# Overview

This Rmd demonstrates discord (rare pattern) discovery using k-means via hanct\_kmeans(k). The model clusters subsequences and identifies discords that are far from any cluster centroid. 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")

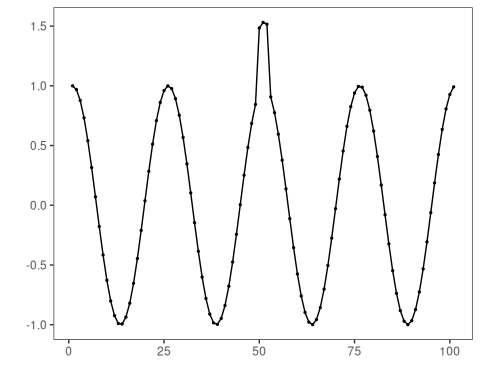
# Load required packages  
library(daltoolbox)  
library(harbinger)

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

# Use the sequence time series (labeled)  
dataset <- examples\_anomalies$sequence  
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 discord detector (k controls number of clusters)  
 model <- hanct\_kmeans(3)

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

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

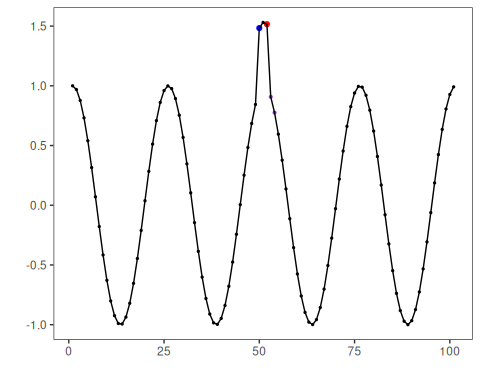
# Show only timestamps flagged as events  
 print(detection |> dplyr::filter(event==TRUE))

## idx event type seq seqlen  
## 1 52 TRUE discord 3 3

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

## event   
## detection TRUE FALSE  
## TRUE 0 1   
## FALSE 1 99

# 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"))

