

Global

```
# Load the following packages.
for (package in c('shiny','dplyr','DT','data.table','datasets','shinydashboard','ggplot2','tidyr',
  'lubridate', 'leaflet.extras', 'tidyverse', 'shinyFiles')) {
  if (!require(package, character.only=T, quietly=T)) {
    install.packages(package, repos="http://cran.us.r-project.org")
    library(package, character.only=T)
  }
}
```

```
library('leaflet')
library('highcharter')
library(leaflet)
library(leaflet.extras)
library(htmltools)
library(sf)
library(dplyr)
library(shinydashboard)
#library(shinysky)
library(DT)
library(lubridate)
library(readr)
library(stringr)
library(tidyverse)
```

```
load(file = "D:\\SHINY\\new\\igik_app_pierwsza wersja\\dobowe_klimat_k_d.RData")
load(file = 'D:/SHINY/new/igik_app_pierwsza wersja/dobowe_opad.RData')
load(file = 'D:/SHINY/new/igik_app_pierwsza wersja/dobowe_klimat.RData')
load(file = 'D:/SHINY/new/igik_app_pierwsza wersja/dobowe_synop.RData')
```

UI

```
dashboardPage(
  dashboardHeader(title = 'Dane synoptyczne Polski',
    titleWidth = 300),

  dashboardSidebar(
    width = 250,
    tags$head(
      tags$style(HTML(' .sidebar { position:fixed; width:250px;
        white-space:nowrap; overflow:visible; height: 100vh; overflow-y: auto; }
      '))
    ),
    br(),
    selectInput('dataset', 'Select the dataset', choices = c('dobowe klimatyczne', 'dobowe synoptyczne',
'dobowe opadowe')),
    br(),
    shinyFilesButton(id="file1",label="Browse files...",
      multiple=F,title="Please select an appropriate file."),
```

```

hr(),
dateRangeInput('time', 'Data',
  start = min(dobowe_klimat_k_d$time, na.rm = T),
  end = max(dobowe_klimat_k_d$time, na.rm=T)),

selectizeInput('Nazwa_stacji', 'Nazwa stacji',
  multiple = T, choices = sort(unique(dobowe_klimat_k_d$Nazwa_stacji), decreasing = F),
  selected = 'WARSZAWA'),
hr()
),

```

```

dashboardBody( tags$head(
  tags$style(HTML(
    '.main-header .logo {
    font-family: "Georgia", Times, "Times New Roman", serif;
    font-weight: bold;
    font-size: 20px;}'
  ))
),
  tabsetPanel(tabPanel(
    strong('Lokalizacja stacji'),
    br(),
    leafletOutput('mapastacji'),
    hr(),
    valueBoxOutput('sumaopadow')
  )),

```

```

  tabPanel(strong('Wizualizacja danych'),
    br(),
    box(title = 'Parametry:',
      fluidRow(
        # 1
        column(4,
          selectInput(inputId = "y",
            label = "Y-axis:",
            choices = c("Maksymalna_temperatura_dobowa", "Status_pomiaru_TMAX",
              'Minimalna_temperatura_dobowa', 'Status_pomiaru_TMIN',
'Srednia_temperatura_dobowa', 'Status_pomiaru_STD',
              'Temperatura_minimalna_przy_gruncie', 'Status_pomiaru_TMNG',
'Suma_dobowa_opadu', 'Status_pomiaru_SMDB',
              'Rodzaj_opadu', 'Wysokosc_pokrywy_snieznej',
'Status_pomiaru_PKSN', 'time'),
            selected = "Wysokosc_pokrywy_snieznej")),
        column(4,
          selectInput(inputId = "x",
            label = "X-axis:",
            choices = c("Maksymalna_temperatura_dobowa", "Status_pomiaru_TMAX",
              'Minimalna_temperatura_dobowa', 'Status_pomiaru_TMIN',
'Srednia_temperatura_dobowa', 'Status_pomiaru_STD',
              'Temperatura_minimalna_przy_gruncie', 'Status_pomiaru_TMNG',
'Suma_dobowa_opadu', 'Status_pomiaru_SMDB',

