Web Scraping iPhone Data from Amazon

Reuben K

2024-10-30

Introduction

In this project, data about iPhones was scraped from the Amazon Indian page, after which a Shiny application was developed. Users are allowed to view prices models and ratings of iPhones, converted from Indian Rupees (INR) to Kenyan Shillings (KES) as of 30th October 2024. My skills in R, web scraping, data manipulation, and visualization are presented in the project.

Project Overview

The main objectives of the project include:

- Web Scraping: Collecting real-time data from the Amazon India website using R's rvest package.
- Data Cleaning and Transformation: Cleaning the scraped data, converting costs into Kenyan Shillings, and organizing it for analysis.
- Data Visualization: Creating plots to visualize the top 5 most expensive iPhones and providing an interactive user interface using Shiny.

Code Explanation

Step 1: Load Necessary Libraries

```
#library(rvest)
#library(dplyr)
#library(shiny)
#library(ggplot2)
```

Here, I load the necessary R packages for web scraping (rvest), data manipulation (dplyr), creating web applications (shiny), and plotting (ggplot2).

Step 2: Web Scraping Function

- The scrape_iphone_data function is responsible for fetching data from Amazon.
- It extracts titles, ratings, and costs of iPhones.

```
#scrape_iphone_data <- function() {
  #amazon <- tryCatch({
  #  read_html("https://www.amazon.in/s?k=iphone")
  #}, error = function(e) {
   # message("Error in loading the webpage: ", e$message)
    #return(NULL)
 # })

  #titles <- amazon %>% html_nodes(".a-size-medium") %>% html_text(trim = TRU
E)
  #ratings <- amazon %>% html_nodes(".a-icon-alt") %>% html_text(trim = TRUE)
  #costs <- amazon %>% html_nodes(".a-price-whole") %>% html_text(trim = TRUE
)

  #min_length <- min(length(titles), length(ratings), length(costs))
  #titles <- titles[1:min_length]
  #ratings <- ratings[1:min_length]
 # costs <- costs[1:min_length]

 # timestamp <- Sys.time()

 # iphone_data <- data.frame(
  #  Timestamp = rep(timestamp, min_length),
  #  Titles = titles,
   # Ratings = ratings,
   # Cost = costs,
    #stringsAsFactors = FALSE
 # )

 # csv_file_path <- "iphone_data.csv"

 # write.table(
   # iphone_data,
   # file = csv_file_path,
   # sep = ",",
   # row.names = FALSE,
   # col.names = !file.exists(csv_file_path),
    #append = TRUE
 # )
```

```
 # return(iphone_data)
#}
```

Step 3: Data Cleaning and Conversion

The clean_and_convert_cost function converts the cost from INR to KES.

```
#clean_and_convert_cost <- function(cost_str, conversion_rate) {
  #cost_str <- gsub(" ", "", cost_str)
  #cost_numeric <- suppressWarnings(as.numeric(gsub(",", "", cost_str)))
  #if (is.na(cost_numeric)) {
   # warning("Invalid cost value: ", cost_str)
   # return(NA)
 # }
  # Convert to KES
  #cost_in_kes <- cost_numeric * conversion_rate
  #return(cost_in_kes)
#}
```

Step 4: User Interface and Server Logic

In the UI section, I create a fluid page with a download button, a table to display the data, and plots to visualize the top 5 most expensive iPhones.

```
#ui <- fluidPage(
 # titlePanel("iPhone Data"),

 # mainPanel(
  # div(style = "text-align: right; margin-bottom: 10px;",
        #downloadButton("download_data", "Download Data as CSV")),
    #plotOutput("top_5_expensive_phones"),
    #tableOutput("iphone_table"),
    #verbatimTextOutput("error_message"),
    #verbatimTextOutput("current_time")
 # )
#)

#server <- function(input, output) {
 # conversion_rate <- 16.06

  #cleaned_data <- reactive({
   # if (file.exists("iphone_data.csv")) {
      #iphone_data <- read.table("iphone_data.csv", sep = ",", header = TRUE)
     # iphone_data$Cost <- sapply(iphone_data$Cost, clean_and_convert_cost, c
onversion_rate)
    #  return(iphone_data)
   # } else {
  #    output$error_message <- renderText({
```

```
      #  "File 'iphone_data.csv' not found. Please ensure the file exists."
     # })
    #  return(NULL)
   # }
  #})

  #output$top_5_expensive_phones <- renderPlot({
    #data <- cleaned_data()
    #if (!is.null(data)) {
     # top_5_phones <- data %>%
      #  arrange(desc(as.numeric(gsub("KSh ", "", Cost)))) %>%
      #  head(5)
      #ggplot(top_5_phones, aes(x = reorder(Titles, -as.numeric(gsub("KSh ",
"", Cost))),
                               # y = as.numeric(gsub("KSh ", "", Cost)))) +
        #geom_bar(stat = "identity", fill = "blue") +
      #  labs(title = "Top 5 Most Expensive iPhones",
          #  x = "iPhone Models", y = "Cost (KES)") +
        #theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
       # scale_y_continuous(labels = scales::comma) +
       # coord_flip() +
       # theme_minimal()
 #   }
 # })
#}

#shinyApp(ui = ui, server = server)
```

Conclusion

This project not only demonstrates my ability to scrape data from the web but also showcases my skills in data manipulation and visualization using R. The Shiny app provides a user-friendly interface for anyone interested in exploring the latest iPhone models available on Amazon.

Future Enhancements

Potential improvements for this project could include:

- Adding functionality for users to choose different currencies for conversion.
- Scheduling the scraping function to run automatically at regular intervals.
- Enhancing the visualizations with more interactive features.