k-means clustering anomaly detection: This approach applies k-means clustering to either points (seq = 1) or sliding-window subsequences (seq > 1). Observations (or windows) that lie far from their nearest centroid are flagged as anomalies or discords, respectively. The implementation wraps stats::kmeans for clustering and uses harutils() to summarize distances and set thresholds.

Objectives: This Rmd demonstrates clustering-based anomaly detection using k-means via hanct\_kmeans(k). The model clusters subsequences and flags anomalies based on distance to cluster centroids. Steps: load packages/data, visualize, define k-means model, fit, detect, evaluate, and plot series and residuals.

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

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

# Load example datasets bundled with harbinger  
data(examples\_anomalies)

# Select a simple synthetic time series with labeled anomalies  
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 time series  
har\_plot(harbinger(), dataset$serie)



# Define k-means anomaly detector (k controls number of clusters)  
 model <- hanct\_kmeans(1)

# Fit the model  
 model <- fit(model, dataset$serie)

# Detect anomalies using k-means distances  
 detection <- detect(model, dataset$serie)

# Show only timestamps flagged as events  
 print(detection |> dplyr::filter(event==TRUE))

## idx event type  
## 1 50 TRUE anomaly

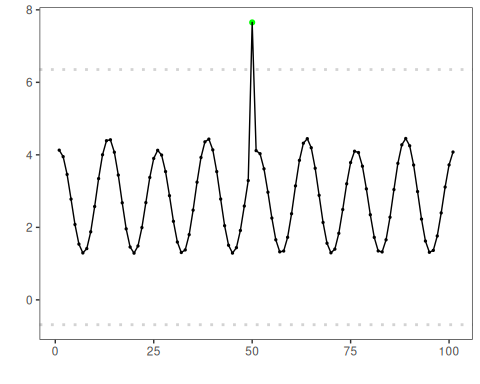
# Evaluate detections against ground-truth labels  
 evaluation <- evaluate(model, detection$event, dataset$event)  
 print(evaluation$confMatrix)

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

# Plot detections over the series  
 har\_plot(model, dataset$serie, detection, dataset$event)



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



References - Ogasawara, E., Salles, R., Porto, F., Pacitti, E. Event Detection in Time Series. 1st ed. Cham: Springer Nature Switzerland, 2025. <doi:10.1007/978-3-031-75941-3>