ldots, X_{n_k+h}.$$

Definition 4.3. *A time series is called* ***covariance-stationary*** *if the following conditions are satisfied for all positive integers* $h, n, n_1,$ *and* n_2*:*

$$\mathbb{E}[X_n] = \mu < \infty$$

$$Cov(X_{n_1}, X_{n_2}) = Cov(X_{n_1+h}, X_{n_2+h}) < \infty$$

All stationary time series are covariance-stationary, but the converse is not generally true.

Definition 4.4. *A time series is called a* ***Gaussian process*** *if the joint distribution of any finite collection of data points follows a multivariate normal distribution.*

Theorem 4.1. *A covariance-stationary Gaussian process is stationary.*

Definition 4.5. *For a covariance-stationary time series, the* ***autocovariance function*** *is defined as:*

$$\rho(h) = Cov(X_n, X_{n+h}),$$

where h *is called the* ***lag****.*

Definition 4.6. *White noise* ε_t *is a sequence of uncorrelated random variables with zero mean and constant variance* σ^2*.*

4.1 Time Series Models

This section provides the mathematical foundations for time series modeling, starting with the basic models like AR(p) and MA(q), and progressing to more complex models such as ARMA(p, q), ARIMA(p, d, q), and ARIMA(p, d, q)(P, D, Q)[s].

4.1.1 AR(p) Process

Definition 4.7. *A time series $\{X_t\}$ is said to follow an **AR(p)** process if it satisfies the equation:*

$$X_t = \sum_{i=1}^{p} \phi_i X_{t-i} + \varepsilon_t$$

*where $\phi_1, \ldots, \phi_p$ are the **parameters** of the model, and ε_t is **white noise** with mean zero and variance σ^2.*

Here, p is called the order of the AR process.

The $AR(p)$ process can be equivalently written using the backshift operator B as:

$$X_t = \sum_{i=1}^{p} \phi_i B^i X_t + \varepsilon_t$$

where the backshift operator B is defined by:

$$B^i X_t = X_{t-i}$$

In an $AR(p)$ process the current value of the series is a linear sum of the previous values and an independent error term.

4.1.2 MA(q) Process

Definition 4.8. *A time series $\{X_t\}$ is said to follow an **MA(q)** process if it satisfies the equation:*

$$X_t = \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} + \varepsilon_t$$

*where $\theta_1, \ldots, \theta_q$ are the **coefficients** of the model, and ε_t's are white noise error terms.*

Here, q is called the order of the MA process.

The MA(q) process can be equivalently written in terms of the backshift operator B as:

$$X_t = (1 + \sum_{i=1}^{q} \theta_i B^i)\varepsilon_t.$$

An $MA(q)$ process can be thought of as a weighted moving sum of a white noise series.

4.1.3 ARMA(p, q) Process

Definition 4.9. *A time series $\{X_t\}$ is said to follow an **ARMA(p, q)** process if it satisfies the equation:*

$$X_t = \sum_{i=1}^{p} \phi_i X_{t-i} + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} + \varepsilon_t$$

where $\phi_1, \ldots, \phi_p$ and $\theta_1, \ldots, \theta_q$ are the coefficients of the model, and ε_t are white noise error terms.

Here, p, q are called the order of the ARMA process.

The ARMA(p, q) process can be equivalently written in terms of the backshift operator B as:

$$\left(1 - \sum_{i=1}^{p} \phi_i B^i\right) X_t = (1 + \sum_{i=1}^{q} \theta_i B^i)\varepsilon_t$$

or equivalently as:

$$\phi(B)X_t = \theta(B)\varepsilon_t$$

ARMA(p, q) process is a combination of the AR an MA models which means the current value is linear sum of the previous values and a weighted sum of some white noise process.

Theorem 4.2. *All MA(q) processes of finite order q are covariance stationary and* $\rho(h) = 0$ *for* $|h| > q$.

Theorem 4.3. *An AR(p) process* $\{X_t\}$ *defined by* $\Phi(B)X_t = \varepsilon_t$ *is covariance stationary if the roots of the polynomial* $\Phi(z) = 0$ *lie outside the unit circle.*

Theorem 4.4. *An ARMA(p, q) process* $\{X_t\}$ *defined by* $\Phi(B)X_t = \Theta(B)\varepsilon_t$ *is covariance stationary if the roots of the polynomial* $\Phi(z) = 0$ *lie outside the unit circle.*

4.1.4 ARIMA(p, d, q) Process

We define the differencing operator ∇ as $\nabla X_t = X_t - X_{t-1}$ or $\nabla = 1 - B$.

Definition 4.10. *A time series* X_t *is said to be an* ***ARIMA(p, d, q)*** *process if* $Y_t = \nabla^d X_t$ *is a stationary ARMA(p, q) process.*

4.1.5 ARIMA(p, d, q)(P, D, Q)[s] Process

The general ARIMA model can be extended to include seasonal factors. The seasonal AR and MA components are represented by polynomials with seasonal lags.

The general ARIMA(p, d, q)(P, D, Q)$_s$ model is given by:

$$\phi(B)\Phi(B^s)(1 - B)^d(1 - B^s)^D z_t = \theta(B)\Theta(B^s)a_t$$

where:

- $\phi(B)$ and $\theta(B)$ are the non-seasonal autoregressive (AR) and moving average (MA) polynomials.
- $\Phi(B^s)$ and $\Theta(B^s)$ are the seasonal AR and MA polynomials with seasonal period s.
- $(1 - B)^d$ represents the non-seasonal differencing operator.
- $(1 - B^s)^D$ represents the seasonal differencing operator.
- a_t is white noise with mean zero and variance σ^2.

4.2 Time Series Decomposition

A time series can often be broken down into three components:

1. **Trend Component** (m_t): The long-term movement or direction in the data.
2. **Seasonal Component** (s_t): The repeating, periodic fluctuation (e.g., annual cycles).
3. **Random Component** (e_t): The stochastic, unpredictable part of the series.

The decomposition can be done additively or multiplicatively:

- **Additive Decomposition**: $X_t = m_t + s_t + e_t$
- **Multiplicative Decomposition**: $X_t = m_t \cdot s_t \cdot e_t$

Seasonal adjustment is the process of removing the seasonal component from a time series to analyze the underlying trend and cycle independently. It is essential for analyzing economic data that exhibits seasonal patterns, such as consumer spending, which often increases before holidays like Christmas.

In many practical applications, the observed time series can be influenced by both deterministic effects (such as calendar effects, trends, or interventions) and stochastic components (captured by ARIMA processes). The **Regression with ARIMA Noise** (RegARIMA) model combines a linear regression model with ARIMA errors. This approach allows us to model both the deterministic structure of the time series and the stochastic dependencies (e.g., autocorrelation) present in the error terms.

The general form of a regression model with ARIMA errors is given by:

$$y_t = X_t\beta + z_t$$

Where:

- y_t is the observed time series at time t.
- X_t is a vecor of regression variables (e.g., calendar effects, external regressors).
- β is the vector of coefficients corresponding to the regression variables.
- z_t is the error term, which follows an ARIMA process.

