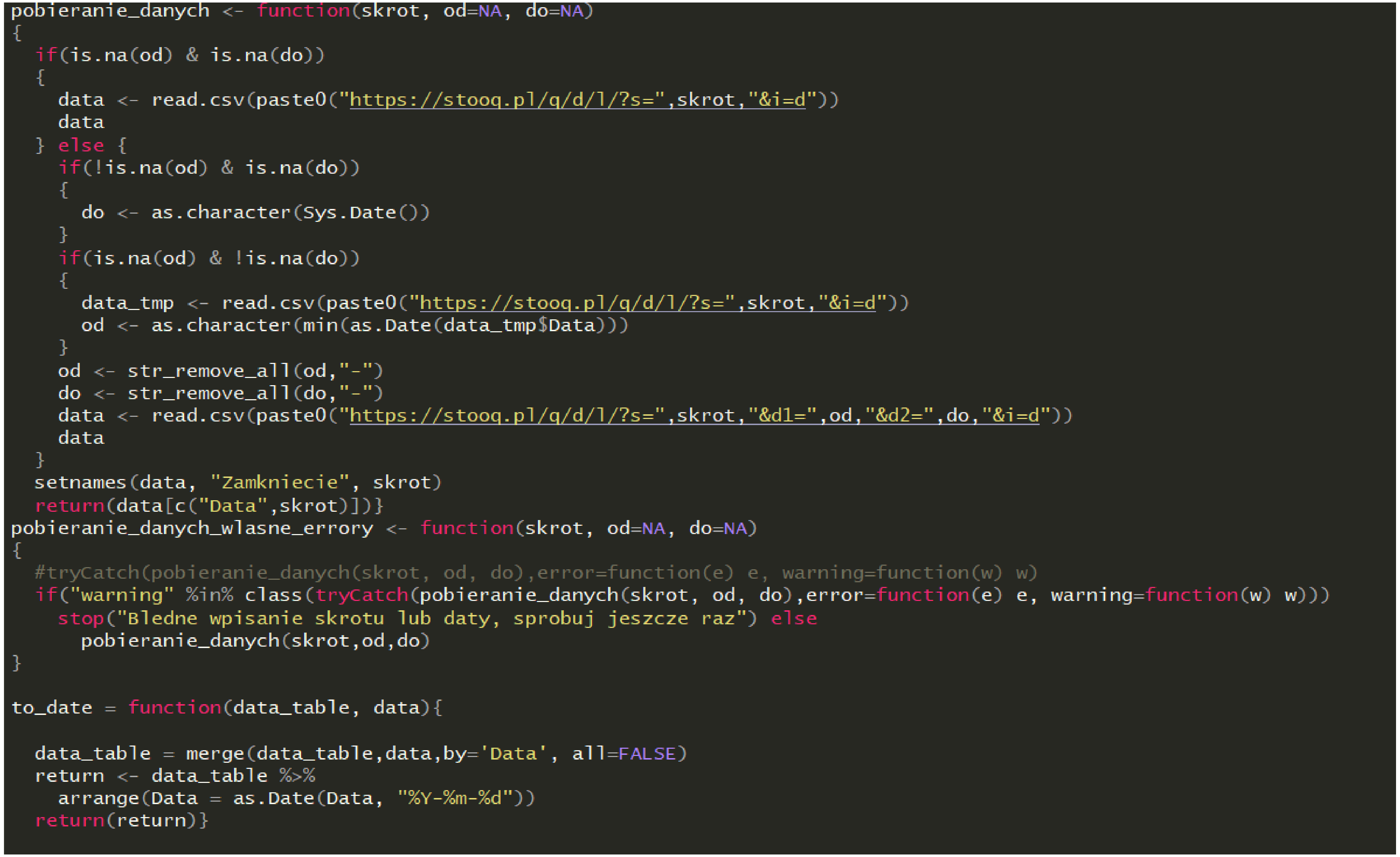
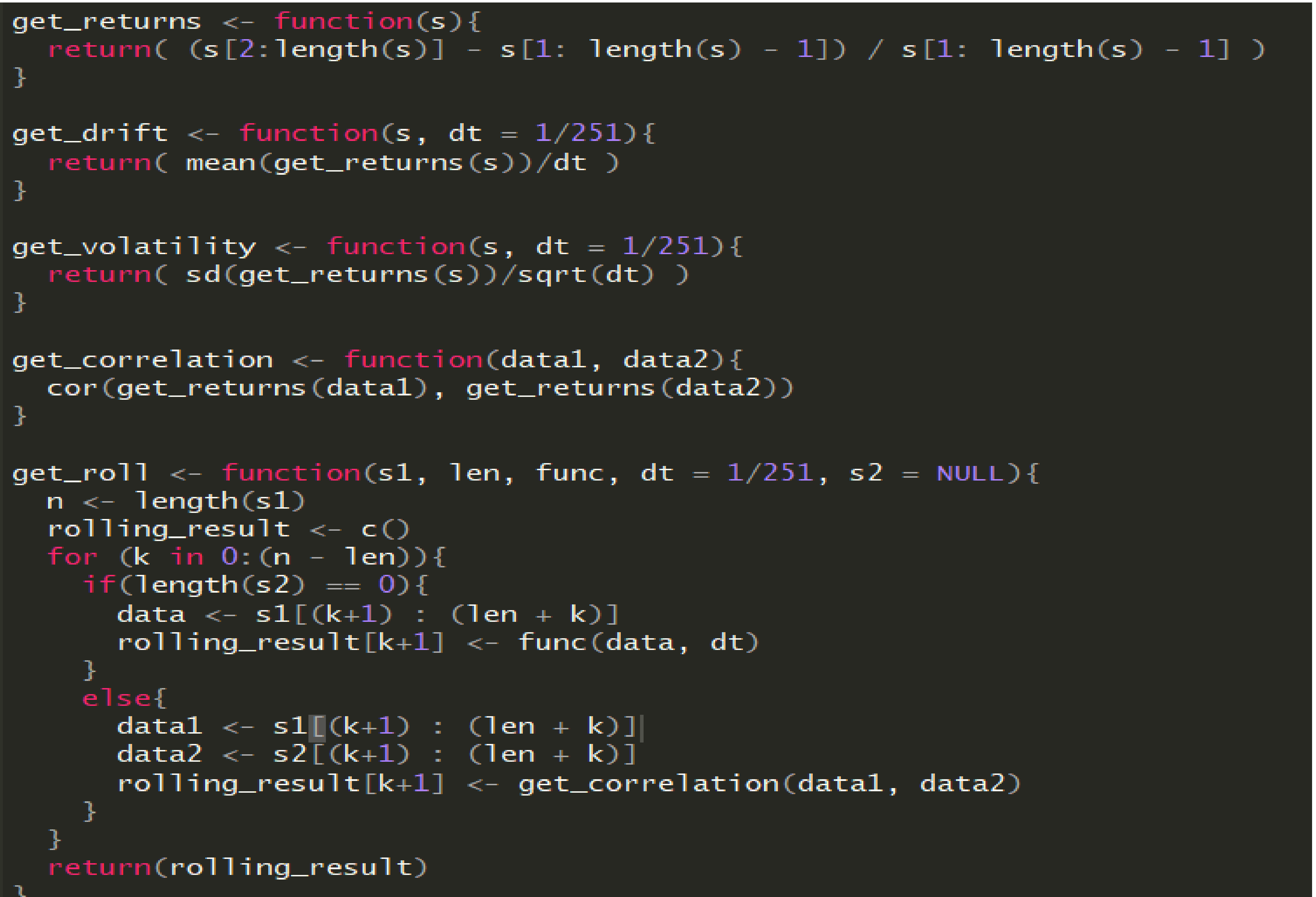
Model Blacka - Scholesa i eksploracja danych

