Introduction and Objective
Data Description
Exploratory Data Analysis
Modeling
Conclusion
Reference

Code

STATS-531 - Midterm Project

Introduction and Objective

Environmental noise pollution is an important issue in urban planning and public health. One of the key metrics for assessing environmental noise is the **equivalent continuous sound level (Leq dBA)**[1], which represents an averaged sound level over a given period. Noise levels exhibit strong temporal patterns influenced by human activity, weather conditions, and geographical factors. Understanding these temporal dynamics can aid in mitigating noise pollution and optimizing urban noise management strategies.

In this project, we investigate the following research question:

What are the main temporal trends, including seasonal, autoregressive, and stochastic components, in the time series of environmental noise levels?

To address these questions, I analyze a dataset containing Leq dBA noise measurements recorded at 10-minute intervals. We employ **exploratory data analysis (EDA), stationarity testing, seasonal decomposition, and time series modeling techniques**, including **ARIMA** and **SARIMA (Seasonal ARIMA)** models, to capture and forecast noise level fluctuations.

Data Description

The dataset consists of **average Leq noise measurements (dBA) recorded every 10 minutes** over a multi-day period. The variables include:

- **time_point**: Timestamp of the measurement.
- **Leq_dBA**: The equivalent continuous sound level in decibels.

This dataset enables us to analyze the underlying time-dependent structure of noise levels, including daily patterns and potential long-term trends.

Exploratory Data Analysis

Before building a forecasting model, we first visualize the raw time series data.

Hide

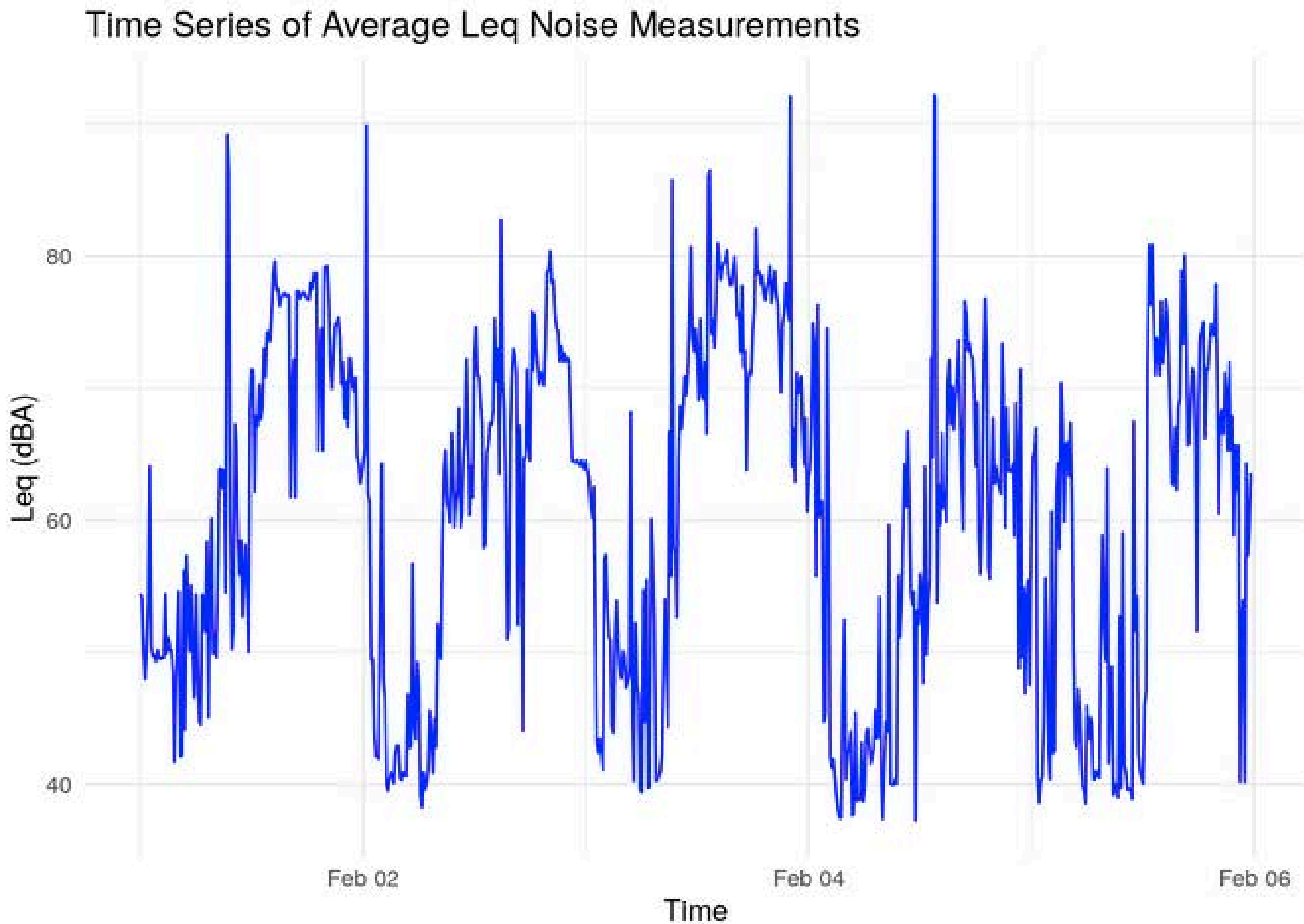
```
#  Load data and Preprocess
noise <- fread("noise_10min_interval.csv")[,-1]

# Convert timestamp to POSIXct format and set as index
noise <- noise %>%
  mutate(time_piont = ymd_hms(time_point)) %>%
  arrange(time_point)

# Check for missing values
sum(is.na(noise))
```

```
## [1] 0
```

Data visualization

Hide

```
ggplot(noise, aes(x = time_point, y = Leq_dBA)) +
  geom_line(color = "blue") +
  labs(title = "Time Series of Average Leq Noise Measurements",
       x = "Time",
       y = "Leq (dBA)") +
  theme_minimal()
```



The time series plot reveals no clear trend over time. However, there is noticeable variability in the data, suggesting that the variance may also be changing over time. These characteristics are indicative of a non-stationary time series.

Check Stationarity

Augmented Dickey-Fuller Test

Hide

```
# Perform Augmented Dickey-Fuller Test
adf_test <- adf.test(noise$Leq_dBA, alternative = "stationary")
adf_test
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  noise$Leq_dBA
## Dickey-Fuller = -3.0426, Lag order = 8, p-value = 0.137
## alternative hypothesis: stationary
```

The ADF test was conducted to statistically assess the stationarity of the series. The test yielded a p-value greater than the significance level of 0.05, leading us to fail to reject the null hypothesis that the series possesses a unit root. This statistical evidence supports the visual assessment that the series is non-stationary.