The residuals z_t are assumed to follow an ARIMA(p, d, q) process, which captures the autocorrelation and the non-stationarity of the data. The ARIMA process for z_t is expressed as:

$$\phi(B)\Phi(B^s)(1-B)^d(1-B^s)^D z_t = \theta(B)\Theta(B^s)a_t$$

Combining the regression component and the ARIMA noise model, we get the complete RegARIMA model:

$$\phi(B)\Phi(B^s)(1-B)^d(1-B^s)^D(y_t - X_t\beta) = \theta(B)\Theta(B^s)a_t$$

This model provides a comprehensive approach to handle both deterministic and stochastic components in the time series. The regression variables X_t account for known, external influences, while the ARIMA model for z_t captures the autocorrelation and non-stationarity in the data.

4.3 Ljung-Box Q Test for Model Residuals

The Ljung-Box Q test is a statistical diagnostic tool used to assess whether the residuals of a time series model exhibit autocorrelation. This test helps in validating the adequacy of models such as ARIMA in time series analysis. The main objective of the Ljung-Box test is to determine if residuals (errors) from a time series model are independently distributed (i.e., they exhibit no serial autocorrelation up to a certain lag). If autocorrelation is detected, this indicates possible model inadequacy or misspecification, suggesting that adjustments or alternative models may be needed for accurate forecasting. Typically, a sufficient sample size is required for reliable conclusions. The hypotheses for the Ljung-Box test are as follows:

- **Null Hypothesis (H_0)**: There is no autocorrelation in residuals up to the specified lag (the residuals are independently distributed).
- **Alternative Hypothesis (H_a)**: There is autocorrelation in residuals, suggesting that the model may not adequately capture the structure of the data.

Rejecting H_0 suggests that the model may be misspecified or incomplete. The Ljung-Box test statistic Q is calculated as:

$$Q = n(n+2)\sum_{k=1}^{m}\frac{\hat{r}_k^2}{n-k}$$

where:

- n is the number of observations,
- $\hat{r}_k$ is the sample autocorrelation at lag k,
- m is the number of lags being tested.

The test statistic Q follows an approximate chi-square (χ^2) distribution with h degrees of freedom, where h is the number of parameters estimated in the model (such as p and q in ARIMA models). At a given significance level α, the critical value is derived from the chi-square distribution. If Q exceeds the critical value, the null hypothesis H_0 is rejected, indicating significant autocorrelation in the residuals and suggesting that the model may be inadequate.

4.4 Hannan-Rissanen Estimation

5 RegARIMA

The **RegARIMA** model in the X-13ARIMA-SEATS software is an advanced tool for handling regression effects, detecting outliers, and fitting ARIMA models to time series data. The process includes several key steps, each involving tests and decisions based on the time series characteristics.

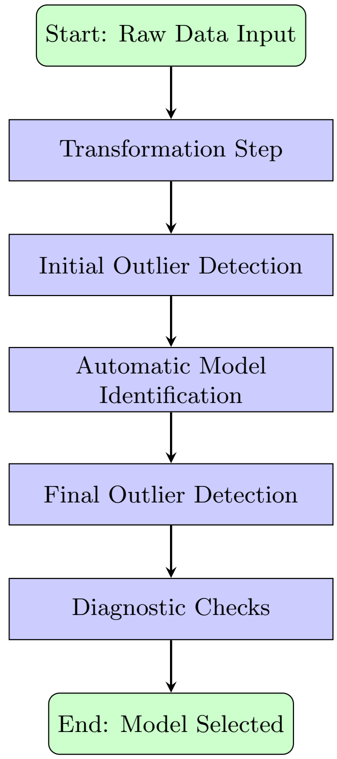

Figure 2: Flowchart Illustrating the Steps Taken During the RegARIMA Process

5.1 Transformation Step

When performing **RegARIMA**, the program can apply several types of transformations to the data. These transformations stabilize variance and ensure that the time series meets the assumptions required for ARIMA modeling. The **TRANSFORM** spec allows the user to specify the type of transformation or to let the program automatically select the best transformation.

5.1.1 Power or Function Transform

The user can specify a power transformation, which applies the Box-Cox transformation to the data. Let Y_t be the original series and y_t be the transformed series. The Box-Cox transformation is defined as:

$$y_t = \begin{cases} \log(Y_t) & \text{if } \lambda = 0 \\ \lambda^2 + \frac{Y_t^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0 \end{cases}$$

By default, $\lambda = 1$, meaning no transformation is applied. This is specified using the `power` option in the `TRANSFORM` spec. Alternatively, the user can specify a function transformation. The `function = auto` option applies a test to check whether a log transformation or no transformation is more appropriate. The following functions can be specified:

value	transformation	range for Y_t	equivalent power argument
none	Y_t	all values	power = 1
log	$\log(Y_t)$	$Y_t > 0$ for all t	power = 0
sqrt	$0.25 + 2\left(\sqrt{Y_t} - 1\right)$	$Y_t \geq 0$ for all t	power = 0.5
inverse	$2 - \left(\frac{1}{Y_t}\right)$	$Y_t \neq 0$ for all t	power = -1
logistic	$\log\left(\frac{Y_t}{1-Y_t}\right)$	$0 < Y_t < 1$ for all t	no equivalent

Table 1: Transformations Available Using the function Argument for Transform

Test for Checking Log Transform vs No Transform
The program tests whether a log transformation is appropriate using the following procedure:

1. A default ARIMA $(0,1,1)(0,1,1)$ model is fit on both the untransformed and log-transformed series. All user-specified regressors are included in both models. If the user has specified a model in the ARIMA spec then that model is used instead of the default model.

2. The AICC value is calculated for both fitted models.

3. If $\text{AICC}_{\text{nolog}} - \text{AICC}_{\text{log}} \leq \text{AICC}_{\text{diff}}$, then no transformation is favored. The default for $\text{AICC}_{\text{diff}}$ is -2, meaning that negative values of $\text{AICC}_{\text{diff}}$ favor the log transformation.

Calculation of AICC
The AICC is a modified version of the Akaike Information Criterion (AIC) that adjusts for small sample sizes. Let $Y_1, Y_2, \ldots, Y_n$ be the original data. The AICC is calculated as:

$$\text{AICC} = -2L_N + 2n_p\left(\frac{1+n_p}{N-n_p-1}\right)$$

Where:

- L_N is the log-likelihood of the model.
- n_p is the number of parameters.
- $N = n - d - sD$ is the effective number of observations, where d is the non-seasonal differencing order, D is the seasonal differencing order, and s is the seasonal period (12 for monthly or 4 for quarterly data).

Log-Likelihood Adjustment After Transformation
Assuming $Y = (Y_1, Y_2, \ldots, Y_N)'$ is the data after differencing and X is the regression matrix we have:

$$\text{AICC}_N = -2L_N + 2n_p\left(\frac{1+n_p}{N}\right)$$

$$L_N = L_N(\phi, \theta, \Phi, \Theta, \sigma^2, \beta | Y, X)$$

is the log likelihood of the data from the regARMA model:

$$L_N(\phi, \theta, \Phi, \Theta, \sigma^2|Y, X) = -\frac{1}{2}\log(2\pi|\Omega|) - \frac{1}{2}(Y - X\beta)'\Omega^{-1}(Y - X\beta)$$

$\Omega(\phi, \theta, \Phi, \Theta, \sigma^2, \beta)$ is the covariance matrix of the data Y following the $ARIMA(p, d, q)(P, D, Q)_s$ process with Gaussian errors.

