Webscraping in real-time in R with R Shiny interface

Reagan Kesseku

2023-01-17

```
# set working directory ---------------------
setwd("D:/Ph.D_materials/Programming/R_programming/mdsr/cryptocurrencies")

# Load functions and packages ---------------------------
source("webscrap_pkg.R")
```

```
## package 'shinydashboard' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\bigco\AppData\Local\Temp\RtmpM1ePwn\downloaded_packages
```

```
# create functions to minimize repeated code usage
# ------------------------------------------------ get.data <- function(x){
# myurl <- read_html('https://coinmarketcap.com/gainers-losers/') # read our
# webpage as html myurl <- html_table(myurl) # convert to an html table for
# ease of use crypto_dat <- myurl[[1]] # pull the first item in the list
# crypto_dat$24h <- gsub('%','',crypto_dat$24h) # cleanup - remove
# non-characters crypto_dat$24h <- as.numeric(crypto_dat$24h) #cleanup -
# convert percentages column to numeric so we can sort # crypto_dat$Symbol <-
# as.factor(crypto_dat$) # cleanup - convert coin symbol to factor # #
# crypto_dat$Symbol <- factor(crypto_dat$Symbol, # levels =
# crypto_dat$Symbol[order(crypto_dat$'% 1h')]) # sort by gain value crypto_dat
# # return the finished data.frame }
```

```
# Import data and wrangle ----------------------- read webpage as html and
# convert to html table
get_data <- function(x) {
    url <- read_html("https://coinmarketcap.com/gainers-losers/") %>%
        html_table()
    url %>%
        typeof()  # this shows the data is a list

    # subset the first item
    crypto_dat <- url[[1]]
    crypto_dat %>%
        names()

    # rename # and 24 in data -----------------------
    crypto_dat <- crypto_dat %>%
        rename(Number = "#", change = "24h") %>%
        separate(Name, into = c("Name", "Other"), sep = "(?<=[A-Za-z])(?=[0-9])")
    # split data name at digit point ------------------------------
    crypto_dat = crypto_dat %>%
        mutate(Other = gsub("[[:digit:]]", "", Other), change = gsub("%", "", crypto_dat$change) %>%
            as.numeric()) %>%
        rename(Symbol = Other)

    # return data
    return(crypto_dat)
}

get_data()
```

```
## # A tibble: 30 x 6
##    Number Name              Symbol Price        change ‘Volume(24h)‘
##     <int> <chr>             <chr>  <chr>         <dbl> <chr>
##  1     30 Internet Computer ICP    $4.00          8.92 $25,672,118
##  2     64 Terra Classic     LUNC   $0.0...09466   7.99 $92,930,830
##  3     51 Fantom            FTM    $0.2557        5.93 $129,873,422
##  4     31 Filecoin          FIL    $3.64          5.08 $211,739,290
##  5     74 Injective         INJ    $5.88          4.25 $44,542,947
##  6     45 Aave              AAVE   $54.77         3.75 $38,551,384
##  7     27 Bitcoin Cash      BCH    $105.48        3.74 $74,463,761
##  8     84 GMX               GMX    $45.11         3.56 $24,068,668
##  9     23 Uniswap           UNI    $4.26          3.35 $98,301,988
## 10     81 Sui               SUI    $0.6845        3.24 $401,830,812
## # i 20 more rows
```

```
# sort data by percentage gain value in descending order
# ------------------------------------------------------
get_data() %>%
    arrange(desc(change))
```

```
## # A tibble: 30 x 6
##    Number Name              Symbol Price        change ‘Volume(24h)‘
##     <int> <chr>             <chr>  <chr>         <dbl> <chr>
##  1     30 Internet Computer ICP    $4.00          8.92 $25,672,118
##  2     64 Terra Classic     LUNC   $0.0...09466   7.99 $92,930,830
##  3     51 Fantom            FTM    $0.2557        5.93 $129,873,422
##  4     31 Filecoin          FIL    $3.64          5.08 $211,739,290
##  5     74 Injective         INJ    $5.88          4.25 $44,542,947
##  6     45 Aave              AAVE   $54.77         3.75 $38,551,384
##  7     27 Bitcoin Cash      BCH    $105.48        3.74 $74,463,761
##  8     84 GMX               GMX    $45.11         3.56 $24,068,668
##  9     23 Uniswap           UNI    $4.26          3.35 $98,301,988
## 10     81 Sui               SUI    $0.6845        3.24 $401,830,812
## # i 20 more rows
```

