```
myApp <- function(...){
library(shiny)
library(shinydashboard)
library(shinydashboardPlus)
library(echarts4r)
dat1 <- read.csv("Harbin.csv")
dat1$DATE <- as.Date(dat1$DATE)
ui <- dashboardPage(
  title = "积温计算器",
  dashboardHeader(title = "积温计算器"),
  dashboardSidebar(),
  dashboardBody(
    tags$style("body { background-color: ghostwhite}"),
    box(
      "请按照条件输入",
      id = "inputbox",
      collapsible = TRUE,
      closable = TRUE,
      dateInput("start.date", "开始播种日期", value = "2012-09-18"),
      dateInput("end.date", "收获日期", value = "2013-09-17"),
      numericInput("obs", "作物零点温度°C:", 10, min = 0, max = 20),
      actionButton("done","确定")
    ),
    br(),
    box(
      "Box body",
      id = "mybox",
      collapsible = TRUE,
      closable = TRUE,
      textOutput("daysmax"),
      textOutput("continous_day"),
      textOutput("accumulated_t"),
      textOutput("effective_a_t"),
      echarts4rOutput("plot")
    )
  )
)

server <- function(input, output, session) {


  observeEvent(input$done, {
    date_range <- reactive(subset(dat1,
                                  DATE>=input$start.date&
                                    DATE<=input$end.date))
    date_range <-    isolate(date_range())
    plant_zero <- isolate(input$obs)
    days <- which(date_range$TAVG>=plant_zero)
    day_count <- length(days )
    output$daysmax <- renderText(
      paste("大于生物学零点的总数为:",day_count,"天")
    )
    max_day <- max_day_rec <- 1
    for(i in 2:day_count)
    {
      if(days[i]==days[(i-1)]+1)
      {
          max_day <- max_day+1
          start_p <- i-max_day+1
      }
      else if(max_day!=1)
      {
          if(max_day>max_day_rec)
          {
              max_day_rec<- max_day
              start_p_rec <-    start_p
              max_day <- 0
          }
      }
    }
    # output2 maximum consecutive days
    final_day <- max(max_day,max_day_rec)
    output$continous_day <- renderText(
      paste("大于生物学零点的最长连续天数为:",final_day,"天")
    )
    start_point <- ifelse(max_day>max_day_rec,
                                          start_p,
                                          start_p_rec)
    start_point_date <- date_range$DATE[days[start_point]]
    end_point_date <- date_range$DATE[days[(start_point+final_day-1)]]
    ac_temp <- sum(date_range$TAVG[
      date_range$DATE>=start_point_date&date_range$
          DATE<=end_point_date])
    output$accumulated_t <- renderText(
      paste("积温:",ac_temp,"°C")
    )
    effective_a_t <- ac_temp-final_day*plant_zero
    output$effective_a_t <- renderText(
      paste("有效积温:", effective_a_t,"°C")
    )
    output$plot <-    renderEcharts4r({
      req(!input$mybox$collapsed)
      yearMonthDate <- htmlwidgets::JS('function (value) {
  var monthShortNames = ["一月", "二月", "三月", "四月", "五月", "六月",
    "七月", "八月", "九月", "十月", "十一月", "十二月"
  ];
  var d = new Date(value);
  var datestring =    monthShortNames[d.getMonth()] + "    "
    var datestring =    d.getFullYear() + "    " + monthShortNames[d.getMonth()]    + "    "
  return datestring
}')

      avg <- list(
        type = "average",
        name = "AVG"
      )
      date_range|>
        e_charts(DATE)|>
        e_line(TAVG)|>
        e_mark_area('TAVG',data = list(
          list(xAxis = start_point_date,yAxis=plant_zero),
          list(xAxis = end_point_date,yAxis=max(dat1$TAVG),
               title = "最大连续积温区域")
        ),
        itemStyle = list(color = "lightgreen"))|>
        e_mark_line(data = list(yAxis = plant_zero),
                    title = "生物学下限温度",
                    itemStyle = list(color = "red")) |>
        e_format_y_axis(suffix = "°C")|>
        e_axis_labels(
          x = "日期",
          y = "日均气温"
        )|>
        e_x_axis(
          DATE,
          axisPointer = list(show = TRUE),
          axisLabel = list(
            rotate = 45,
            formatter = yearMonthDate
          ))|>
        e_tooltip(trigger = "axis")|>
        e_datazoom(
          type = "slider",
          toolbox = FALSE,
          bottom = -5
        )
    })
    output$box_state <- renderText({
      state <- if (input$mybox$collapsed) "collapsed" else "uncollapsed"
      paste("My box is", state)
    })
  })
}
shinyApp(ui, server)
}
```