If a log transformation is applied, the log-likelihood must be adjusted to account for the transformation. The log-likelihood after transformation is calculated as:

$$L_N^* = \tilde{L}_N + \sum \log \left| \frac{df(Y_t)}{dY_t} \right|$$

Where:

- $\tilde{L}_N$ is the log-likelihood for the transformed data.
- The summation term adjusts for the Jacobian of the transformation.

This adjustment ensures that the log-likelihoods of the transformed and untransformed models are comparable, which is important for calculating information criteria like AICC and BIC.

Proof for Log-Likelihood Adjustment

Let $X \sim F(x|\theta)$ be a random variable, and let $Y = f(X)$. If $f_X(x|\theta)$ is the probability density function (pdf) of X, then the pdf of Y is given by:

$$f_Y(y) = f_X(f^{-1}(y)|\theta) \cdot \left| \frac{dx}{dy} \right|$$

Taking the logarithm:

$$\log f_Y(y|\theta) = \log f_X(x|\theta) + \log \left| \frac{dx}{dy} \right|$$

$$L(\theta|X) = L(\theta|Y) + \log \left| \frac{df(X)}{dX} \right|$$

For i.i.d observations, the log-likelihood is adjusted as follows:

$$L(\theta|X = (X_1, X_2 \ldots X_N)') = L(\theta|Y = (Y_1, Y_2 \ldots Y_N)') + \sum_{i=1}^{n} \log \left| \frac{df(X_i)}{dX_i} \right|$$

5.1.2 Length of Month (LoM) and Length of Quarter (LoQ) Adjustments

For monthly or quarterly series, users can specify adjustments for the length of the month (LoM) or the length of the quarter (LoQ). Many time series, such as GDP, are cumulative sums over time, so longer months or quarters may have larger values. These adjustments correct for this deterministic effect.

The LoM adjustment factor is calculated as:

$$\text{LoM}_t = \frac{m_t - \bar{m}}{\bar{m}}$$

Where:

- m_t is the number of days in month t.
- $\bar{m} = 30.4375$ is the average length of a month, accounting for leap years.

The Leap year correction factors for each time point (for monthly data) are calculated as follows:
Note that 28.25 is the average length of February.

$$LeapYear_t = \begin{cases} \frac{28}{28.25} & \text{for 28-day months} \\ \frac{29}{28.25} & \text{for 29-day months (leap year February)} \\ 1 & \text{for all other months} \end{cases}$$

5.1.3 User-Specified Adjustments

Users can also provide custom adjustment factors for each time point using the `mode` option in the `TRANSFORM` spec. If `mode` is set to `ratio`, the adjustment factors $C_1, C_2, \ldots, C_N$ are provided by the user, and the adjustment is performed as follows:

$$f(x_i) = \frac{x_i}{C_i}$$

or

$$f(x_i) = x_i - C_i$$

if `mode` is set to `diff`.

6 Outlier Detection

The **X-13 ARIMA-SEATS** program provides capabilities for detecting and adjusting various types of outliers. The program can detect the following types of outliers:

1. **Additive Outlier (AO)**: This is an unexpected high or low value at a given time point.
2. **Level Shift Outlier (LS)**: This occurs when the mean of a series changes abruptly.
3. **Temporary Change (TC)**: This occurs when the series mean changes temporarily and shows a decay to the previous mean.

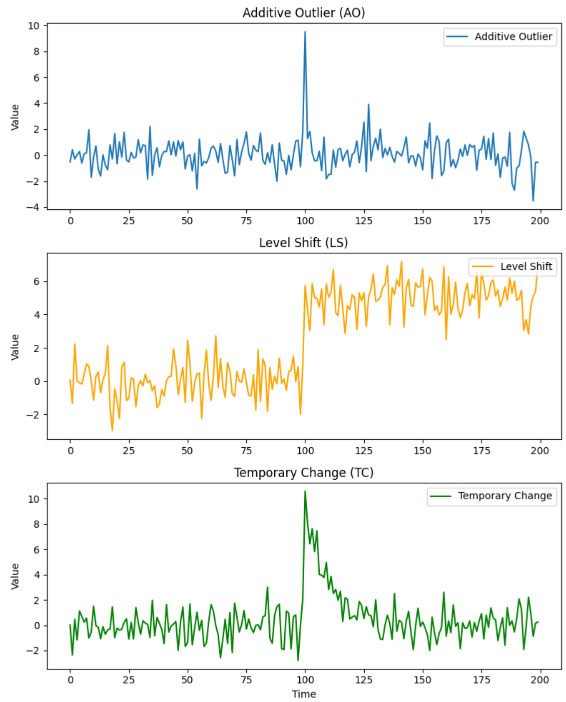

Figure 3: Visual Representation of Outlier Types: Additive Outlier (AO), Level Shift (LS), and Temporary Change (TC)

An outlier at a time point t_0 can be detected using regression variables. Additive outliers at t_0 can be detected as follows:
Suppose $Y = (Y_1, Y_2, \ldots, Y_N)^T$ is our time series with outliers that follows a regARIMA $(p, d, q)(P, D, Q)_s$ process. Let X_t be the vector of regression variables for the t-th observation. The model can be written as:

$$Y_t = X_t^T \beta + Z_t$$

where Z_t follows an ARIMA $(p, d, q)(P, D, Q)_s$ process.
To detect an outlier, say an additive outlier at time t_0, we add a new regression variable $AO_t^{(t_0)}$ to the X_t vector, so:

$$\tilde{X}_t^T = [X_t^T \quad AO_t^{(t_0)}]^T$$

where $AO_t^{(t_0)}$ is defined as:

$$AO_t^{(t_0)} = \begin{cases} 1, & \text{if } t = t_0 \\ 0, & \text{if } t \neq t_0 \end{cases}$$

This regressor contributes to Y only at time point t_0. The equation for the time series with an additive outlier (AO) included becomes:

$$Y_t = \tilde{X}_t^T \tilde{\beta} + \tilde{Z}_t$$

Now, we can fit the new model and check if the coefficient corresponding to the regressor $AO_{(t_0)}$ is significantly different from 0. The program calculates the t-value for the regression coefficient and compares it with a critical value. The critical values are taken from a table, which contains the critical values used for different values of N. This table is based on the researcher's experience. Similarly, the other two types of outliers can be modeled using such regressors.

- **Level Shift (LS)**: We can define a regressor as follows:

$$LS^t_{(t_0)} = \begin{cases} -1, & t < t_0 \\ 0, & t \geq t_0 \end{cases}$$

- **Temporary Shift Change (TC)**: We define a regressor as follows:

$$TC^t_{(t_0)} = \begin{cases} 0, & t < t_0 \\ \alpha^{(t-t_0)}, & t \geq t_0 \end{cases}$$

  where α is called the decay rate.