```

```

        'Rodzaj_opadu','Wysokosc_pokrywy_snieznej',
'Status_pomiaru_PKSN', 'time'),
        selected = "Minimalna_temperatura_dobowa")),
column(4, selectInput(inputId = "z",
        label = "Color by:",
        choices = colnames(dobowe_klimat_k_d[9:21]),
        selected = ""))),
solidHeader = TRUE, width = NULL, status = 'primary', collapsible = TRUE),

box(title = 'Wykres', solidHeader = TRUE, width = NULL, status = 'warning',collapsible = TRUE,
fluidRow(
column(12, plotOutput('wykres')))),

box(title = 'Korelacja', solidHeader = TRUE, width = NULL, status = 'warning',collapsible = TRUE,
fluidRow(

column(12, plotOutput('correlation')))),

box(title = 'Histogram', solidHeader = TRUE, width = NULL, status = 'warning',collapsible =
TRUE,
fluidRow(
column(12, plotOutput('histogram'))))
),

tabPanel(strong('Dane stacji'),
br(),
downloadButton('downloadfile', 'Download', icon = icon('download')),
hr(),

dataTableOutput('stacjeTable')
),id = 'tabVals')
)

```

Server

```

function(input, output, session, .data) {

napp <- reactiveVal(0)
file_selected <- reactiveVal()

# depending on OS
if(Sys.info()["sysname"]=="Windows"){
volumes = c('home' = '~','C:' = 'C:/','Users' = 'C:/Users/')
} else {
volumes = getVolumes()
}

observe({
shinyFileChoose(input, "file1", roots = volumes, session = session)

```

```

})

datasets <- reactive({
  switch(input$datasets,
    'dobowe synoptyczne' = dobowe_synop,
    'dobowe klimatyczne' = dobowe_klimat,
    'dobowe opadowe' = dobowe_opad_o_d)
})

data <- reactive({
  subset(dobowe_klimat_k_d, dobowe_klimat_k_d$time >= input$time[1] &
    dobowe_klimat_k_d$time <= input$time[2] &
    dobowe_klimat_k_d$Nazwa_stacji %in% input$Nazwa_stacji &
    dobowe_klimat_k_d$Rok %in% input$Rok &
    dobowe_klimat_k_d$Miesiac %in% input$Miesiac &
    dobowe_klimat_k_d$Rodzaj_opadu %in% input$Rodzaj_opadu &
    dobowe_klimat_k_d$Maksymalna_temperatura_dobowa >=
input$Maksymalna_temperatura_dobowa[1] &
    dobowe_klimat_k_d$Maksymalna_temperatura_dobowa <=
input$Maksymalna_temperatura_dobowa[2] &
    dobowe_klimat_k_d$Minimalna_temperatura_dobowa >=
input$Minimalna_temperatura_dobowa[1] &
    dobowe_klimat_k_d$Minimalna_temperatura_dobowa <=
input$Minimalna_temperatura_dobowa[2] &
    dobowe_klimat_k_d$Wysokosc_pokrywy_snieznej %in% input$Wysokosc_pokrywy_snieznej
&
    dobowe_klimat_k_d$X %in% input$X &
    dobowe_klimat_k_d$Y %in% input$Y &
    dobowe_klimat_k_d$Suma_dobowa_opadu >= input$Suma_dobowa_opadu[1] &
    dobowe_klimat_k_d$Suma_dobowa_opadu <= input$Suma_dobowa_opadu[2])
  # dobowe_klimat_k_d$Suma_dobowa_opadu %in% input$Suma_dobowa_opadu)
})

output$sumaopadow <- renderValueBox({
  data_sum <- subset(datasets(), base$Nazwa_stacji==input$Nazwa_stacji &
dobowe_klimat_k_d$time >= input$time[1] &
    dobowe_klimat_k_d$time <= input$time[2])
  valueBox(formatC(round(sum(data_sum$Suma_dobowa_opadu, na.rm=T)),format = "d",
big.mark = ","),
    ,"Roczna suma opadow: ", icon = icon('home'),
    color = 'purple')
})

output$mapastacji <- renderLeaflet({
  data_map <- subset(dobowe_klimat_k_d, dobowe_klimat_k_d$Nazwa_stacji
==input$Nazwa_stacji & dobowe_klimat_k_d$time >= input$time[1] &
    dobowe_klimat_k_d$time <= input$time[2])

  m <- leaflet(data = data_map) %>%

