

# Camel series shiny app presentation

Daniel Černý

2025-01-29

## Introduction

This presentation demonstrates the functionality of shiny app designed for displaying and basic analysis of hydrological data from Camel series. The app allows user to explore variables for specific catchments over selected date range.

## App Features

- Select a catchment and variable (`R`, `Q`, or `Tmax`).
- Specify a date range for visualization with slide bar.
- View plot of the selected variable over specified range.
- Calculate descriptive statistics for selected variable over specified range (e.g., min, max, mean for `Q` and `Tmax`, and min, max, sum for `R`).

## Data Preparation

Five catchments were chosen as example. The data is stored in text files and loaded into R. The following code prepares the data:

```
library(shiny)
library(ggplot2)
setwd("D:/testcamel/")
povodi_list <- list()
for (i in 1:5) {
  Qcamel <- read.table(paste0("./vyberdata/", i, "_q.txt"), header = FALSE)
  Rcamel <- read.table(paste0("./vyberdata/", i, "_r.txt"), header = FALSE, skip = 4)
  cur_pov <- data.frame(
    Date = as.Date(paste(Rcamel$V1, Rcamel$V2, Rcamel$V3, sep = "-")),
    Q = (Qcamel$V5 * 0.0283168466),
    R = Rcamel$V6,
    Tmax = Rcamel$V9
  )
  povodi_list[[i]] <- cur_pov
  rm(cur_pov, Qcamel, Rcamel)
}
names(povodi_list) <- 1:5
```

## UI settings

The following codes include app UI and server settings.

```
ui <- fluidPage(
  titlePanel("Camel Series"),
  tabsetPanel(
```

```

tabPanel(
  "Stats",
  fluidRow(
    column(3, selectInput("povodi", "Catchment number:", choices = NULL)),
    column(3, selectInput("variable", "Variable:", choices = c("R", "Tmax", "Q"))),
    column(6, sliderInput("daterange", "Time range:",
                          min = as.Date("2000-01-01"),
                          max = as.Date("2000-12-31"),
                          value = c(as.Date("2000-01-01"), as.Date("2000-12-31"))))
  ),
  fluidRow(column(12, plotOutput("plot"))),
  fluidRow(column(4, actionButton("calc_stats", "Count")),
            column(8, verbatimTextOutput("stats")))
)
)
)

```

## Server settings

```

server <- function(input, output, session) {
  observe({ updateSelectInput(session, "povodi", choices = names(povodi_list)) })
  observeEvent(input$povodi, {
    req(input$povodi)
    data <- povodi_list[[input$povodi]]
    updateSliderInput(session, "daterange",
                      min = min(data$Date),
                      max = max(data$Date),
                      value = range(data$Date))
  })

  selected_data <- reactive({
    req(input$povodi, input$daterange)
    povodi_list[[input$povodi]] %>%
      filter(Date >= input$daterange[1], Date <= input$daterange[2])
  })

  stats_result <- reactiveVal(NULL)

  observeEvent(input$calc_stats, {
    data <- selected_data()
    req(nrow(data) > 0)
    variable_data <- data[[input$variable]]
    if (input$variable == "R") {
      stats_result(paste("R:\nMin:", min(variable_data),
                        "\nMax:", max(variable_data),
                        "\nSum:", sum(variable_data)))
    } else {
      stats_result(paste(input$variable, "\nMin:", min(variable_data),
                        "\nMax:", max(variable_data),
                        "\nMean:", mean(variable_data)))
    }
  })
}

```

```

output$stats <- renderText({ stats_result() })
output$plot <- renderPlot({
  req(selected_data())
  ggplot(selected_data(), aes(Date, !!sym(input$variable))) +
    geom_line() + labs(title = paste("Graph of", input$variable, "for", input$povodi))
})
}
shinyApp(ui = ui, server = server)