$L^{(a)}_t$ signifies that the mean before time t_0 was less than the mean after time t_0. The critical decay rate for $TC_{(t_0)}$ can be specified by the user. The default value is $\alpha = (0.7)^{1/s}$ (s is the number of observations in a year). The regressors in the regression matrix may look like this:

	AO1953.02	LS1953.04	TC1953.03
1953.01	0	−1	0
1953.02	1	−1	0
1953.03	0	−1	1
1953.04	0	0	0.5
1953.05	0	0	0.25
1953.06	0	0	0.125

here $\alpha = 0.5$.
When an outlier is detected, it can be adjusted by using the corresponding regressor in the model. The user can specify the type of outlier they want the program to detect automatically using the `type` argument of the **outlier_spec**.

```
outlier{
types= (ls ao tc)
}
```

The default for types argument in the outlier spec is `types = (ls ao)`, but the user can provide any combination of ao, ls, and tc.

The automatic outlier detection procedure discussed here is used to detect only Additive Outliers (AO), Level Shifts (LS), and Temporary Changes (TC) but X-13 ARIMA-SEATS also provides functionality to check for other types of outliers at specific time points provided by the user. For instance, if a user suspects a temporary level shift from time t_0 to t_1, indicating that the mean of the series changes temporarily between these points, they can specify the corresponding time range for the regressor in the regression specification as follows:

```
regresssion{variables = ts1952.03-1954.06}
```

This approach differs from the automatic outlier detection procedure because the specific time points for potential outliers must be predefined by the user. In contrast, the automatic detection process tests for the presence of a given type of outlier at all possible time points.
There are two algorithms available for automatic outlier detection:

1. **Addone Method**: Tests one potential outlier at a time.
2. **AddAll Method**: Tests all potential outliers at once.

The default method is **Addone**.

Regression effect	Variable definition(s)
Seasonal Outlier at t_0 `sodate`$_{t_0}$	$$SO_t^{(t_0)} = \begin{cases} 0, & \text{for } t \geq t_0 \\ -1, & \text{for } t < t_0 \\ 1/(s-1), & \text{otherwise} \end{cases}$$
Ramp, to t_1 `rpdate`$_{t_0}$`-date`$_{t_1}$	$$RP_t^{(t_0,t_1)} = \begin{cases} t_0 - t_1, & \text{for } t \leq t_0 \\ t - t_1, & \text{for } t_0 < t < t_1 \\ 0, & \text{for } t \geq t_1 \end{cases}$$
Quadratic Ramp (Increasing), to t_1 `qidate`$_{t_0}$`-date`$_{t_1}$	$$QI_t^{(t_0,t_1)} = \begin{cases} -(t_1 - t_0)^2, & \text{for } t \leq t_0 \\ (t - t_0)^2 - (t_1 - t_0)^2, & \text{for } t_0 < t < t_1 \\ 0, & \text{for } t \geq t_1 \end{cases}$$
Quadratic Ramp (Decreasing), to t_1 `qddate`$_{t_0}$`-date`$_{t_1}$	$$QD_t^{(t_0,t_1)} = \begin{cases} -(t_1 - t_0)^2, & \text{for } t \leq t_0 \\ -(t - t_0)^2 + (t_1 - t_0)^2, & \text{for } t_0 < t < t_1 \\ 0, & \text{for } t \geq t_1 \end{cases}$$
Temporary Level Shift, to t_1 `tl1date`$_{t_0}$`-date`$_{t_1}$	$$TL_t^{(t_0,t_1)} = \begin{cases} 1, & \text{for } t_0 \leq t \leq t_1 \\ 0, & \text{otherwise} \end{cases}$$
Additive Outlier Sequence, to t_1 `aosdate`$_{t_0}$`-date`$_{t_1}$	This adds a sequence of AO variables (as defined previously) beginning at t_0 and ending on t_1. For example, `aos1950.jan-1950.oct` is equivalent to listing `ao1950.jan ao1950.feb ao1950.mar ... ao1950.oct` individually. For convenience, `0.0` represents the end of the series; e.g., `aos2020.jan-0.0` would add a sequence of AO variables beginning at January 2020.
Level Shift Sequence, to t_1 `lssdate`$_{t_0}$`-date`$_{t_1}$	The level shift counterpart to AOS, this adds a sequence of LS variables (as defined previously) beginning at t_0 and ending on t_1.

Table 2: Outlier regression effects and their variable definitions

6.1 Outlier Flagging Process

Let $Y = Y_1, \ldots, Y_N$ represent the time series, and let

$$X = \begin{bmatrix} X_1^T \\ X_2^T \\ \vdots \\ X_N^T \end{bmatrix}$$

represent the original regression matrix. Suppose we are detecting level shifts and additive outliers. The process of outlier flagging involves adding a regression variable corresponding to an outlier at a given time point to the model and fitting the model to find parameter estimates. The t-value for an additive outlier $AO_{(t_0)}$ is defined as:

$$t_{AO^{(t_0)}} = \frac{\beta_{AO^{(t_0)}}}{\sqrt{\mathrm{Var}(\beta_{AO^{(t_0)}})}}$$

where $t_{AO^{(t_0)}}$ is the t-value for the regression coefficient $\beta_{AO^{(t_0)}}$ corresponding to the regressor $AO^{(t_0)}$. If $|t_{AO^{(t_0)}}|$ exceeds the critical value, the point is flagged as an outlier. Proceeding with the detection at the next time point, we add $AO^{(t_0+1)}$ to the original set of regressors X_t and follow the same process. This process is repeated across all time points and outlier types to make a list of flagged outliers. If during this process the regression matrix becomes singular we ignore it and move to the next time point.