Differencing

Hide

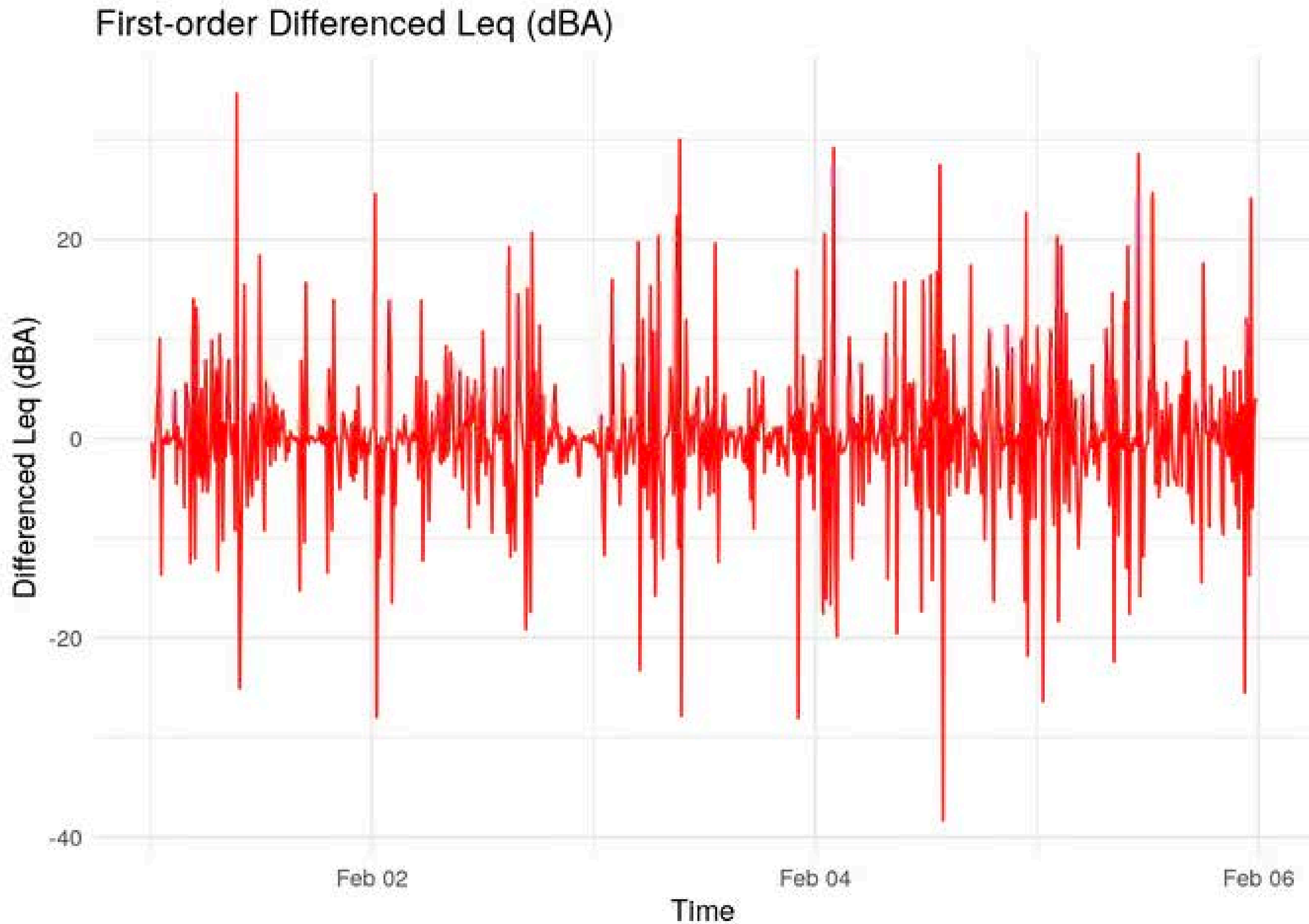
```
noise$diff_Leq <-c(NA, diff(noise$Leq_dBA, lag = 1, differences = 1))
```

Apply first-order differencing to remove the trend component.

Recheck stationarity

Hide

```
ggplot(noise, aes(x = time_point, y = diff_Leq)) +
  geom_line(color = "red") +
  labs(title = "First-order Differenced Leq (dBA)",
       x = "Time",
       y = "Differenced Leq (dBA)") +
  theme_minimal()
```



Hide

```
adf.test(na.omit(noise$diff_Leq), alternative = "stationary")
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  na.omit(noise$diff_Leq)
## Dickey-Fuller = -12.885, Lag order = 8, p-value = 0.01
## alternative hypothesis: stationary
```

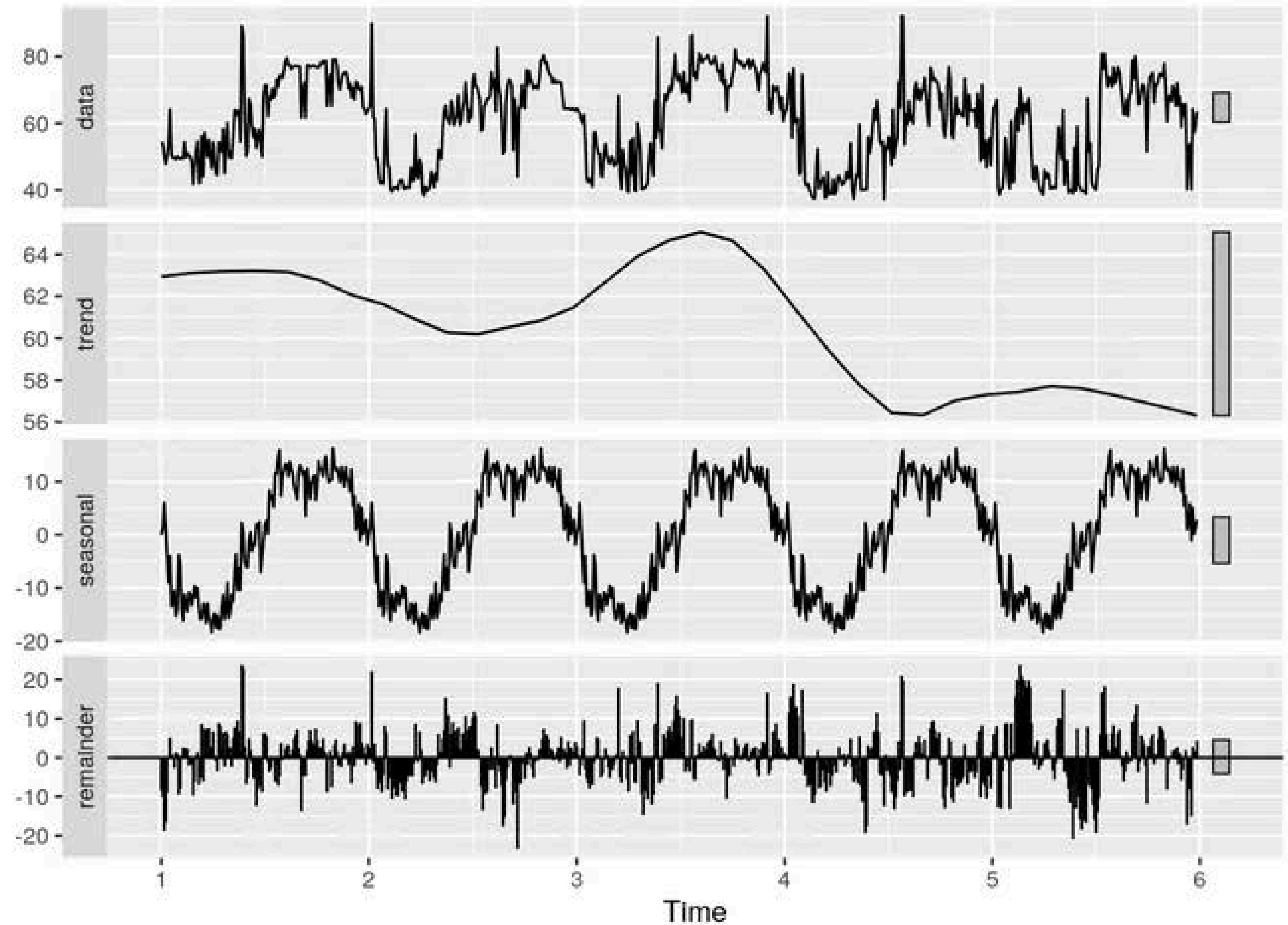
The first-order differenced time series plot shows that the transformed series fluctuates around a mean of zero, with no apparent trend. The variance appears relatively stable over time, suggesting that the series no longer exhibits heteroscedasticity. However, some large spikes remain, which may be attributed to occasional outliers or periods of increased variability. Despite these fluctuations, the overall characteristics of the series now align with the properties of a stationary time series, making it more suitable for further modeling.

The results of ADF test provide statistical confirmation of stationarity.