Pobieranie danych



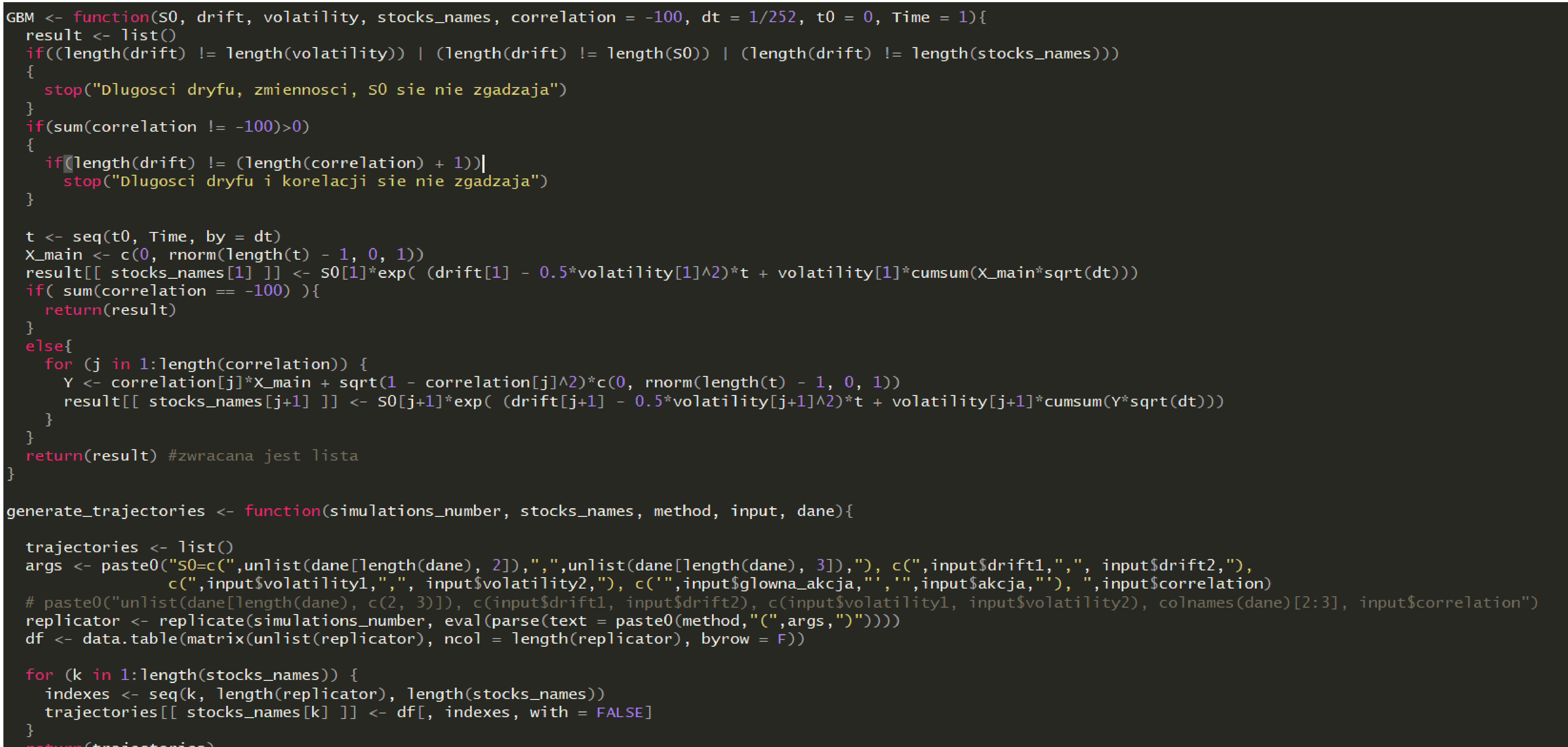
```r
pobieranie_danych <- function(skrot, od=NA, do=NA)
{
  if(is.na(od) & is.na(do))
  {
    data <- read.csv(paste0("https://stooq.pl/q/d/l/?s=",skrot,"&i=d"))
    data
  } else {
    if(!is.na(od) & is.na(do))
    {
      do <- as.character(Sys.Date())
    }
    if(is.na(od) & !is.na(do))
    {
      data_tmp <- read.csv(paste0("https://stooq.pl/q/d/l/?s=",skrot,"&i=d"))
      od <- as.character(min(as.Date(data_tmp$Data)))
    }
    od <- str_remove_all(od,"-")
    do <- str_remove_all(do,"-")
    data <- read.csv(paste0("https://stooq.pl/q/d/l/?s=",skrot,"&d1=",od,"&d2=",do,"&i=d"))
    data
  }
  setnames(data, "Zamkniecie", skrot)
  return(data[c("Data",skrot)])}
pobieranie_danych_wlasne_errory <- function(skrot, od=NA, do=NA)
{
  #tryCatch(pobieranie_danych(skrot, od, do),error=function(e) e, warning=function(w) w)
  if("warning" %in% class(tryCatch(pobieranie_danych(skrot, od, do),error=function(e) e, warning=function(w) w)))
    stop("Bledne wpisanie skrotu lub daty, sprobuj jeszcze raz") else
      pobieranie_danych(skrot,od,do)
}

to_date = function(data_table, data){

  data_table = merge(data_table,data,by='Data', all=FALSE)
  return <- data_table %>%
    arrange(Data = as.Date(Data, "%Y-%m-%d"))
  return(return)}
```