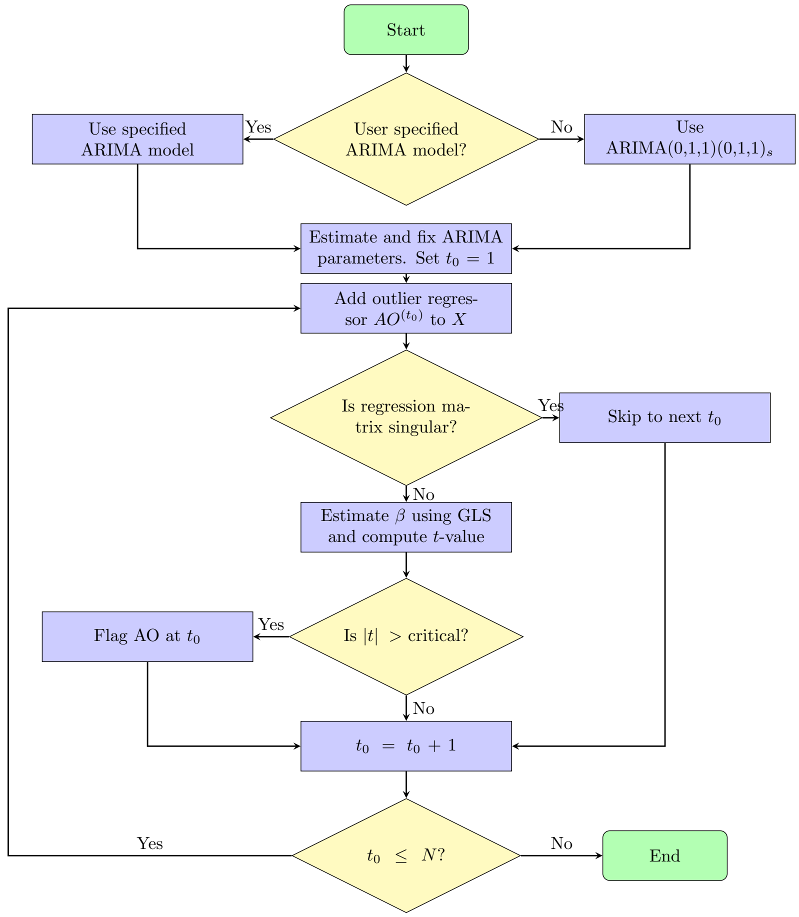

Figure 4: Flowchart of the Additive Outlier Flagging Process

In outlier detection, fitting the model repeatedly for all outliers can be time- and resource-intensive. To address this, the program employs the following strategy:
We aim to fit the model:

$$Y_t = X\beta + Z_t$$

where $Z_t \sim \text{ARIMA}(p, d, q)(P, D, Q)$.
This involves estimating two sets of parameters:

- β: the regression parameters
- Parameters of the ARIMA process

If the user has specified a particular ARIMA $(p, d, q)(P, D, Q)$ model, it is used to estimate the parameters of the ARIMA part in $Y_t = X\beta + Z_t$, otherwise the default model ARIMA$(0, 1, 1)(0, 1, 1)_s$ is used. The model parameters are then fixed at the estimated values. During the flagging process, when new regression variables are added, we calculate only the regression parameter estimates, β, using **Generalized Least Squares (GLS)** estimation as follows: The log-likelihood maximization reduces to a weighted least squares estimation:

$$L_N = -\frac{1}{2}\log(2\pi\Omega) - \frac{1}{2}(Y - X\beta)^T\Omega^{-1}(Y - X\beta)$$

Maximizing L_N is equivalent to minimizing:

$$\min_{\beta}(Y - X\beta)^T\Omega^{-1}(Y - X\beta)$$

Since Ω depends only on the parameters of the ARIMA model (which are fixed), there is a closed-form solution for β:

$$\hat{\beta} = (X^T\Omega^{-1}X)^{-1}X^T\Omega^{-\frac{1}{2}}Y$$

where $\Omega^{-\frac{1}{2}}$ is a matrix such that $(\Omega^{-\frac{1}{2}})(\Omega^{-\frac{1}{2}}) = \Omega^{-1}$, and $\Omega^{-\frac{1}{2}}$ exists if Ω is positive definite.

6.2 AddOne Method

It is the default method of outlier detection which consists of identifying outliers and adding the most significant one to the model. This process is repeated till no outliers are left. The following is an outline of the algorithm.

1. **Step 1**: Flag all potential outliers using the current set of regression variables.
2. **Step 2**: Identify the most significant outlier and add it to the list of regression variables. This outlier remains in the regression matrix for subsequent iterations.
3. **Step 3**: If no new outliers are detected, proceed to Step 4. Otherwise, return to Step 1 with the updated list of regressors. Note that in subsequent iterations, the t-value for this flagged outlier will be set to zero to prevent reflagging.
4. **Step 4**: Now that no new outliers are identified, we remove previously added outliers that have become insignificant. The model is refitted, and t-values for each previously flagged outlier are recalculated. Outliers with t-values below the critical threshold are flagged for removal.
5. **Step 5**: If outliers flagged in Step 4 are found, remove them from the set of regressors and return to Step 4. Otherwise, the process stops.

6.3 AddAll Method

The steps in the **AddAll** method are the same as those in the **AddOne** method, except that during the forward pass, all flagged regressors are added to the regression variable list rather than only the regressor with the highest t-value. However, in the backward pass, flagged outliers are still removed one at a time.

Remarks

1. Different methods may result in different sets of outliers. The choice of ARIMA model may also affect the outliers detected. One should be aware that certain combinations of outliers produce arithmetically equivalent effects. For example, the following are equivalent:

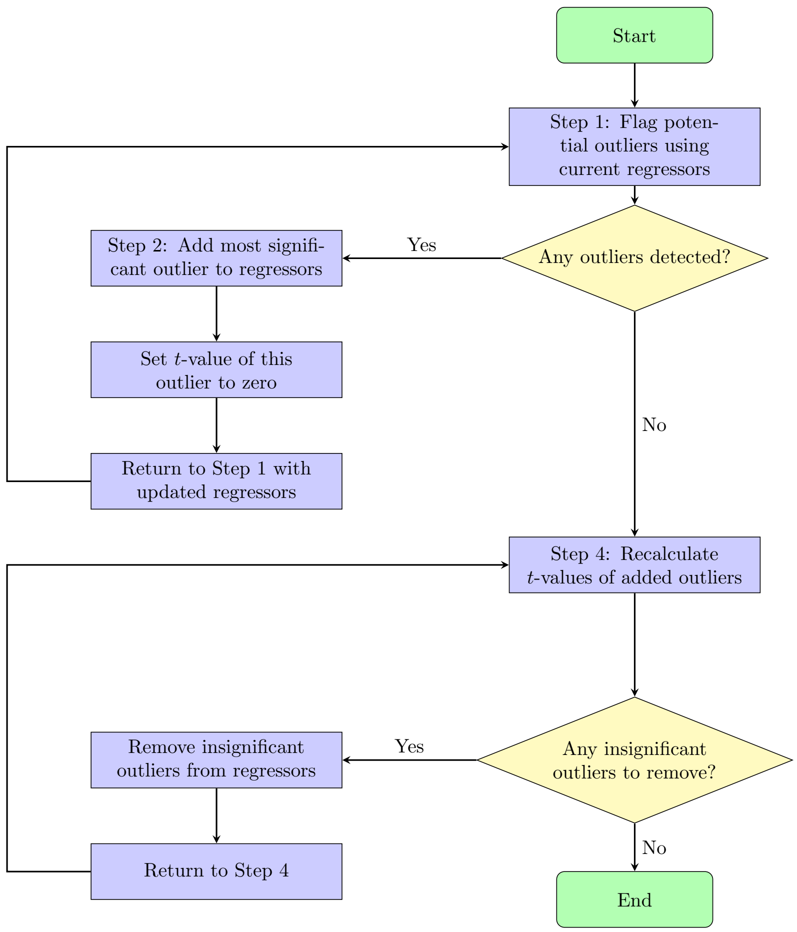

Figure 5: Flowchart of the AddOne Method for Outlier Detection

- (i) An Additive Outlier (AO) at time t_0 followed by a Level Shift (LS) at $t_0 + 1$.
- (ii) Level Shifts (LS) at both t_0 and $t_0 + 1$.
- (iii) Both an AO and an LS at t_0.

