Binary Segmentation (BinSeg) recursively identifies multiple change points by splitting the series at the strongest change and repeating. In this tutorial we will:

* Load a dataset with change points and visualize it
* Configure and run the BinSeg detector (hcp\_binseg)
* Inspect detections and evaluate against ground truth
* Plot the detections on the series

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

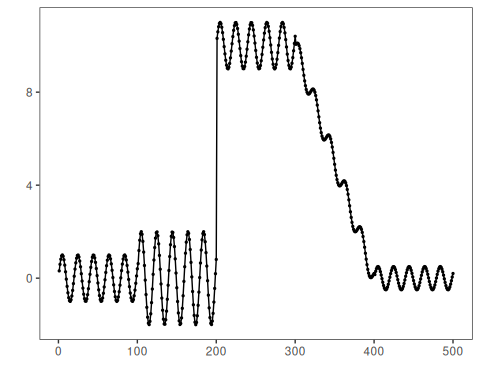
# Load required packages  
library(daltoolbox)  
library(harbinger)

# Load example change-point datasets  
data(examples\_changepoints)

# Select a dataset ("complex" contains multiple regimes)  
dataset <- examples\_changepoints$complex  
head(dataset)

## serie event  
## 1 0.3129618 FALSE  
## 2 0.5944808 FALSE  
## 3 0.8162731 FALSE  
## 4 0.9560557 FALSE  
## 5 0.9997847 FALSE  
## 6 0.9430667 FALSE

# Plot the time series to visualize regimes  
har\_plot(harbinger(), dataset$serie)



# Configure BinSeg; Q is the max number of change points to search  
model <- hcp\_binseg(Q = 10)

# Fit the detector (keeps parameters on object)  
model <- fit(model, dataset$serie)

# Run detection over the series  
detection <- detect(model, dataset$serie)

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

## idx event type  
## 1 101 TRUE changepoint  
## 2 200 TRUE changepoint  
## 3 312 TRUE changepoint  
## 4 327 TRUE changepoint  
## 5 349 TRUE changepoint  
## 6 368 TRUE changepoint  
## 7 389 TRUE changepoint

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

## event   
## detection TRUE FALSE  
## TRUE 1 6   
## FALSE 3 490

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