Check for Seasonality

Hide

```
decomposition <- stl(ts(noise$Leq_dBA, frequency = 144), s.window = "periodic")
autoplot(decomposition)
```

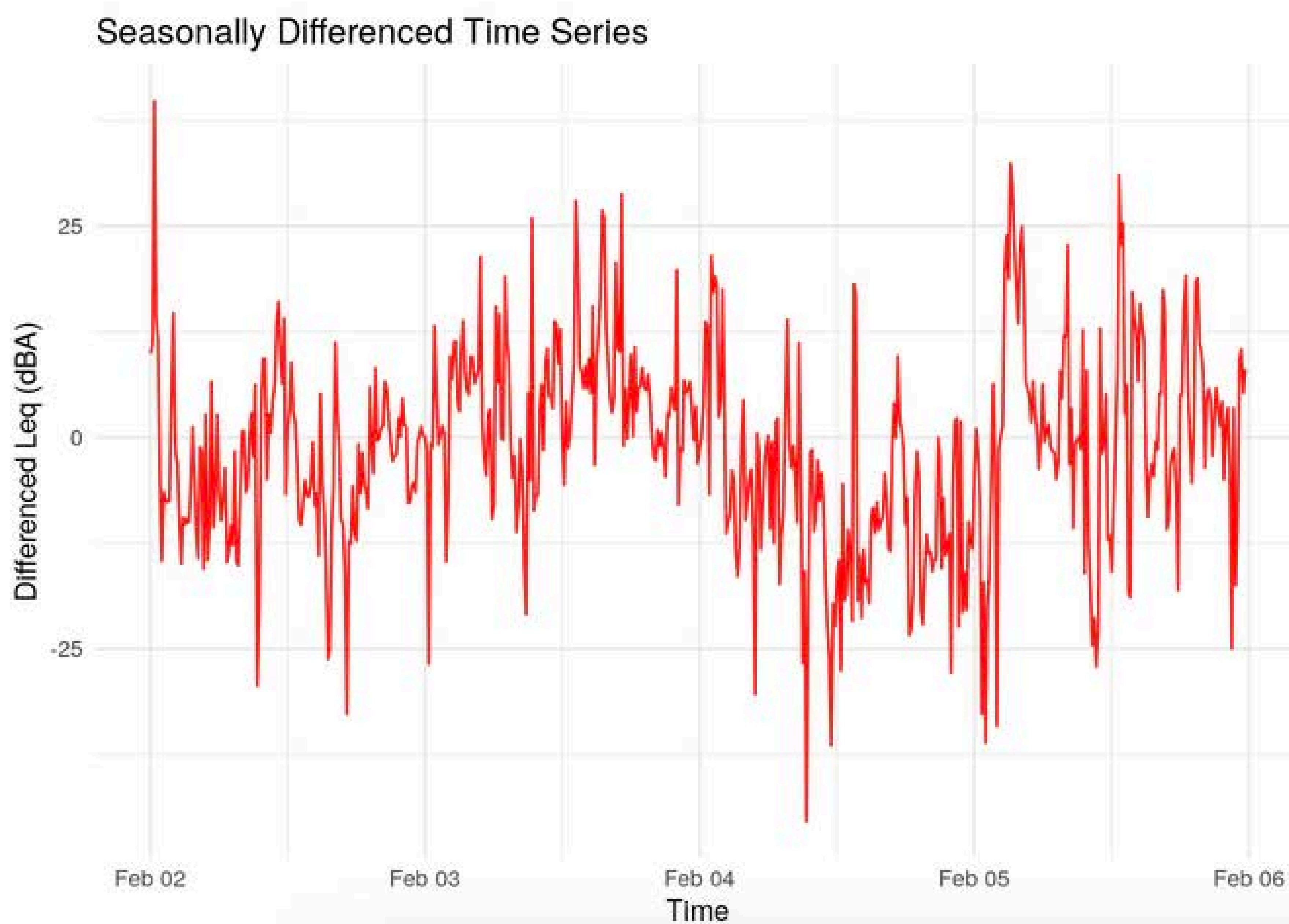


The STL decomposition [2] of the Leq (dBA) time series reveals distinct components. The trend component shows a slow-moving variation over time, indicating that there may be long-term changes in noise levels. The seasonal component displays a repeating pattern, suggesting the presence of periodic fluctuations, likely corresponding to daily or other cyclic environmental influences. The remainder component (residuals) appears to exhibit random noise but also contains some noticeable spikes, which may correspond to anomalies or events not captured by the trend or seasonality. Given the clear seasonal pattern, a seasonal ARIMA (SARIMA) model may be more appropriate than a standard ARIMA model.

Seasonally Differencing

Hide

```
noise$diff_seasonal <- c(rep(NA, 144), diff(noise$Leq_dBA, lag = 144))

ggplot(na.omit(noise), aes(x = time_point, y = diff_seasonal)) +
  geom_line(color = "red") +
  labs(title = "Seasonally Differenced Time Series",
       y = "Differenced Leq (dBA)", x = "Time") +
  theme_minimal()
```



```
adf.test(na.omit(noise$diff_seasonal), alternative = "stationary")
```

```
## Warning in adf.test(na.omit(noise$diff_seasonal), alternative = "stationary"):
## p-value smaller than printed p-value
```

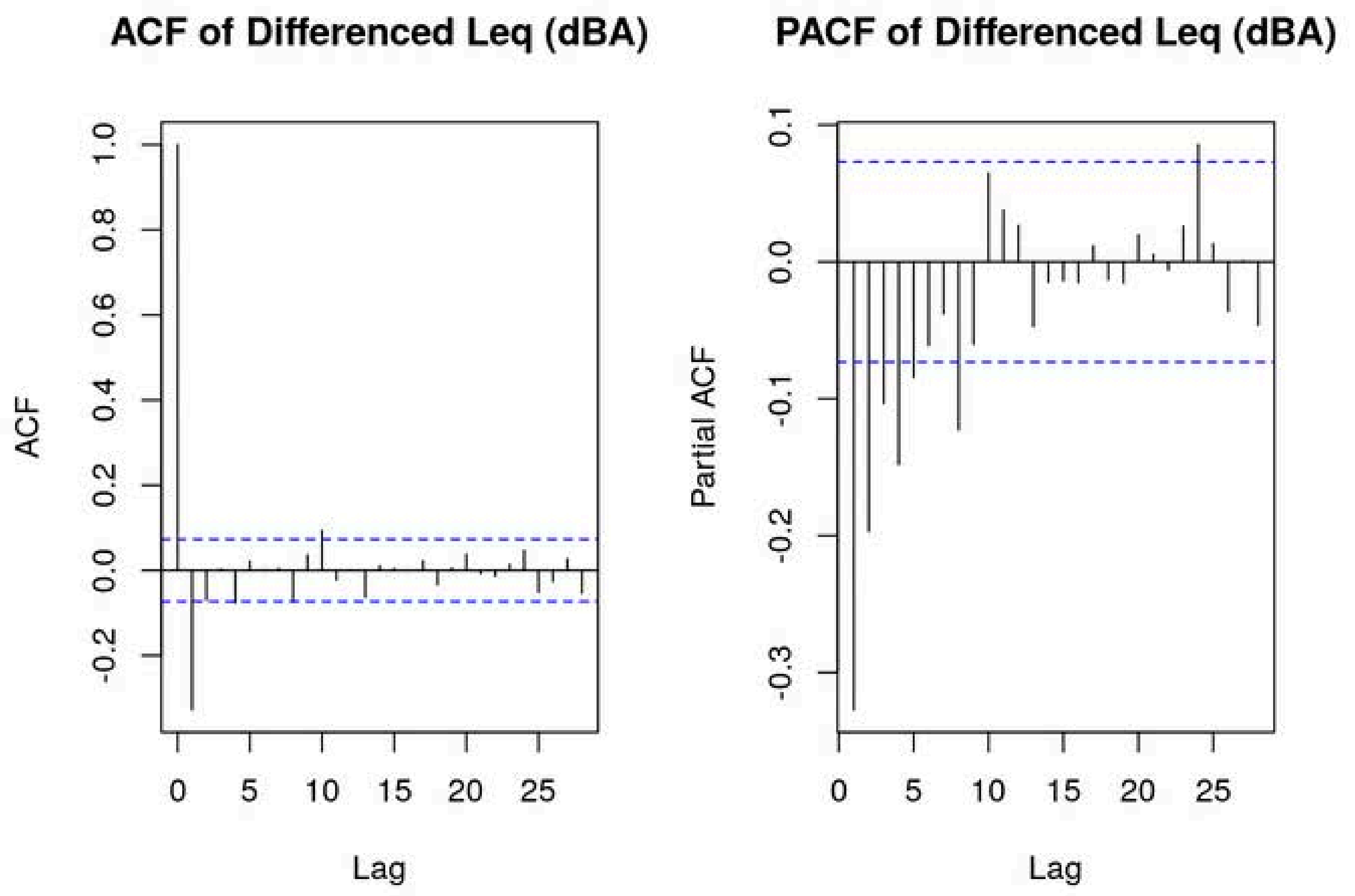
```
##
##  Augmented Dickey-Fuller Test
##
## data:  na.omit(noise$diff_seasonal)
## Dickey-Fuller = -4.1529, Lag order = 8, p-value = 0.01
## alternative hypothesis: stationary
```