However, an LS at t_0 followed by an AO at t_0+1 is not equivalent to these other combinations. Because during seasonal extraction AOs are assigned to the irregular component and LSs to the trend-cycle, one might prefer one combination of equivalent outliers over another based on the intended interpretation.

2. Certain outliers cannot be distinguished or calculated at specific data points:

   - An LS at the first data point cannot be estimated since the level of the series prior to the given data is unknown. Therefore, no LS test statistic is calculated for the first data point.
   - An LS at the last data point cannot be distinguished from an AO there, and an LS at the second data point cannot be distinguished from an AO at the first data point. Hence, LS statistics are calculated for the second and last data points only if AOs are not also being detected.
   - A temporary change (TC) outlier at the last data point cannot be distinguished from an AO there, so no TC statistic is calculated for the last data point if an AO is also being detected.

   LS and TC test statistics that are not calculated due to these limitations are set to zero during the flagging process.

3. When a model contains two or more level shifts (including those obtained from outlier detection as well as any specified in the regression spec), **X-13ARIMA-SEATS** can optionally produce t-statistics for testing the null hypothesis that each sequence of two, three, etc., successive level shifts cancels to yield a net effect of zero beyond the last level shift in the sequence. The t-statistics are computed as the sums of the estimated parameters for each sequence of successive level shifts divided by the appropriate standard error. An insignificant t-statistic (e.g., one less than 2 in magnitude) fails to reject the null hypothesis that the corresponding level shifts offset each other.

   - Two successive level shifts cancel if the sum of their corresponding regression parameters is zero, which can be re-expressed as a temporary level shift starting at the time of the first level shift and ending one time point before the second level shift.
   - Similarly, three successive level shifts cancel if the sum of their three regression parameters is zero, and these can be re-expressed as two adjacent temporary level shifts.

   There is a user-specified limit on the number of successive level shifts in the sequences tested. These cancellation tests help users assess the impacts of level shifts in a model. If one or more of these t-statistics are insignificant, the user might consider re-specifying the model with the relevant level shift regression variables replaced by appropriate temporary level shift variables.

4. During the forward pass, a robust estimate of σ is used, which is calculated as follows:

$$\hat{\sigma} = 1.48 \times \text{median}\{|Z_t - \tilde{Z}_t| : t \in [1, 2, \ldots, N]\}$$

   where Z_t represents the observed residuals, and $\tilde{Z}_t$ are the fitted values. This robust estimate helps to limit the influence of outliers during the initial outlier detection process.

5. During the backward pass, the usual mean square error (MSE) is used to estimate σ:

$$\hat{\sigma}^2 = \frac{1}{N} \sum_{t=1}^{N} \left(Z_t - \tilde{Z}_t \right)^2$$

   This approach provides a standard estimate of variance once the initial set of outliers has been flagged and potentially mitigates the effects of outliers on the variance estimate.

7 Model Estimation

The `automdl` specification in **X-13ARIMA-SEATS** allows for automatic model selection based on a modified version of the TRAMO program developed by Gomez and Maravall. The procedure can be outlined as follows:

1. **Default Model Estimation**: A default model, usually the airline model ARIMA(0 1 1)(0 1 1)$_s$, is estimated. Initial outlier detection, regressor significance tests, and residual diagnostics are performed.

2. **Differencing Order Identification**: Empirical unit root tests are conducted to determine the appropriate differencing orders for the model.

3. **ARMA Model Order Identification**: An iterative procedure is applied to determine the orders of the AR and MA parameters.

4. **Comparison of Identified Model with Default Model**: The identified model is compared with the default model using diagnostic measures.

5. **Final Model Checks**: The chosen model undergoes further diagnostics to ensure adequacy.

7.1 Default Model Selection

The first step of the automatic outlier procedure is to estimate a default model. For monthly and quarterly series, this is initially an *airline* model: ARIMA(0 1 1)(0 1 1)$_s$.
The default model is used to perform several tasks:

- If tests for trading day, Easter, or user-defined regressors are requested by the user in the regression specification, an initial check for the significance of these effects is performed using the default model.

- The `X-13ARIMA-SEATS` program's `aictest` option is utilized to assess the significance of the regressors AICC criteria discussed before.

The procedure then checks the significance of including a constant term in the regARIMA model. This is done by fitting the model without a constant term and a t-statistic for the mean of the model residuals is generated and compared against a critical value of 1.96. Once these tests are complete, the program performs automatic outlier identification if specified by the user in the outlier specification. After outlier identification, the trading day, Easter, and constant regressors are reassessed for significance:

- t-tests are generated.
- A critical value of 1.96 is used to determine if the regressors are significant, except for the constant regressor, which uses the value specified in `armalimit` argument of the `automdl` spec. Default value of `armalimit` is 1.0.
- For the trading day regressor, at least one of the regressors must have a critical value greater than 1.96.

Note that this test is conducted for trading day and Easter regressors only if the `aictest` argument is provided in the regression specification; the constant regressor is always tested.

After determining the regression part of the default model, the program generates residual diagnostics for this model, which include:

- The Ljung-Box Q statistic for the model residuals (at lag 24 for monthly series or lag 16 for quarterly series).
- The confidence coefficient of the Ljung-Box Q statistic.
- A t-value for the mean of the regARIMA model residuals.
- An estimate of the residual standard error.

The confidence coefficient is defined as $1 - p$-value of the Ljung-Box Q statistic, as described in Lehman (1986). The TRAMO documentation (Gómez and Maravall, 1996) refers to the confidence coefficient as the significance level. These diagnostics will later be compared to those of the model selected by the automatic model identification procedure. The model identified by this procedure must show some improvement over the default model in these residual diagnostics; otherwise, the program will accept the default model.

Just before the model identification phase begins, the program removes the regression effects estimated by the default model from the original series. It is this adjusted series, rather than the original series, that is used in the model identification routines. This approach aims to robustify the model identification process, ensuring that the choice of differencing and model orders are not unduly affected by outliers, calendar effects, and other regression effects. In the TRAMO documentation, this adjusted series is referred to as the *linearized series*.

7.2 Identification of Differencing Orders

The purpose of differencing is to make the series stationary. We want to find the appropriate order of differencing so that the differenced series becomes stationary. In the model equation of ARIMA differencing order is represented by a differencing polynomial on the left hand side of the following form:

$$\phi(B)\Phi(B)\delta(B)z_t = \theta(B)\Theta(B)\epsilon_t$$

$$\delta(B) = (1-B)^d(1-B^s)^D$$

Notice that the roots of these polynomials play a very important role. The conditions for stationarity and invertibility for an ARIMA process are checked using the roots. The ARMA part of the model is stationary if all the roots of the AR polynomials i.e.$\phi(B)$ and $\Phi(B)$ have a modulus greater than 1. If the modulus of a root of the AR polynomials is equal to 1 or very close to 1 then it suggests that that factor should be a part of the differencing polynomial, not the AR polynomials. (The AR roots can never have modulus less than one as the series would explode). This means if we fit an AR(2) model to a series and one of the roots of the AR polynomial comes out to be 1 or close to 1 then we should rather fit an ARIMA$(1,1,0)$ model. This will ensure that the unit root is incorporated by the differencing and the ARMA part remains stationary. Following this spirit the program attempts to identify an appropriate order of differencing for the linearized series computed earlier. The maximum permissible values of regular and seasonal differencing are 2 and 1 respectively. This is achieved by performing a series of unit root tests and fitting different ARMA models to the (possibly differenced) linearized series. The estimation of these models utilizes the Hannan-Rissanen method.