```

```

addTiles() %>%
setView(lng = 19.599316, lat = 52.160, zoom = 5.5) %>%
addCircleMarkers(lng = data_map$X,
  lat = data_map$Y,
  fill=T,
  opacity = 0,
  fillOpacity = 10,
  fillColor = 'orange',
  weight = 1,
  radius = 6,
  label = paste0("Minimalna temperatura dobowa:
",formatC(data_map$Minimalna_temperatura_dobowa,format = "d",big.mark = ","), " ; ",
  'Maksymalna temperatura dobowa:
',data_map$Maksymalna_temperatura_dobowa))
  m
})

output$correlation <- renderPlot({
  data_cor <- subset(dobowe_klimat_k_d, dobowe_klimat_k_d$Nazwa_stacji==input$Nazwa_stacji
& dobowe_klimat_k_d$time >= input$time[1] &
  dobowe_klimat_k_d$time <= input$time[2])
  ggplot(data = data_cor, aes_string(x = input$x, y = input$y, color = input$z)) +
  geom_point()
})

output$histogram <- renderPlot({
  data_hist<- subset(dobowe_klimat_k_d, dobowe_klimat_k_d$Nazwa_stacji == input$Nazwa_stacji
& dobowe_klimat_k_d$time >= input$time[1] & dobowe_klimat_k_d$time <= input$time[2])
  ggplot(data = data_hist, aes_string(x = input$x)) +
  geom_histogram()
})

output$wykres <- renderPlot({
  data_wyk<- subset(dobowe_klimat_k_d, dobowe_klimat_k_d$Nazwa_stacji == input$Nazwa_stacji
& dobowe_klimat_k_d$time >= input$time[1] &
  dobowe_klimat_k_d$time <= input$time[2])
  ggplot(data = data_wyk, aes_string(y = input$y, x=sort(data_wyk$time, decreasing = F))) +
  geom_line()
})

output$avgx <- renderText({
  data_avgx<- subset(dobowe_klimat_k_d, dobowe_klimat_k_d$Nazwa_stacji ==
input$Nazwa_stacji & dobowe_klimat_k_d$time >= input$time[1] &
  dobowe_klimat_k_d$time <= input$time[2])
  paste('Srednia zmiennej: ', input$x, 'stacji: ', input$Nazwa_stacji, 'wynosi: ',
round(mean(x=data_avgx$x, na.rm=T),3)
})

```

```

output$avgy <- renderText({
  data_avgy<- subset(dobowe_klimat_k_d, dobowe_klimat_k_d$Nazwa_stacji ==
input$Nazwa_stacji & dobowe_klimat_k_d$time >= input$time[1] &
  dobowe_klimat_k_d$time <= input$time[2])
  paste('Srednia zmiennej: ', input$y, 'stacji: ', input$Nazwa_stacji, 'wynosi: ',
round(mean(x=data_avgy$y, na.rm=T),3))

})

output$stacjeTable <- renderDataTable({

  datatable(subset(dobowe_klimat_k_d, dobowe_klimat_k_d$Nazwa_stacji == input$Nazwa_stacji
&
  dobowe_klimat_k_d$time >= input$time[1] & dobowe_klimat_k_d$time <=
input$time[2]),
  options = list(paging = TRUE, searching = TRUE, lengthChange = TRUE, bInfo=0,
  bAutoWidth = TRUE,rownames=T, deferRender = TRUE))

}))

# DOWNLOAD -----
output$downloadfile <- downloadHandler(

  filename = function() {
    paste('dane_synoptyczne','.csv', sep='')
  },

  content = function(file) {
    write.csv(as.data.frame(data()), file)
  }

)

```