The seasonally differenced time series plot indicates that the data fluctuates around a constant mean, with no clear upward or downward trend, suggesting an improvement in stationarity compared to the original series. The ADF test confirms this observation. Therefore, we reject the null hypothesis of non-stationarity, indicating that seasonal differencing alone is sufficient to achieve stationarity.

Check Autocorrelation

Hide

```
# Plot ACF and PACF for the differenced series
par(mfrow = c(1, 2))
acf(noise$diff_Leq[-1], main = "ACF of Differenced Leq (dBA)")
pacf(noise$diff_Leq[-1], main = "PACF of Differenced Leq (dBA)")
```

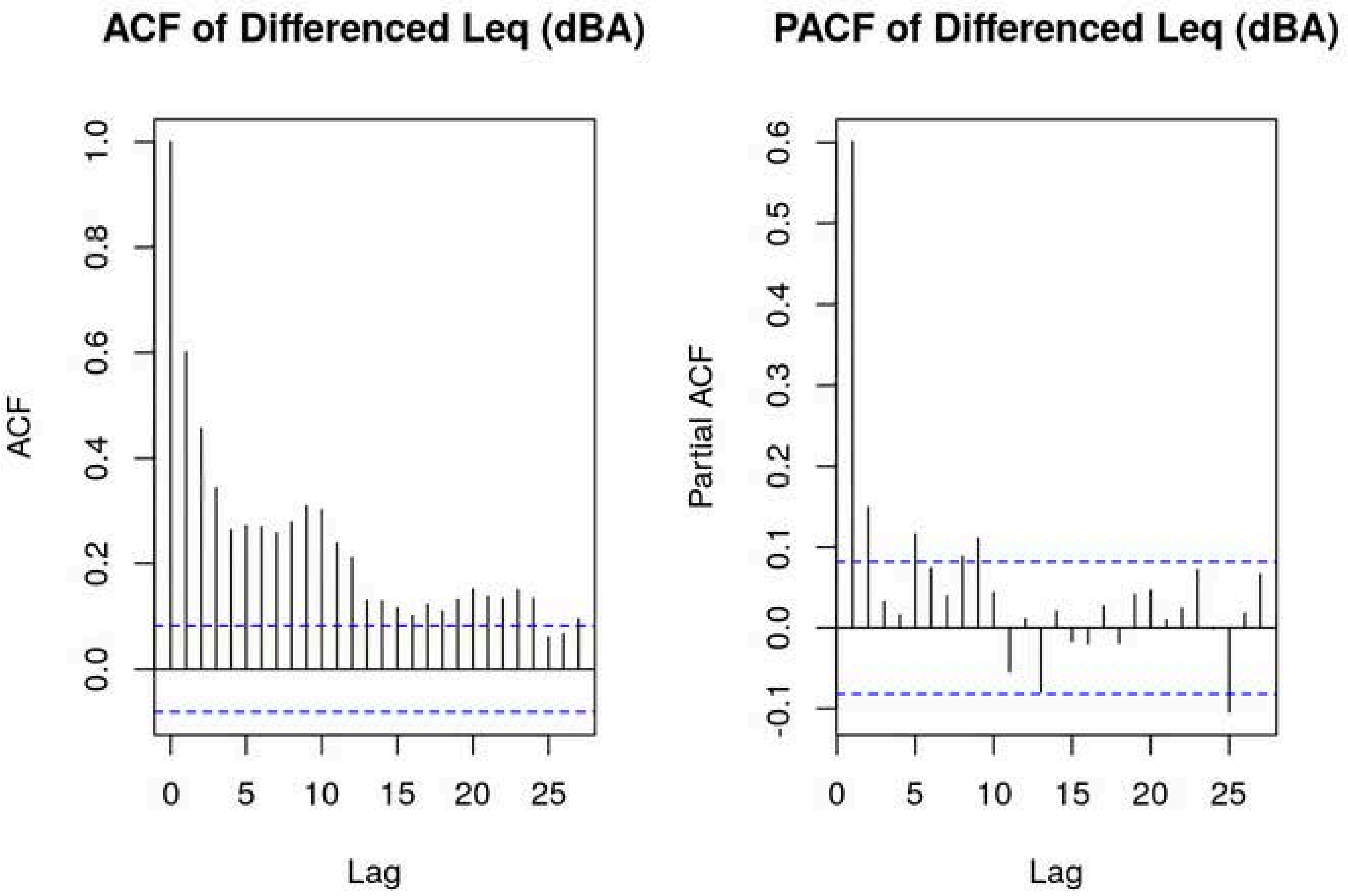


The ACF plot of the differenced Leq (dBA) series shows a significant spike at lag 1, followed by values that quickly drop within the confidence bands. This suggests that the time series has short-term dependencies but does not exhibit a strong long-term autocorrelation pattern. The sharp drop after lag 1 is characteristic of a moving average (MA) process, indicating that an MA component should be included in the ARIMA model.

The PACF plot displays a significant negative spike at lag 1, followed by a gradual decline, with most subsequent lags remaining within the confidence intervals. This pattern is indicative of an autoregressive (AR) process of order 1, where the first lag significantly influences the current value, but higher-order lags do not contribute meaningfully. Given this behavior, an ARIMA(1,1,1) model (one autoregressive term, one differencing step, and one moving average term) appears to be a reasonable starting choice for modeling the time series.

Hide

```
par(mfrow = c(1, 2))
acf(na.omit(noise$diff_seasonal), main = "ACF of Differenced Leq (dBA)")
pacf(na.omit(noise$diff_seasonal), main = "PACF of Differenced Leq (dBA)")
```



The ACF plot shows a gradual decay, with significant spikes at multiple lags, indicating the presence of autocorrelation and suggesting the need for a moving average (MA) component in the SARIMA model. The PACF plot exhibits a sharp drop after lag 1, followed by smaller but still noticeable spikes, suggesting the presence of an autoregressive (AR) component. Based on these observations, an initial SARIMA model with AR terms and MA terms around 1 or 2 could be appropriate, while the seasonal components should also be explored using seasonal ACF/PACF patterns.

Modeling

In this project, we fitted both an ARIMA and a SARIMA model to analyze and forecast noise levels. These models were chosen based on exploratory data analysis, stationarity tests, and seasonal decomposition.

I applied auto.arima() [3] for selecting the best-fitting ARIMA model by searching through multiple combinations of autoregressive, differencing, and moving average terms while optimizing for model performance based on the AIC.