Step 1: The first stage of the procedure involves fitting an ARIMA$(2,0,0)(1,0,0)_s$ model to the linearized series using the Hannan-Rissanen method. The real AR roots of the estimated model are then examined. A root is considered a unit root if the modulus of the root is less than 1.042. If such a root is identified, the corresponding order of differencing (seasonal or nonseasonal) is increased by one. If the Hannan-Rissanen procedure estimates a model with roots inside the unit circle, the `X-13ARIMA-SEATS` program re-estimates the model using exact maximum likelihood estimation. The modulus test described above is then applied to the resulting estimates to determine the necessity of additional differencing.

Step 2: If differencing was identified in Step 1, the linearized series is differenced accordingly at the start of Step 2. An ARMA $(1,1)(1,1)_s$ model is then fitted to the differenced series. The roots of the AR polynomial of this model are examined to determine if they are close to one. If no root close to 1 is found then we are done and the differencing identified in step one is chosen for the final model.

1. If an AR coefficient meets the criterion of being close to one, the program checks for a common factor in the corresponding AR and MA polynomials of the ARMA model that can be canceled.

2. If no such cancellation exists, the differencing order is altered by increasing the appropriate differencing order. The linearized series is then differenced using the new set of differencing orders.

3. The ARMA$(1,1)(1,1)_s$ model is re-fitted to the newly differenced series, and the program checks for any additional necessary differencing.

4. This process repeats iteratively until no further differencing is required.

Once the differencing orders are established, a t-statistic for the mean term of the fully differenced series is generated. This statistic is based on either the sample mean (if no differencing is identified) or by adding a constant term to the regARIMA model. The critical value for the test is determined based on the number of observations in the series. This step determines whether we will use an intercept term or not.

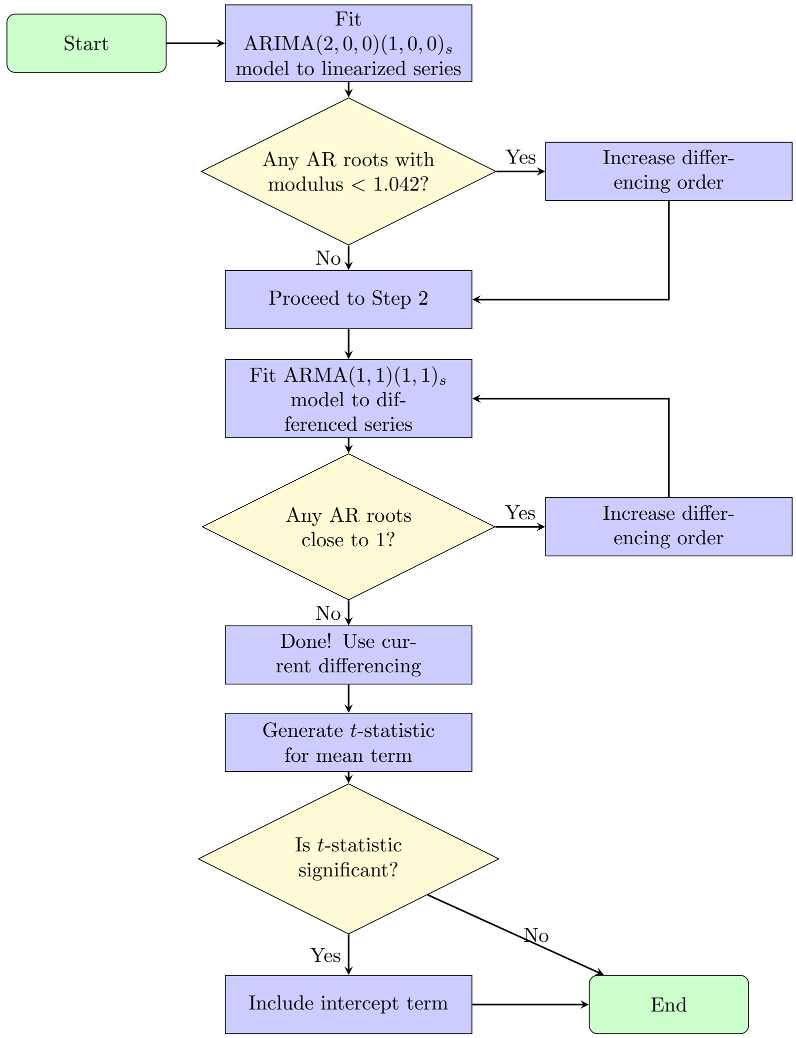

Figure 6: Flowchart for Identifying Appropriate Differencing Orders in ARIMA Modeling

7.3 Identification of ARMA Model Orders

With the appropriate differencing orders determined, the next step involves identifying the orders of the ARMA model components. This process aims to select the best-fitting ARMA model by comparing various candidate models based on their Bayesian Information Criterion (BIC) values. The BIC is a criterion for model selection among a finite set of models. It is based on the likelihood function and includes a penalty term for the number of parameters in the model to discourage overfitting. The classical BIC formula used in the `X-13ARIMA-SEATS` output is:

$$\text{BIC} = -2L_N + n_p \cdot \ln N,$$

where:

- L_N is the maximized value of the log-likelihood function evaluated over N observations.
- n_p is the number of estimated parameters in the model, including the white noise variance.
- N is the number of observations remaining after applying the model's differencing operations.

For the automatic model identification procedure, a variant of BIC used by TRAMO is employed:

$$\text{BIC2} = \frac{-2L_N + n_p \cdot \ln N}{N}.$$

This adjusted criterion allows for a fair comparison of models with different numbers of parameters and observations. The identification process consists of a three-stage procedure, as detailed in Gomez and Maravall (2000), to efficiently explore potential ARMA model orders up to specified maximum orders for regular and seasonal components.

Stage 1: Initial Seasonal Order Selection

1. **Model Specification**: Consider ARIMA models of the form $(3, d, 0)(P, D, Q)_s$, where:
   - d and D are the previously determined regular and seasonal differencing orders, respectively.
   - P and Q range from 0 up to the maximum seasonal order m_s (default is 1).
   - The regular AR order is fixed at 3, and the regular MA order is fixed at 0 for initial estimation.
2. **Model Estimation**: For each combination of P and Q, estimate the model parameters using the exact maximum likelihood.
3. **BIC2 Calculation**: Compute the BIC2 value for each model.
4. **Selection of Seasonal Orders**: Select the pair (P, Q) that results in the lowest BIC2 value.

