The ARIMA-based anomaly detector fits an ARIMA model to the time series and flags large residuals as anomalies. In this walkthrough we will:

* Load a synthetic anomaly dataset and visualize it
* Configure and run the ARIMA detector (hanr\_arima)
* Inspect detections, evaluate against ground truth, and plot residuals and thresholds

# Install Harbinger (if needed)  
#install.packages("harbinger")

# Load required packages  
library(daltoolbox)  
library(harbinger)

# Load example anomaly datasets  
data(examples\_anomalies)

# Select a simple anomaly dataset  
dataset <- examples\_anomalies$simple  
head(dataset)

## serie event  
## 1 1.0000000 FALSE  
## 2 0.9689124 FALSE  
## 3 0.8775826 FALSE  
## 4 0.7316889 FALSE  
## 5 0.5403023 FALSE  
## 6 0.3153224 FALSE

# Plot the raw time series  
har\_plot(harbinger(), dataset$serie)



# Configure ARIMA-based anomaly detector  
model <- hanr\_arima()

# Fit the detector (estimates ARIMA order and caches parameters)  
model <- fit(model, dataset$serie)

# Run detection to compute residual magnitudes and flags  
detection <- detect(model, dataset$serie)

# Show detected anomaly indices  
print(detection |> dplyr::filter(event == TRUE))

## idx event type  
## 1 50 TRUE anomaly

# Evaluate detections against labeled events  
evaluation <- evaluate(model, detection$event, dataset$event)  
print(evaluation$confMatrix)

## event   
## detection TRUE FALSE  
## TRUE 1 0   
## FALSE 0 100

# Plot detections vs. ground truth  
har\_plot(model, dataset$serie, detection, dataset$event)



# Plot the residual magnitude and the decision thresholds  
har\_plot(model, attr(detection, "res"), detection, dataset$event, yline = attr(detection, "threshold"))