Estymacje parametrów



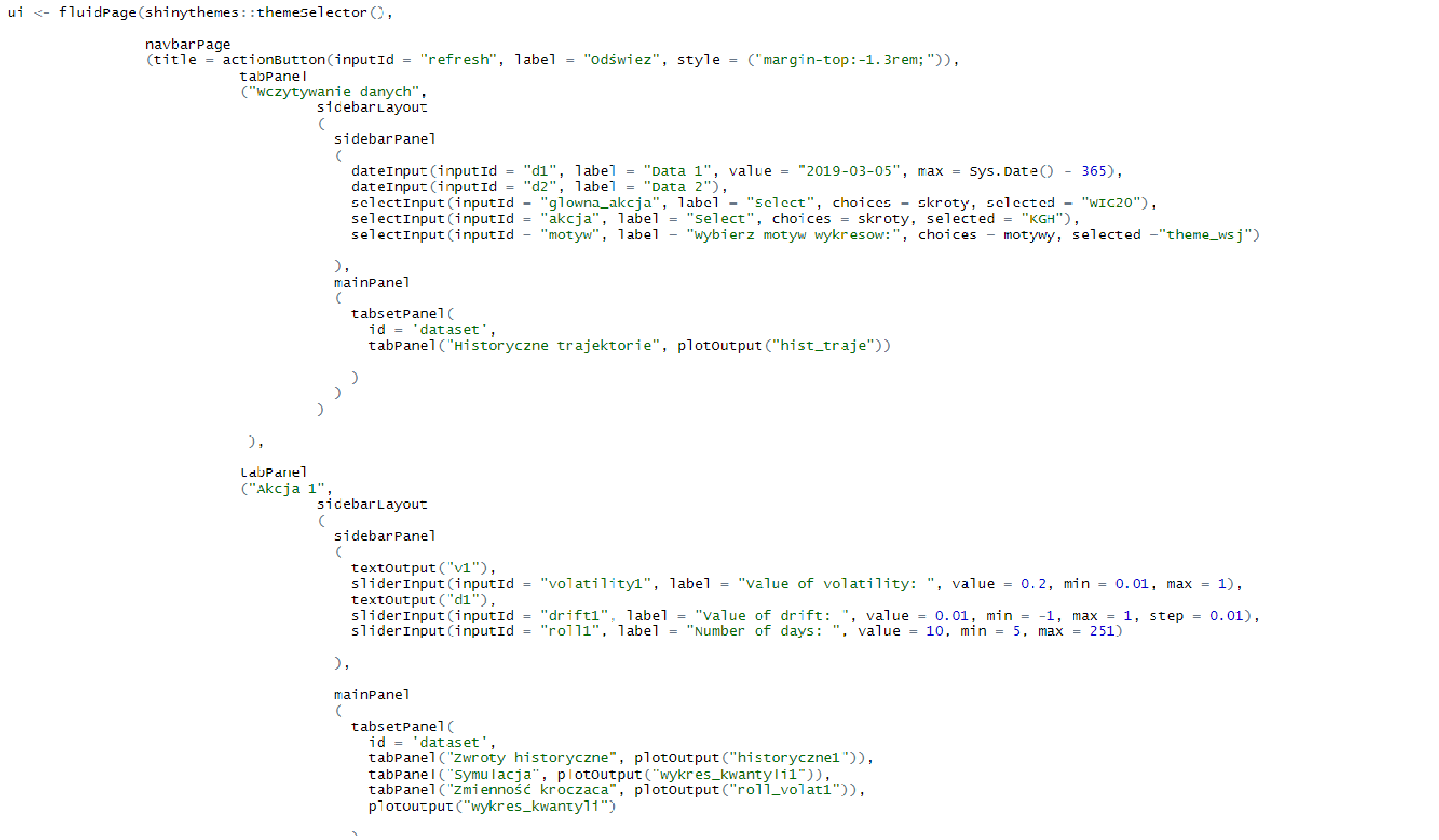
```r
get_returns <- function(s){
  return( (s[2:length(s)] - s[1: length(s) - 1]) / s[1: length(s) - 1] )
}

get_drift <- function(s, dt = 1/251){
  return( mean(get_returns(s))/dt )
}

get_volatility <- function(s, dt = 1/251){
  return( sd(get_returns(s))/sqrt(dt) )
}

get_correlation <- function(data1, data2){
  cor(get_returns(data1), get_returns(data2))
}

get_roll <- function(s1, len, func, dt = 1/251, s2 = NULL){
  n <- length(s1)
  rolling_result <- c()
  for (k in 0:(n - len)){
    if(length(s2) == 0){
      data <- s1[(k+1) : (len + k)]
      rolling_result[k+1] <- func(data, dt)
    }
    else{
      data1 <- s1[(k+1) : (len + k)]
      data2 <- s2[(k+1) : (len + k)]
      rolling_result[k+1] <- get_correlation(data1, data2)
    }
  }
  return(rolling_result)
}
```