```

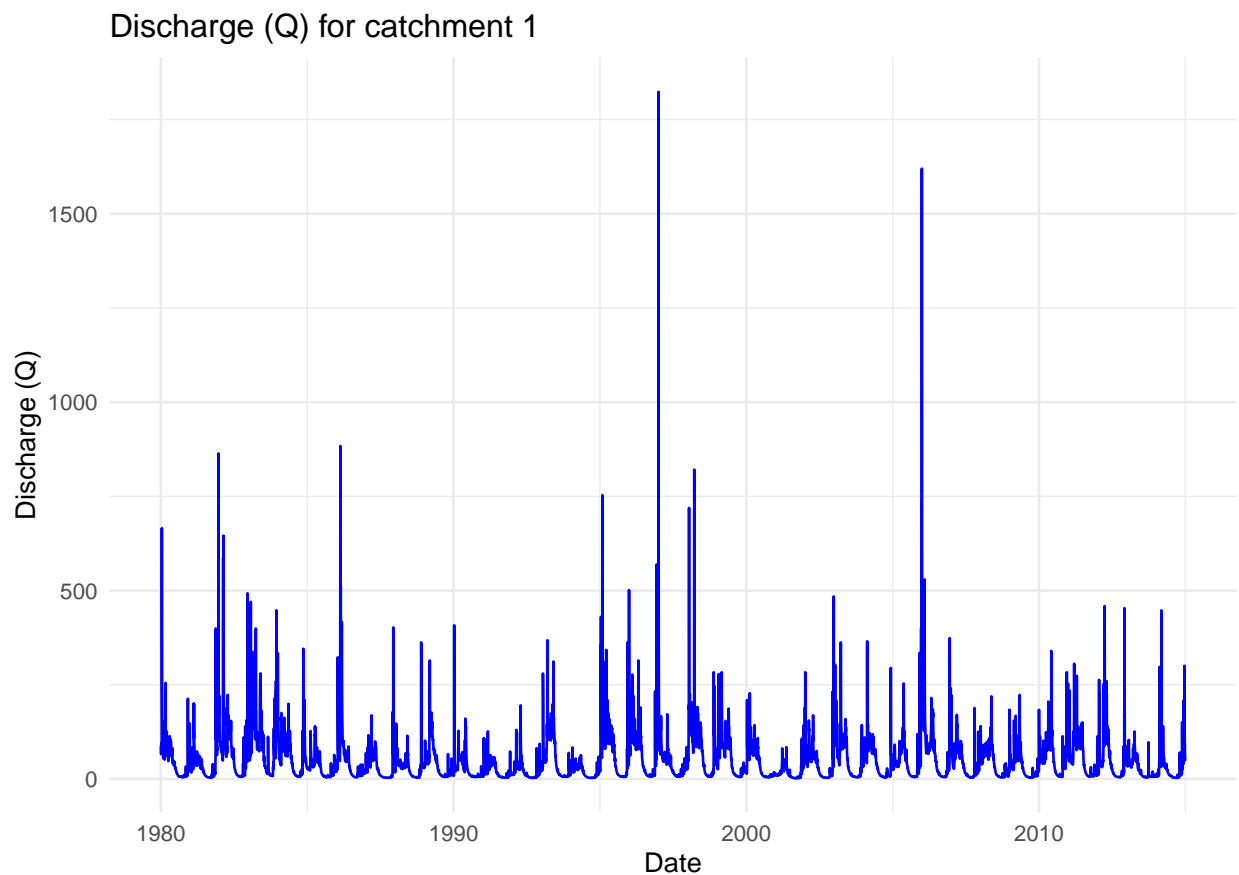
## Plot Example

The following plot shows an example of the time series data for a chosen catchment and variable:

```

# Example: Basin 1, Variable "Q"
data <- povodi_list[[1]]
ggplot(data, aes(x = Date, y = Q)) +
  geom_line(color = "blue") +
  labs(title = "Discharge (Q) for catchment 1", x = "Date", y = "Discharge (Q)") +
  theme_minimal()

```



## Statistics

Below are calculated statistics (min, max,sum) for selected variable in the dataset:

```
# Example: Basin 1, Variable "R"
selected_basin <- povodi_list[[1]]
variable <- "R"

min_val <- min(selected_basin[[variable]], na.rm = TRUE)
max_val <- max(selected_basin[[variable]], na.rm = TRUE)
sum_val <- sum(selected_basin[[variable]], na.rm = TRUE)

cat(
  "Statistics for Basin 1 (Rainfall):\n",
  "Min:", min_val, "\n",
  "Max:", max_val, "\n",
  "Sum:", sum_val, "\n"
)

## Statistics for Basin 1 (Rainfall):
##  Min: 0
##  Max: 151.12
##  Sum: 46093.87
```