```
# Top coin name
top_coin_name <- get_data() %>%
    arrange(desc(change)) %>%
    select(Name) %>%
    first()
```

```
top_coin_gain <- get_data() %>%
    arrange(desc(change)) %>%
    select(change) %>%
    first()
```

```
# Build shiny app
# ----------------
# user interface (UI)
ui <- dashboardPage(skin = "yellow",

  # Create H E A D E R
  dashboardHeader(title = "Top Cryptocurrency Gainers"),

  # create S I D E B A R
  dashboardSidebar(
    h5("An interactive dashboard that shows the top gainers from the last 24hours from
       coinmarketcap.com and refreshes every 60 seconds."),

    br(),
    br(),
    br(),
    br(),
    br(),
    br(),
    br(),
    br(),
    br(),
    br(),

    h6("Built by Brad Lindblad in the R computing language
      [  R Core Team (2018). R: A language and environment for statistical computing. R Foundation for S
  Vienna, Austria. URL https://www.R-project.org/]"),
    br(),
    h6("R version 3.4.4 (2018-03-15) 'Someone to Lean On'")

  ),

  # B O D Y
  dashboardBody(
  fluidRow(
    # InfoBox
    infoBoxOutput("top.coin",
                  width = 3),
    # InfoBox
    infoBoxOutput("top.name",
                  width = 3)
    ),

  fluidRow(
column(
    # Datatable
      box(
        status = "primary",
        headerPanel("Crytocurrency data"),
        solidHeader = T,
        br(),
        DT::dataTableOutput("table", height = "340px"),
        width = 6,
        height = "550px"
      ),

      # Chart
      box(
        status = "primary",
        headerPanel("Chart"),
        solidHeader = T,
        br(),
        plotOutput("plot", height = "400px"),
        width = 6,
        height = "500px"
      ),
      width = 12
      )
    )
  )
)
#####################
#### S E R V E R ####
#####################
server <- function(input, output) {
# R E A C T I V E
  liveish_data <- reactive({
    invalidateLater(60000)    # refresh the report every 60k milliseconds (60 seconds)
    get_data()                # call our function from above
  })


  live.infobox.val <- reactive({
    invalidateLater(60000)    # refresh the report every 60k milliseconds (60 seconds)
    top_coin_gain         # call our function from above
  })


  live.infobox.coin <- reactive({
    invalidateLater(60000)    # refresh the report every 60k milliseconds (60 seconds)
    top_coin_name        # call our function from above
  })

  # D A T A   T A B L E   O U T P U T
  output$table <- DT::renderDataTable(DT::datatable(
    data <- liveish_data()))

  # P L O T   O U T P U T
  output$plot <- renderPlot({
    (ggplot(data = liveish_data(), aes(x= reorder(Symbol, - change), y = change)) +
                                 geom_bar(stat="identity", aes(fill = change > 10 ))  +
```

```r
                                 scale_fill_manual(values = c("pink","green")) +
                                 theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
                                 labs(x = "Name of Cryptocurrency", y = "Change (%) in last 24hours " )
                                 ggtitle("Gainers from the Last 24 Hours below and above 10%")) +
                                 theme_bw()
  })



  # I N F O B O X   O U T P U T - V A L
  output$top.coin <- renderInfoBox({
    infoBox(
      "Gain in Last 24 Hours",
      paste0(live.infobox.val(), "%"),
      icon = icon("signal"),
      color = "teal",
      fill = TRUE)
  })


  # I N F O B O X   O U T P U T - N A M E
  output$top.name <- renderInfoBox({
    infoBox(
      "Top Coin Name",
      live.infobox.coin(),
      icon = icon("bitcoin"),
      color = "teal",
      fill = TRUE)
  })

}
#####################
#### D E P L O Y ####
#####################
# Return a Shiny app objectshinyApp(ui = ui, server = server)
shinyApp(ui = ui, server = server)
```

Crytocurrency data

Chart