Estymacja geometrycznego ruchu Browna



```r
GBM <- function(S0, drift, volatility, stocks_names, correlation = -100, dt = 1/252, t0 = 0, Time = 1){
  result <- list()
  if((length(drift) != length(volatility)) | (length(drift) != length(S0)) | (length(drift) != length(stocks_names)))
  {
    stop("Dlugosci dryfu, zmiennosci, S0 sie nie zgadzaja")
  }
  if(sum(correlation != -100)>0)
  {
    if(length(drift) != (length(correlation) + 1))
      stop("Dlugosci dryfu i korelacji sie nie zgadzaja")
  }

  t <- seq(t0, Time, by = dt)
  X_main <- c(0, rnorm(length(t) - 1, 0, 1))
  result[[ stocks_names[1] ]] <- S0[1]*exp( (drift[1] - 0.5*volatility[1]^2)*t + volatility[1]*cumsum(X_main*sqrt(dt)))
  if( sum(correlation == -100) ){
    return(result)
  }
  else{
    for (j in 1:length(correlation)) {
      Y <- correlation[j]*X_main + sqrt(1 - correlation[j]^2)*c(0, rnorm(length(t) - 1, 0, 1))
      result[[ stocks_names[j+1] ]] <- S0[j+1]*exp( (drift[j+1] - 0.5*volatility[j+1]^2)*t + volatility[j+1]*cumsum(Y*sqrt(dt)))
    }
  }
  return(result) #zwracana jest lista
}

generate_trajectories <- function(simulations_number, stocks_names, method, input, dane){

  trajectories <- list()
  args <- paste0("S0=c(",unlist(dane[length(dane), 2]),",",unlist(dane[length(dane), 3]),"), c(",input$drift1,",", input$drift2,"),
                 c(",input$volatility1,",", input$volatility2,"), c('",input$glowna_akcja,"','",input$akcja,"'), ",input$correlation)
  # paste0("unlist(dane[length(dane), c(2, 3)]), c(input$drift1, input$drift2), c(input$volatility1, input$volatility2), colnames(dane)[2:3], input$correlation")
  replicator <- replicate(simulations_number, eval(parse(text = paste0(method,"(",args,")"))))
  df <- data.table(matrix(unlist(replicator), ncol = length(replicator), byrow = F))

  for (k in 1:length(stocks_names)) {
    indexes <- seq(k, length(replicator), length(stocks_names))
    trajectories[[ stocks_names[k] ]] <- df[, indexes, with = FALSE]
  }
```

UI



```r
ui <- fluidPage(shinythemes::themeSelector(),

                navbarPage
                (title = actionButton(inputId = "refresh", label = "Odśwież", style = ("margin-top:-1.3rem;")),
                           tabPanel
                           ("Wczytywanie danych",
                                    sidebarLayout
                                    (
                                      sidebarPanel
                                      (
                                        dateInput(inputId = "d1", label = "Data 1", value = "2019-03-05", max = Sys.Date() - 365),
                                        dateInput(inputId = "d2", label = "Data 2"),
                                        selectInput(inputId = "glowna_akcja", label = "Select", choices = skroty, selected = "WIG20"),
                                        selectInput(inputId = "akcja", label = "Select", choices = skroty, selected = "KGH"),
                                        selectInput(inputId = "motyw", label = "Wybierz motyw wykresow:", choices = motywy, selected ="theme_wsj")

                                      ),
                                      mainPanel
                                      (
                                        tabsetPanel(
                                          id = 'dataset',
                                          tabPanel("Historyczne trajektorie", plotOutput("hist_traje"))

                                        )
                                      )
                                    )

                            ),

                           tabPanel
                           ("Akcja 1",
                                    sidebarLayout
                                    (
                                      sidebarPanel
                                      (
                                        textOutput("v1"),
                                        sliderInput(inputId = "volatility1", label = "Value of volatility: ", value = 0.2, min = 0.01, max = 1),
                                        textOutput("d1"),
                                        sliderInput(inputId = "drift1", label = "Value of drift: ", value = 0.01, min = -1, max = 1, step = 0.01),
                                        sliderInput(inputId = "roll1", label = "Number of days: ", value = 10, min = 5, max = 251)

                                      ),

                                      mainPanel
                                      (
                                        tabsetPanel(
                                          id = 'dataset',
                                          tabPanel("Zwroty historyczne", plotOutput("historyczne1")),
                                          tabPanel("Symulacja", plotOutput("wykres_kwantyli1")),
                                          tabPanel("Zmienność kroczaca", plotOutput("roll_volat1")),
                                          plotOutput("wykres_kwantyli")

                                        )
```