Stage 2: Nonseasonal Order Selection

1. **Model Specification**: Using the selected P and Q from Stage 1, consider ARIMA models of the form $(p, d, q)(P, D, Q)_s$, where:
   - p and q range from 0 up to the maximum regular order m_r (default is 2).
2. **Model Estimation**: For each combination of p and q, estimate the model parameters.
3. **BIC2 Calculation**: Compute the BIC2 value for each model.
4. **Selection of Nonseasonal Orders**: Select the pair (p, q) that results in the lowest BIC2 value.

Stage 3: Refinement of Seasonal Orders

1. **Model Specification**: With the selected p and q from Stage 2, reconsider ARIMA models of the form $(p, d, q)(P, D, Q)_s$, varying P and Q within the specified seasonal order limits (from 0 to m_s).

2. **Exception**: If no seasonal AR term ($P = 0$) was selected in Stage 1 and seasonal differencing is present ($D > 0$), only $P = 0$ is considered, and Q is varied.

3. **Model Estimation**: Estimate the model parameters for each combination of P and Q.

4. **BIC2 Calculation**: Compute the BIC2 value for each model.

5. **Selection of Seasonal Orders**: Select the pair (P, Q) that results in the lowest BIC2 value.

Model Selection and Final Checks

During the ARMA order selection process, `X-13ARIMA-SEATS` keeps track of the models with the five smallest BIC2 values. Once the identification phase is over, the program compares the BIC2 of the best model with those of the other four models to determine if there are models with BIC2s that are "close enough" such that there is no significant difference between the models. The criteria for being "close enough" depend on several factors, including the length of the series and the magnitude of the differences between the BIC2 values. (No further information about this was available.) If the program finds a model that is close enough to the best model, it also checks whether the alternative model is more parsimonious, especially in the seasonal operator, than the best model. Parsimony refers to the simplicity of the model—the model with fewer parameters is considered more parsimonious. If the alternative model is more parsimonious, the program will accept it over the model with the lowest BIC2.

The program also checks for *model balance*. A model is said to be more balanced than a competing model if the absolute difference between the total orders of the AR (including differencing) and MA operators is smaller. In mathematical terms, for a model with AR order p, differencing order d, and MA order q, the total orders are $p+d$ and q, respectively. A balanced model minimizes $|(p + d) - q|$. While balanced models are useful for model-based seasonal adjustment, it is unclear whether this criterion is beneficial for the types of operations performed by `X-13ARIMA-SEATS`. This is because it can introduce a small bias toward mixed ARMA models, which can be difficult to estimate due to near cancellation of terms. `X-13ARIMA-SEATS` makes checking for model balance optional at this stage; by default, it does not test for model balance.

If the identified model is different from the default model, the program redoes several steps that determined the regressors of the default model:

- **Outlier Regressors**: Outlier regressors identified for the default model are removed from the identified model. Outlier detection is then re-performed using the identified model to ensure that any new outliers are appropriately accounted for.

- **Regressor Testing**: If the user has specified Akaike Information Criterion (AIC) testing of trading day effects, Easter effects, or user-defined regressors, this testing is redone using the identified model. This ensures that the inclusion or exclusion of these regressors is optimal for the new model.

Subsequently, outlier identification is redone for the identified model. By repeating these steps, the program ensures that all aspects of the model specification are consistent and optimized for the selected ARIMA structure.

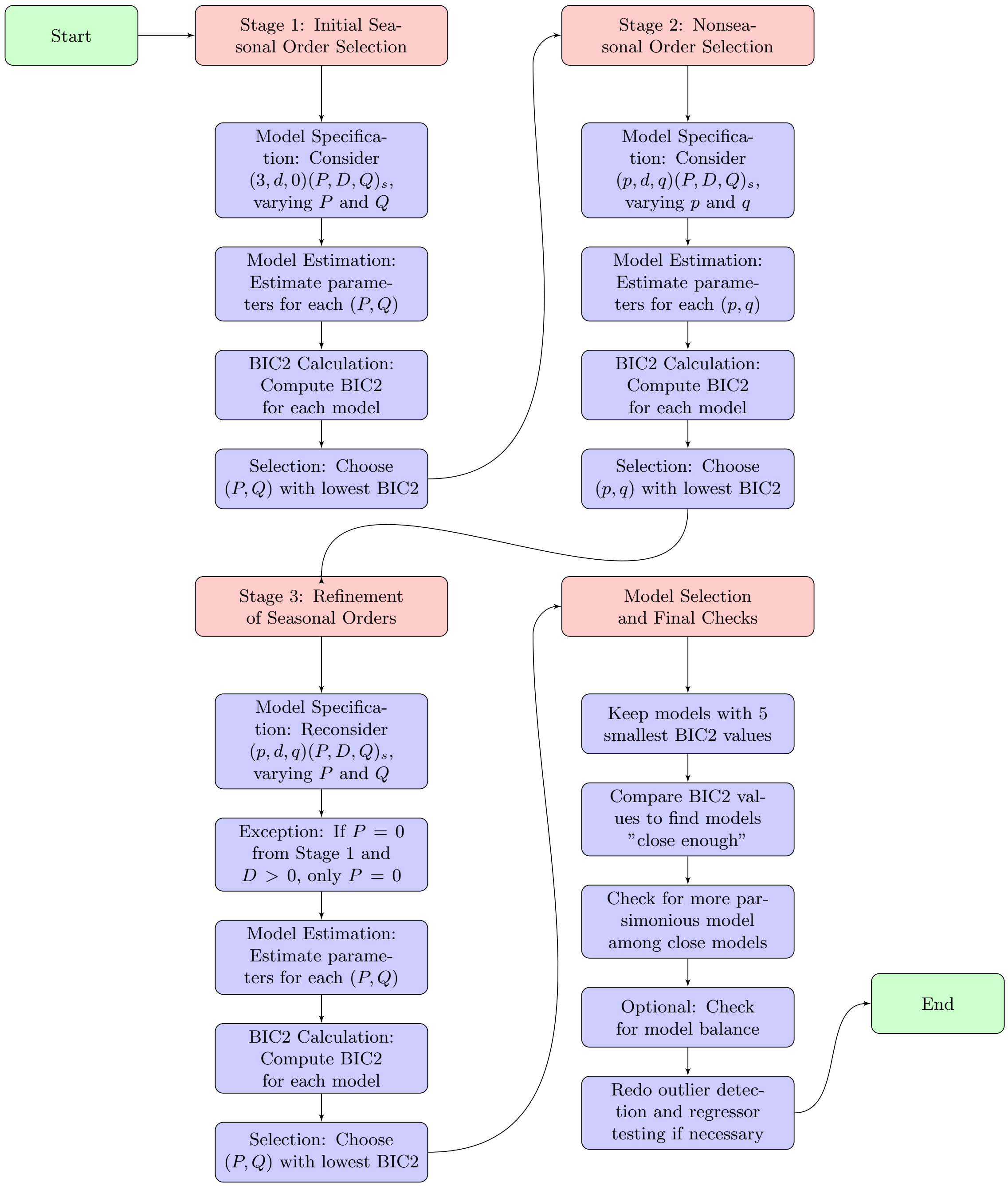

Figure 7: Flowchart for Identification of ARMA Model Orders