The function in R's package automatically selects the best-fitting ARIMA or Seasonal ARIMA (SARIMA) model for a given time series by minimizing an information criterion such as AIC, AICc, or BIC. The function first determines the appropriate order of differencing (d) by performing unit root tests (e.g., Augmented Dickey-Fuller Test). If the series is non-stationary, differencing is applied:

$$Y_t^{(d)} = (1 - B)^d Y_t$$

where B is the backward shift operator. After ensuring stationarity, the function selects the autoregressive (p) and moving average (q) orders using a stepwise search combined with maximum likelihood estimation. The AR and MA components are modeled as:

$$Y_t = \phi_1 Y_{t-1} + \cdots + \phi_p Y_{t-p} + \epsilon_t$$

$$Y_t = \epsilon_t + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q}$$

If seasonality is detected, the model is extended to a SARIMA representation:

$$\Phi(B^s)(1 - B^s)^D Y_t = \Theta(B^s)\epsilon_t$$

where s is the seasonal period, and P, D, Q are the seasonal orders for AR, differencing, and MA terms. The best model is selected by minimizing the Akaike Information Criterion (AIC):

$$AIC = -2\log L + 2k$$

or its corrected version, AICc:

$$AICc = AIC + \frac{2k(k+1)}{n-k-1}$$

For large datasets, the Bayesian Information Criterion (BIC) is used:

$$BIC = -2\log L + k\log n$$

where L is the likelihood of the model, k is the number of estimated parameters, and n is the sample size. The function iteratively tests different (p, d, q) values and evaluates their statistical significance, ensuring that the final model residuals exhibit white noise behavior. The resulting SARIMA model is formulated as:

$$\Phi(B^s)(1 - B)^d(1 - B^s)^D Y_t = \theta(B^s)\epsilon_t$$

$$\Phi(B^s)(1-B)^d(1-B^s)^D Y_t = \theta(B^s)\epsilon_t$$

where $\Phi(B^s)$ and $\theta(B^s)$ capture seasonal and non-seasonal dependencies. Ultimately, provides an automated yet statistically rigorous way to determine the most appropriate time series model.[4]

ARIMA Model

[5]The ARIMA model is defined as:

$$\phi(B)(1-B)^d Y_t = \theta(B)\epsilon_t$$

where:

- B is the backward shift operator such that $BY_t = Y_{t-1}$.
- d represents the order of differencing applied to make the time series stationary.
- $\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p$ is the autoregressive (AR) polynomial of order p.
- $\theta(B) = 1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^q$ is the moving average (MA) polynomial of order q.
- ϵ_t is a white noise process with mean zero and constant variance.

Fit Model

Show

```
##
##  Fitting models using approximations to speed things up...
##
##  ARIMA(2,0,2) with non-zero mean : 4840.386
##  ARIMA(0,0,0) with non-zero mean : 5742.496
##  ARIMA(1,0,0) with non-zero mean : 4921.39
##  ARIMA(0,0,1) with non-zero mean : 5324.83
##  ARIMA(0,0,0) with zero mean     : 7976.732
##  ARIMA(1,0,2) with non-zero mean : 4839.851
##  ARIMA(0,0,2) with non-zero mean : 5169.173
##  ARIMA(1,0,1) with non-zero mean : 4847.601
##  ARIMA(1,0,3) with non-zero mean : 4840.316
##  ARIMA(0,0,3) with non-zero mean : 5052.846
##  ARIMA(2,0,1) with non-zero mean : 4838.725
##  ARIMA(2,0,0) with non-zero mean : 4872.838
##  ARIMA(3,0,1) with non-zero mean : 4841.069
##  ARIMA(3,0,0) with non-zero mean : 4857.62
##  ARIMA(3,0,2) with non-zero mean : 4843.096
##  ARIMA(2,0,1) with zero mean     : Inf
##
##  Now re-fitting the best model(s) without approximations...
##
##  ARIMA(2,0,1) with non-zero mean : 4838.649
##
##  Best model: ARIMA(2,0,1) with non-zero mean
```

Show

```
## Series: noise$Leq_dBA
## ARIMA(2,0,1) with non-zero mean
##
## Coefficients:
##          ar1      ar2      ma1     mean
##       1.2446  -0.2663  -0.7090  60.4130
## s.e.  0.0709   0.0660   0.0546   3.3486
##
## sigma^2 = 48.49:  log likelihood = -2414.28
## AIC=4838.56   AICc=4838.65   BIC=4861.45
##
## Training set error measures:
##                      ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 0.02360858 6.943963 4.882795 -1.438566 8.637658 0.9921461
##                      ACF1
## Training set -0.004111738
```

Hide

```
arima_model <- auto.arima(noise$Leq_dBA, d = 1, seasonal = FALSE, trace = TRUE)
```

```
##
##  Fitting models using approximations to speed things up...
##
##  ARIMA(2,1,2) with drift         : 4839.751
##  ARIMA(0,1,0) with drift         : 4973.933
##  ARIMA(1,1,0) with drift         : 4895.661
##  ARIMA(0,1,1) with drift         : 4851.272
##  ARIMA(0,1,0)                    : 4971.924
##  ARIMA(1,1,2) with drift         : 4836.801
##  ARIMA(0,1,2) with drift         : 4837.907
##  ARIMA(1,1,1) with drift         : 4835.426
##  ARIMA(2,1,1) with drift         : 4837.742
##  ARIMA(2,1,0) with drift         : 4869.909
##  ARIMA(1,1,1)                    : 4833.414
##  ARIMA(0,1,1)                    : 4849.261
##  ARIMA(1,1,0)                    : 4893.648
##  ARIMA(2,1,1)                    : 4835.736
##  ARIMA(1,1,2)                    : 4834.783
##  ARIMA(0,1,2)                    : 4835.892
##  ARIMA(2,1,0)                    : 4867.894
##  ARIMA(2,1,2)                    : 4837.738
##
##  Now re-fitting the best model(s) without approximations...
##
##  ARIMA(1,1,1)                    : 4836.905
##
##  Best model: ARIMA(1,1,1)
```

Hide

```
summary(arima_model)
```

Hide

```
summary(arima_model)
```

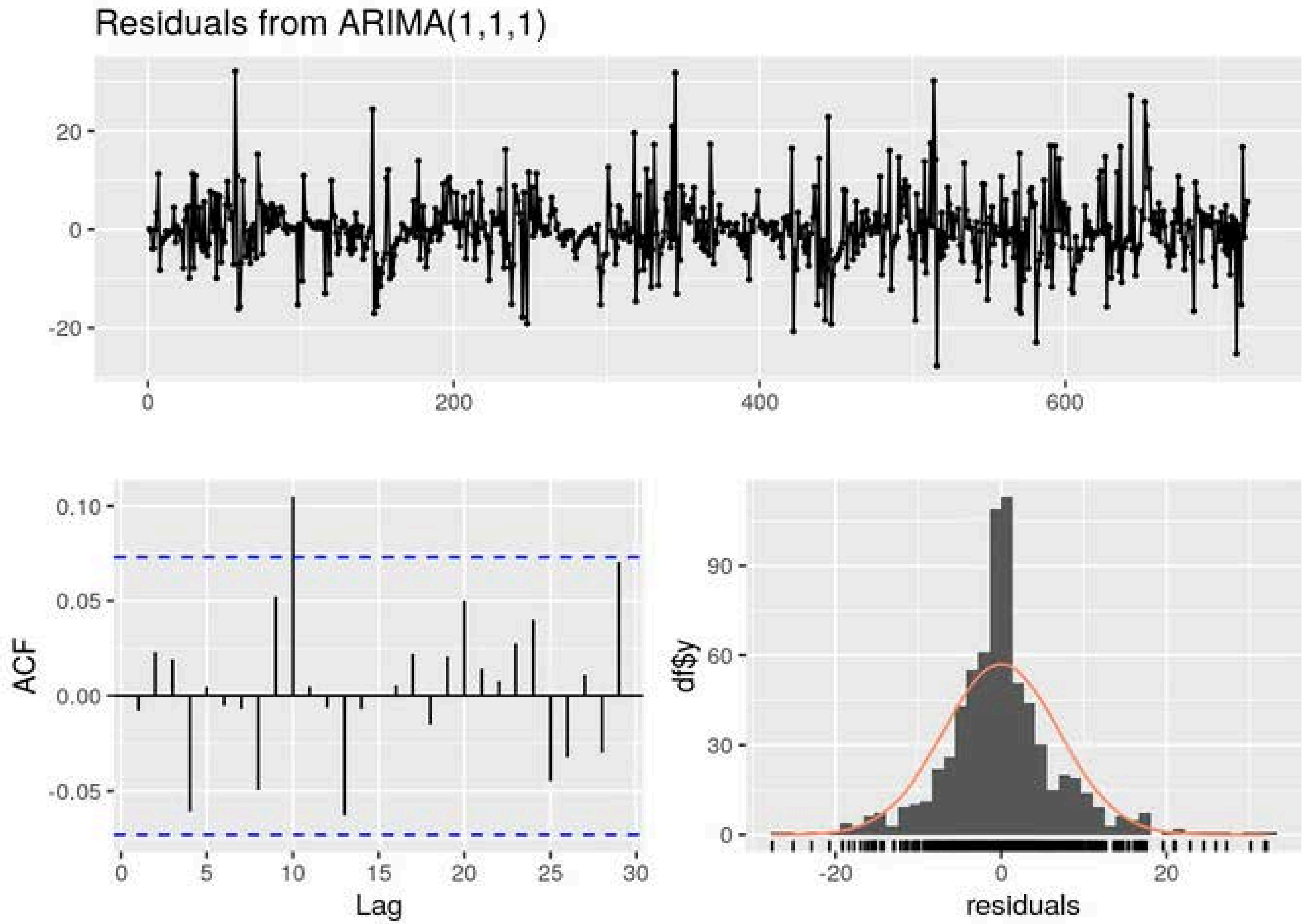
```
## Series: noise$Leq_dBA
## ARIMA(1,1,1)
##
## Coefficients:
##          ar1      ma1
##       0.3018  -0.7499
## s.e.  0.0608   0.0421
##
## sigma^2 = 49.04:  log likelihood = -2415.44
## AIC=4836.87   AICc=4836.9   BIC=4850.6
##
## Training set error measures:
##                      ME     RMSE      MAE       MPE    MAPE     MASE
## Training set 0.02095817 6.988021 4.753144 -1.179891 8.34541 0.965802
##                      ACF1
## Training set -0.007850169
```