Server

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

  values <- reactiveValues()
  observe(
    {

    dane1 <- pobieranie_danych_wlasne_errory(input$glowna_akcja, input$d1, input$d2)
    dane2 <- pobieranie_danych_wlasne_errory(input$akcja, input$d1, input$d2)
    dane <- to_date(dane1, dane2)

    updateSliderInput(session, "roll1", max = nrow(dane) - 10)
    updateSliderInput(session, "roll2", max = nrow(dane) - 10)
    values$v1 <- paste0("Wyestymowana zmiennosc: ", as.character(round(get_volatility(dane[[input$glowna_akcja]]), 4)))
    values$v2 <- paste0("Wyestymowana zmiennosc: ", as.character(round(get_volatility(dane[[input$akcja]]), 4)))

    values$d1 <- paste0("Wyestymowany dryf: ", as.character(round(get_drift(dane[[input$glowna_akcja]]), 4)))
    values$d2 <- paste0("Wyestymowany dryf: ", as.character(round(get_drift(dane[[input$akcja]]), 4)))

    values$c <- paste0("Wyestymowana korelacja: ", as.character(round(get_correlation(dane[[input$glowna_akcja]], dane[[input$akcja]]), 4)))


    simulation <- generate_trajectories(1000, colnames(dane)[2:3], "GBM", input, dane)
    values$kwantyle1 <- quantile_plot(simulation[[input$glowna_akcja]], quantiles = c(0.05, 0.25, 0.5, 0.75, 0.95), motyw = input$motyw)
    values$kwantyle2 <- quantile_plot(simulation[[input$akcja]], quantiles = c(0.05, 0.25, 0.5, 0.75, 0.95), motyw = input$motyw)
    values$plot_ret1 <- returns_plot(dane[[input$glowna_akcja]], motyw = input$motyw)
    values$plot_ret2 <- returns_plot(dane[[input$akcja]], motyw = input$motyw)
    values$hist_plot1 <- hist_plot(dane[[input$glowna_akcja]], motyw = input$motyw)
    values$hist_plot2 <- hist_plot(dane[[input$akcja]], motyw = input$motyw)
    values$hist_traje <- historical_trajectories(dane, input$glowna_akcja, input$akcja, motyw = input$motyw)
    values$roll_volatility1 <- roll_plot(dane, input$roll1, get_volatility, nazwa1 = input$glowna_akcja, motyw = input$motyw)
    values$roll_volatility2 <- roll_plot(dane, input$roll2, get_volatility, nazwa1 = input$akcja, motyw = input$motyw)
    #values$liczba_dni <- nrow(dane)

    }
  )

  output$v1 <- renderText({values$v1})
  output$v2 <- renderText({values$v2})
  output$d1 <- renderText({values$d1})
  output$d2 <- renderText({values$d2})
  output$c <- renderText({values$c})
  output$wykres_kwantyli1 <- renderPlot({values$kwantyle1})
  output$wykres_kwantyli2 <- renderPlot({values$kwantyle2})
  output$wykres_zwrotow1 <- renderPlot({values$plot_ret1})
  output$wykres_zwrotow2 <- renderPlot({values$plot_ret2})
  output$historyczne1 <- renderPlot({grid.arrange(values$hist_plot1, values$plot_ret1, ncol = 1)})
  output$historyczne2 <- renderPlot({grid.arrange(values$hist_plot2, values$plot_ret2, ncol = 1)})
  output$hist_plot1 <- renderPlot({values$hist_plot1})
  output$hist_plot2 <- renderPlot({values$hist_plot2})
  output$hist_traje <- renderPlot({values$hist_traje})
  output$roll_volat1 <- renderPlot({values$roll_volatility1})
  output$roll_volat2 <- renderPlot({values$roll_volatility2})
```