```
# Load required libraries
library(readr)
library(lubridate)
library(ggplot2)

# Load data
rain_original <- read_csv(here::here("input_data", "rain.csv"))
flow_original <- read_csv(here::here("input_data", "flow.csv"))

# Prepare data in units required to run the functions
multiple <- 24 # to convert from mm/h to the original units of the timeseries (mm/day)
rain <- rain_original[, 7] / multiple # mm/h
flow <- flow_original[, 7] / multiple # mm/h
time <- as.POSIXct(paste(rain_original[, 1], rain_original[, 2], rain_original[, 3],
rain_original[, 4], rain_original[, 5], rain_original[, 6], sep = "-"), format = "%Y-%m-
%d-%H-%M-%S")

# Identifying events
rain_min <- 0.02 # if we consider Rmin=0.1 mm at hourly scale and we follow the approach
in Text S1: Rmin=0.1*24^(-0.5)=0.02 at daily scale
max_window <- 100 # this means we expect the catchment response time (Giani et al., 2021)
to be maximum 49 time steps (in this case 49 days)
EVENT_IDENTIFICATION_DMCA <- function(rain, flow, time, rain_min, max_window) {
  # Your implementation of the EVENT_IDENTIFICATION_DMCA function
  # Return a list with the following elements:
  # BEGINNING_RAIN, END_RAIN, BEGINNING_FLOW, END_FLOW
}
result <- EVENT_IDENTIFICATION_DMCA(rain, flow, time, rain_min, max_window)
BEGINNING_RAIN <- result$BEGINNING_RAIN
END_RAIN <- result$END_RAIN
BEGINNING_FLOW <- result$BEGINNING_FLOW
END_FLOW <- result$END_FLOW

# Baseflow curve
BASEFLOW_CURVE <- function(BEGINNING_FLOW, END_FLOW, flow, time) {
  # Your implementation of the BASEFLOW_CURVE function
  # Return the baseflow in mm/h
}
baseflow <- BASEFLOW_CURVE(BEGINNING_FLOW, END_FLOW, flow, time)
baseflow_original <- baseflow * multiple # mm/day

# Events analysis
flag <- 0 # the output we provide is total streamflow
EVENT_ANALYSIS <- function(BEGINNING_RAIN, END_RAIN, BEGINNING_FLOW, END_FLOW, rain, flow,
time, flag, multiple) {
  # Your implementation of the EVENT_ANALYSIS function
  # Return DURATION_RAIN, VOLUME_RAIN, DURATION_RUNOFF, VOLUME_RUNOFF, RUNOFF_RATIO
}
result <- EVENT_ANALYSIS(BEGINNING_RAIN, END_RAIN, BEGINNING_FLOW, END_FLOW, rain, flow,
time, flag, multiple)
DURATION_RAIN <- result$DURATION_RAIN
VOLUME_RAIN <- result$VOLUME_RAIN
DURATION_RUNOFF <- result$DURATION_RUNOFF
VOLUME_RUNOFF <- result$VOLUME_RUNOFF
RUNOFF_RATIO <- result$RUNOFF_RATIO

# Plotting timeseries and identified events
index_start_rain <- sapply(BEGINNING_RAIN, function(x) which.min(abs(time - x)))
index_finish_rain <- sapply(END_RAIN, function(x) which.min(abs(time - x)))
index_start_flow <- sapply(BEGINNING_FLOW, function(x) which.min(abs(time - x)))
index_finish_flow <- sapply(END_FLOW, function(x) which.min(abs(time - x)))

ggplot() +
  geom_line(aes(x = time, y = rain_original[, 7]), color = "magenta", size = 2) +
```

```
  geom_point(aes(x = time[index_start_rain - 1], y = rain_original[index_start_rain - 1,
7]), color = "magenta", size = 10) +
  geom_point(aes(x = time[index_finish_rain + 1], y = rain_original[index_finish_rain + 1,
7]), color = "magenta", size = 5, shape = 8) +
  geom_line(aes(x = time, y = flow_original[, 7]), color = "blue", size = 2) +
  geom_line(aes(x = time, y = baseflow_original), color = "black", size = 2) +
  geom_point(aes(x = time[index_start_flow], y = flow_original[index_start_flow, 7]),
color = "blue", size = 10) +
  geom_point(aes(x = time[index_finish_flow], y = flow_original[index_finish_flow, 7]),
color = "blue", size = 5, shape = 8) +
  labs(x = "Time [dd/mm/yy]", y = "Rainfall [mm/day]", y2 = "Streamflow [mm/day]") +
  scale_y_continuous(name = "Rainfall [mm/day]", sec.axis = sec_axis(~., name =
"Streamflow [mm/day]")) +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1))
```