The results from auto.arima() indicate that the best-fitting model is ARIMA(1,1,1).

Model Diagnostic [6]

Hide

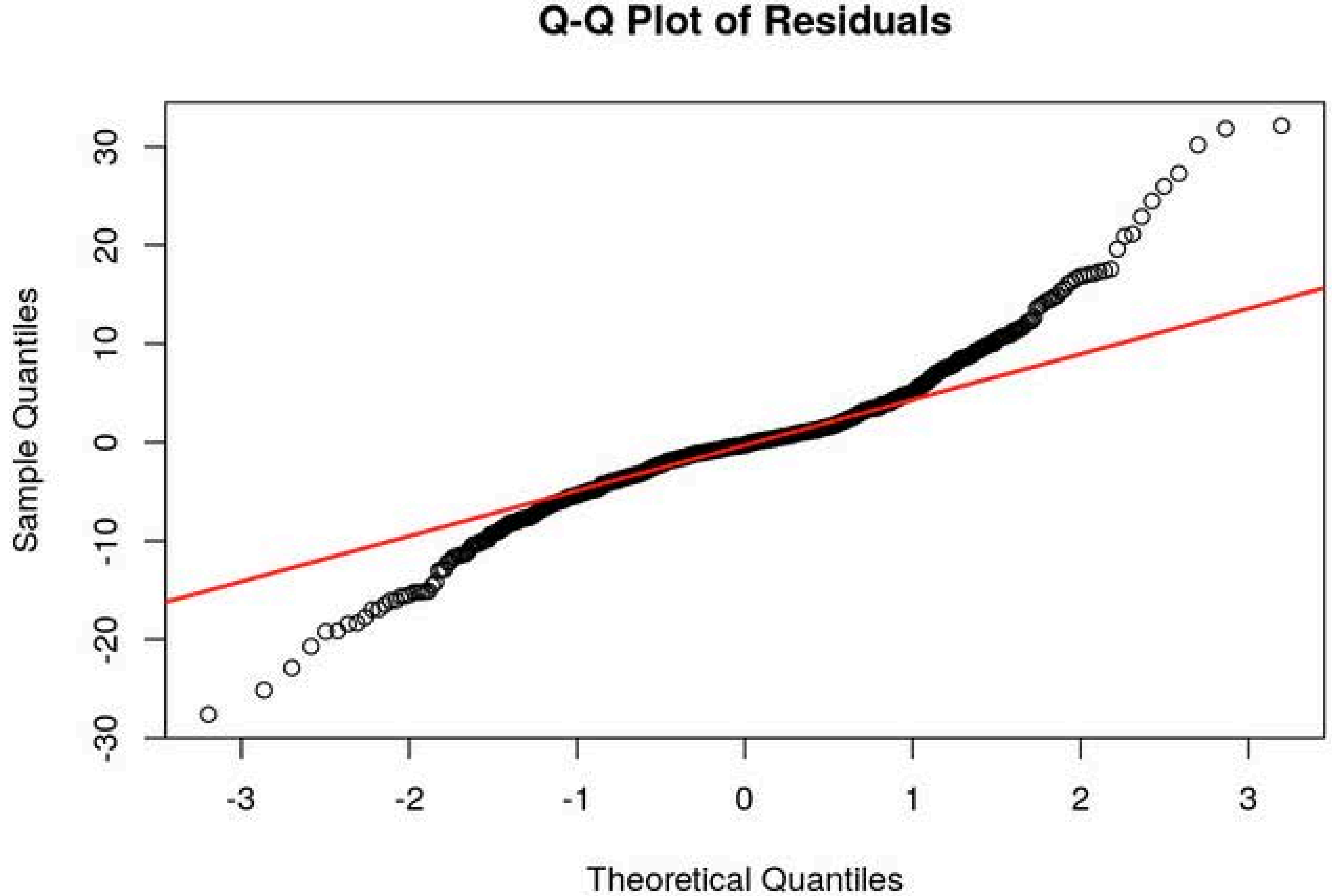
```
checkresiduals(arima_model)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,1)
## Q* = 15.256, df = 8, p-value = 0.05435
##
## Model df: 2.   Total lags used: 10
```

Hide

```
qqnorm(residuals(arima_model), main = "Q-Q Plot of Residuals")
qqline(residuals(arima_model), col = "red", lwd = 2)
```



Hide

```
shapiro.test(residuals(arima_model))
```

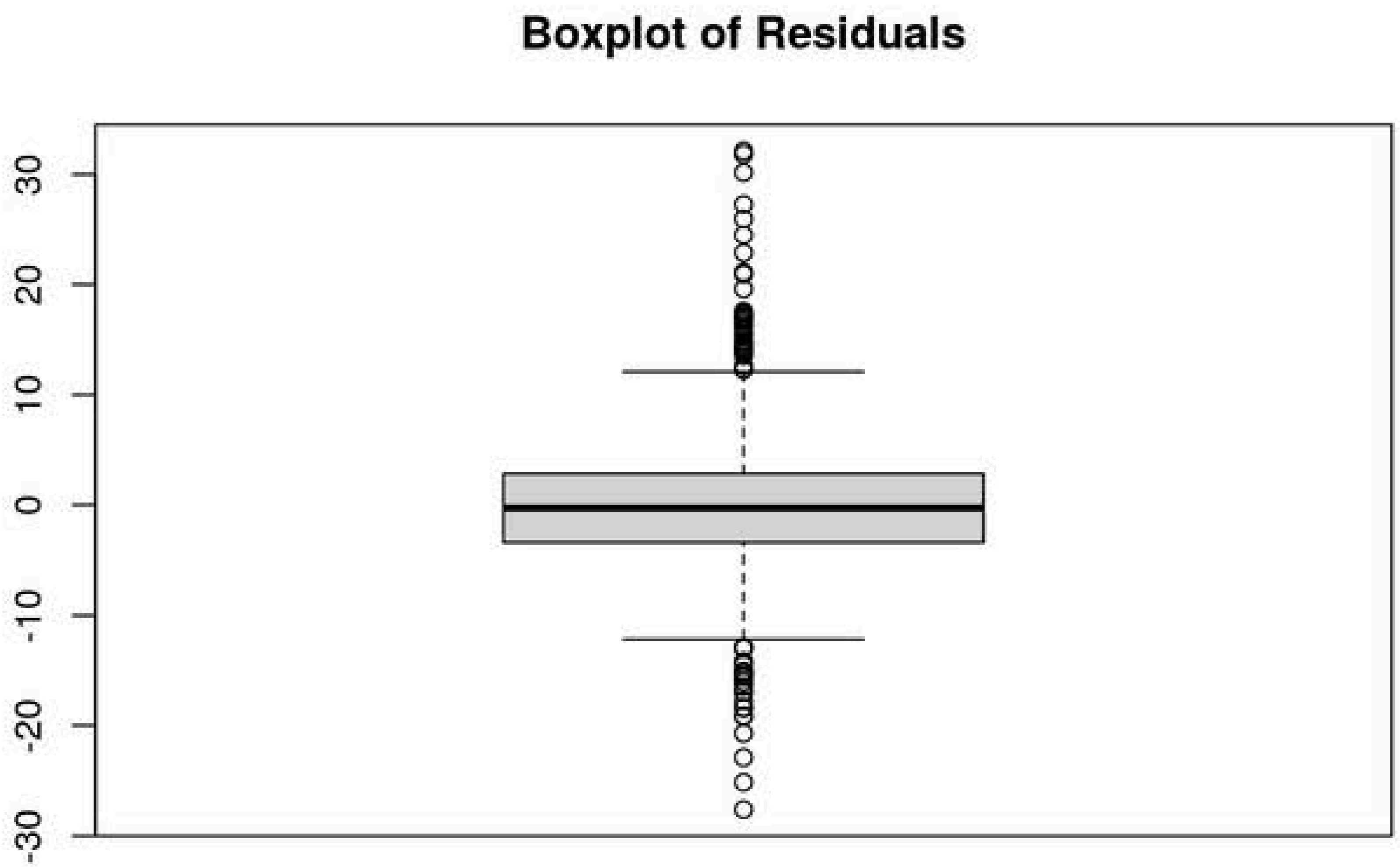
```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(arima_model)
## W = 0.94285, p-value = 5.165e-16
```

The Q-Q plot of residuals shows deviations at both tails, particularly at the extreme ends, suggesting the presence of heavy tails or outliers in the residuals. This indicates there may be some extreme values that do not fit a normal distribution. The p-value of Shapiro-Wilk test is extremely small, meaning we reject the null hypothesis of normality. This confirms that the residuals are not normally distributed.

Check for outliers

Hide

```
boxplot(residuals(arima_model), main = "Boxplot of Residuals")
```



The boxplot shows numerous outliers on both the upper and lower tails, which explains the deviation from normality. These outliers suggest occasional extreme values in the residuals, which might be due to unmodeled structure, probably daily effects.

SARIMA Model

Given the strong daily seasonality observed in the time series, we also fit a SARIMA$(p,d,q)(P,D,Q)_s$ model, which extends ARIMA by incorporating seasonal components. The SARIMA model is defined as [5]:

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

where:

- s is the seasonal period (in this case, $s = 144$, representing 24 hours of 10-minute intervals).
- D is the order of seasonal differencing.
- $\Phi(B^s) = 1 - \Phi_1 B^s - \Phi_2 B^{2s} - \cdots - \Phi_P B^{Ps}$ represents the seasonal autoregressive (SAR) polynomial of order P.
- $\Theta(B^s) = 1 + \Theta_1 B^s + \Theta_2 B^{2s} + \cdots + \Theta_Q B^{Qs}$ represents the seasonal moving average (SMA) polynomial of order Q.
- The remaining terms $(\phi(B), \theta(B), d)$ are the same as in the standard ARIMA model.

Fit Model

Hide

```
sarima_model <- auto.arima(noise$Leq_dBA, D = 1,seasonal = TRUE, trace = TRUE)
```

```
##
##  Fitting models using approximations to speed things up...
##
##  ARIMA(2,0,2) with non-zero mean : 4840.386
##  ARIMA(0,0,0) with non-zero mean : 5742.496
##  ARIMA(1,0,0) with non-zero mean : 4921.39
##  ARIMA(0,0,1) with non-zero mean : 5324.83
##  ARIMA(0,0,0) with zero mean     : 7976.732
##  ARIMA(1,0,2) with non-zero mean : 4839.851
##  ARIMA(0,0,2) with non-zero mean : 5169.173
##  ARIMA(1,0,1) with non-zero mean : 4847.601
##  ARIMA(1,0,3) with non-zero mean : 4840.316
##  ARIMA(0,0,3) with non-zero mean : 5052.846
##  ARIMA(2,0,1) with non-zero mean : 4838.725
##  ARIMA(2,0,0) with non-zero mean : 4872.838
##  ARIMA(3,0,1) with non-zero mean : 4841.069
##  ARIMA(3,0,0) with non-zero mean : 4857.62
##  ARIMA(3,0,2) with non-zero mean : 4843.096
##  ARIMA(2,0,1) with zero mean     : Inf
##
##  Now re-fitting the best model(s) without approximations...
##
##  ARIMA(2,0,1) with non-zero mean : 4838.649
##
##  Best model: ARIMA(2,0,1) with non-zero mean
```

Hide

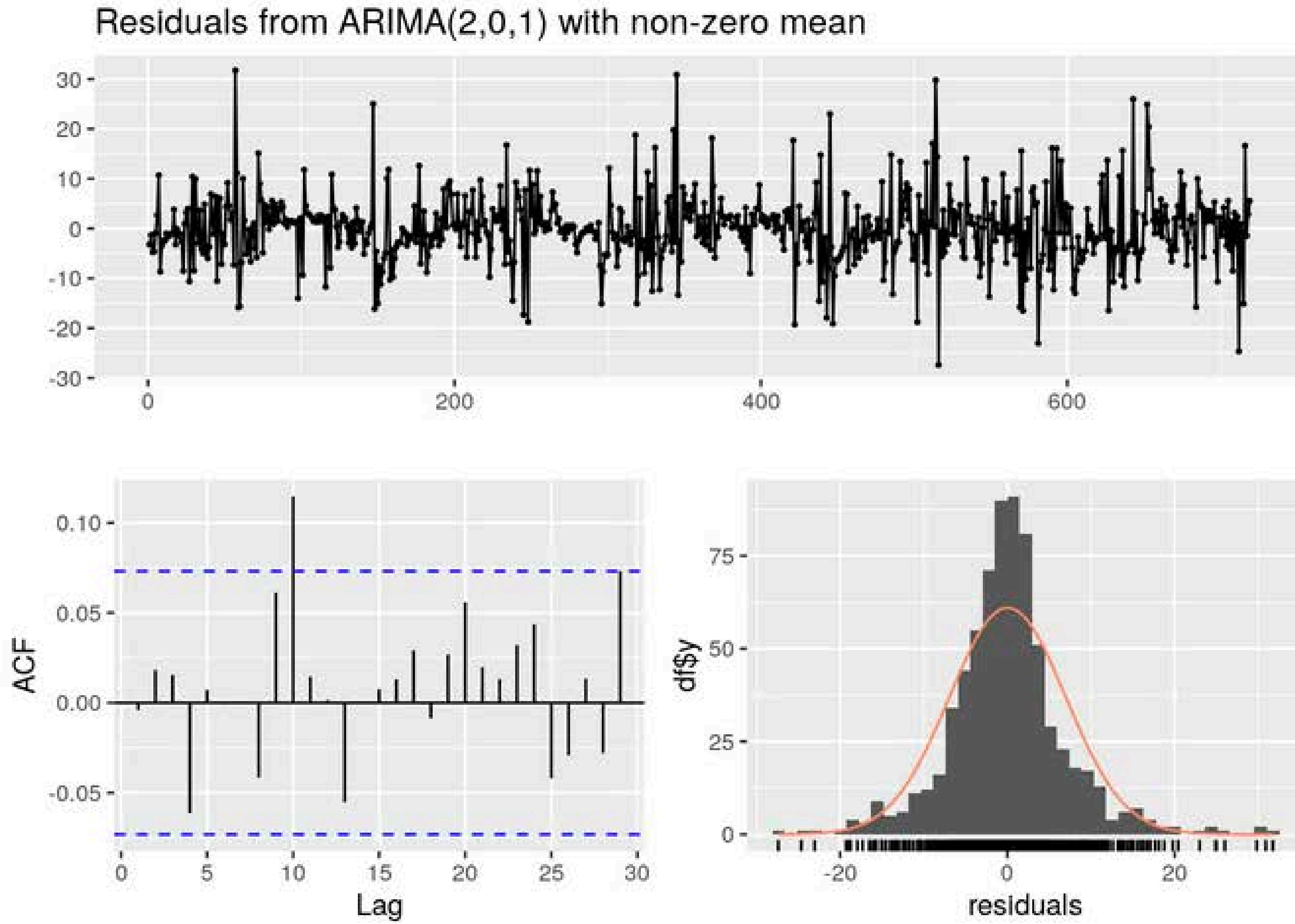
```
summary(sarima_model)
```

```
## Series: noise$Leq_dBA
## ARIMA(2,0,1) with non-zero mean
##
## Coefficients:
##          ar1      ar2      ma1     mean
##       1.2446  -0.2663  -0.7090  60.4130
## s.e.  0.0709   0.0660   0.0546   3.3486
##
## sigma^2 = 48.49:  log likelihood = -2414.28
## AIC=4838.56   AICc=4838.65   BIC=4861.45
##
## Training set error measures:
##                      ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 0.02360858 6.943963 4.882795 -1.438566 8.637658 0.9921461
##                      ACF1
## Training set -0.004111738
```

Model Diagnostic

Hide

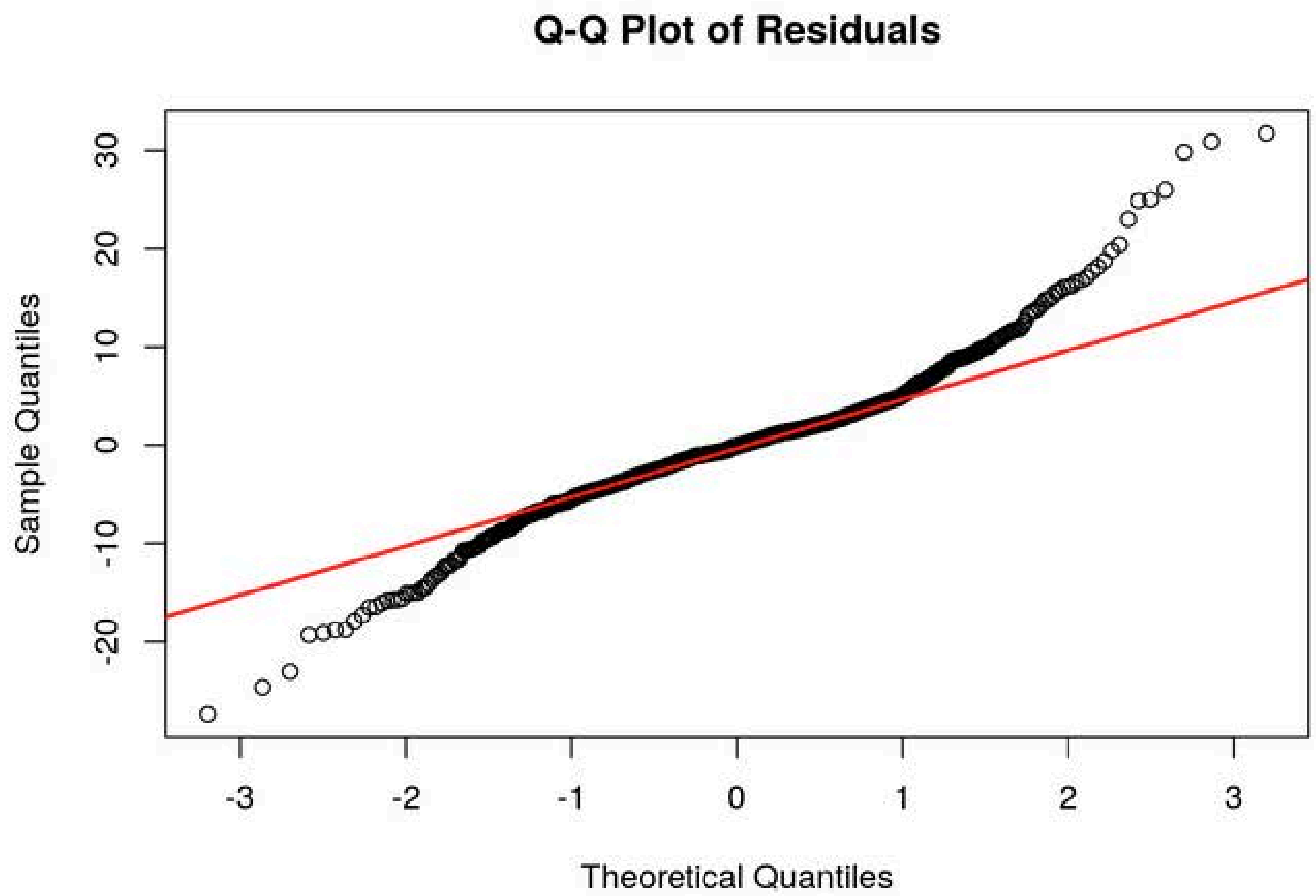
```
checkresiduals(sarima_model)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,0,1) with non-zero mean
## Q* = 16.788, df = 7, p-value = 0.01882
##
## Model df: 3.   Total lags used: 10
```

Hide

```
qqnorm(residuals(sarima_model), main = "Q-Q Plot of Residuals")
qqline(residuals(sarima_model), col = "red", lwd = 2)
```



Hide

```
shapiro.test(residuals(sarima_model))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(sarima_model)
## W = 0.95465, p-value = 4.392e-14
```

The best-selected model is SARIMA(2,0,1). The residual diagnostics suggest reasonable model adequacy. The ACF plot of residuals shows no significant autocorrelation at most lags, suggesting that the model has captured most of the dependencies in the data. The histogram and Q-Q plot of residuals show slight deviations from normality, particularly in the tails, indicating the presence of some extreme values or outliers. However, given that the residuals are mostly centered around zero and appear to exhibit white noise behavior, the model is likely sufficient for forecasting.

Conclusion

In this project, I conducted a comprehensive time series analysis of environmental noise levels using SARIMA modeling. The analysis began with exploratory data visualization, where clear daily patterns were observed, prompting the need for seasonal modeling. Stationarity testing using the Augmented Dickey-Fuller test confirmed that first-order seasonal differencing was necessary to stabilize the mean. Further, autocorrelation and partial autocorrelation analyses guided the selection of appropriate autoregressive and moving average terms.

I explored both ARIMA and SARIMA models, with auto.arima() assisting in model selection based on the Akaike Information Criterion. The best-selected model was SARIMA(2,0,1)[144], which effectively captured the seasonal and stochastic dependencies in the data. Residual diagnostics, including ACF of residuals and normality tests, suggested that the model sufficiently removed autocorrelation and provided a reasonable fit, though slight deviations from normality were noted.

Overall, this study demonstrates the effectiveness of SARIMA modeling in capturing complex temporal patterns in environmental noise data.

Reference

[1] Wikipedia https://en.wikipedia.org/wiki/Sound_level_meter#LAT_or_Leq:_Equivalent_continuous_sound_level

[2] R Documentaion https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/stl

[3] R Documentation https://www.rdocumentation.org/packages/forecast/versions/8.21/topics/auto.arima

[4] Hyndman, R. J., & Khandakar, Y. (2008). Automatic Time Series Forecasting: The forecast Package for R. Journal of Statistical Software, 27(3), 1–22. https://doi.org/10.18637/jss.v027.i03

[5] Class material https://ionides.github.io/531w25/

[6] R Documentation https://www.rdocumentation.org/packages/forecast/versions/8.23.0/topics/checkresiduals