VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELGAUM-590014

A Mini project

On

Web Scraping and Time Series Analysis using R Programming

A mini project report submitted in partial fulfilment of the requirements for the IV Semester degree of ***Bachelor of Engineering in Computer Science and Technology*** *of Visvesvaraya Technological University, Belgaum.*

Submitted by

Siddharth R	1DT21CS149
Suhas Pyde	1DT21CS160
Tharun V	1DT21CS170
Thej Venkat P	1DT21CS171

Under the Guidance of

Shilpa M
Assistant Professor
Computer Science and Engineering
Dayananda Sagar Academy of Technology & Management

Department of Computer Science and Engineering

DAYANANDA SAGAR ACADEMY OF TECHNOLOGY AND MANAGEMENT

Udayapura, Kanakapura Road, Bangalore-560082 ,2021-2022

CONTENTS

ABSTRACT

Web scraping is the process of extracting data from websites. It involves accessing and using data from websites without the explicit permission of the website owner. Web scraping can raise ethical concerns, and it is a good practice to respect the terms of use for a website and to seek written permission before scraping large amounts of data .
R is a programming language built for data analysis. It provides useful tools for data wrangling and dynamic typing.

Rvest is one of the most popular R packages providing web scraping functionalities . In order to get started with web scraping in R, you will first need to install R and RStudio (if needed) and then install the rvest package .
Time series analysis is a statistical technique that deals with time series data, or data that is indexed by time. It is used to analyze trends, patterns, and relationships in data over time . R provides several packages for time series analysis, including forecast, tseries, and zoo .

In summary, web scraping and time series analysis are two important techniques in data science. Web scraping allows you to extract data from websites, while time series analysis allows you to analyze trends and patterns in time series data. Both techniques can be performed using R, which provides several packages for each task.

INTRODUCTION

Web scraping is a powerful technique that has many applications in data science. It can be used to extract data from websites for research, analysis, or other purposes. For example, web scraping can be used to gather data on product prices, customer reviews, or social media activity. Web scraping can also be used to monitor changes to websites over time, such as changes to stock prices or news articles.

Time series analysis is another important technique in data science. It is used to analyze data that is indexed by time, such as stock prices or weather data. Time series analysis can be used to identify trends and patterns in data over time, which can be useful for forecasting future trends or making decisions based on historical data.

One of the main advantages of web scraping and time series analysis is that they allow you to work with large amounts of data quickly and efficiently. With web scraping, you can extract data from multiple websites at once, which can save you time and effort. With time series analysis, you can analyze large datasets over long periods of time, which can help you identify long-term trends and patterns.

Another advantage of web scraping and time series analysis is that they are both highly customizable. With web scraping, you can choose which websites to scrape and which data to extract. With time series analysis, you can choose which statistical techniques to use and how to visualize your results.

In summary, web scraping and time series analysis are two powerful techniques in data science that have many applications and advantages. By using these techniques together, you can extract data from websites and analyze trends and patterns over time.

In Our Project We are utilizing the above mentioned technologies for retrieving the Amazon Product data (Price) using the WebScraping and storing it in the Local Space to later use the data for various analysis of the prices like Time Series Analysis.

PROGRAM CODE

Amazon Price Checker

```
{

library(csv)
library(rvest)

getPrice=function(link){
  page=read_html(link)
  price<- page %>% html_elements(".a-price-whole")
  price=price[1]
  price=as(price,"character")
  len <- nchar(price)
  price=substr(price,29,len-45)
  price=strsplit(price,",")
  priceint=""
  for(i in price[[1]]){
    priceint=paste0(priceint,i)
  }
  price=as.integer(priceint)
  print(price)
  return(price)
}

bookEx <- read_excel("C:/Users/Thej Venkat/Desktop/Thej/sem4/r
programming/Products.xlsx", col_names = FALSE)
ReadPrices<-function(){
  for (i in 1:nrow(bookEx)) {
    name=as.character(bookEx[i,1])
    print(paste("Name:", name))
    link=bookEx[i,2]
    link=as.character(link)
    pr<-getPrice(link)
    writeFile(name,pr)
  }
}

writeFile=function(file_name,price){
  file_name=paste0("C:/Users/Thej Venkat/Desktop/Thej/sem4/r
programming/",file_name,".csv")
  tryCatch({
```

```
    new_row <- data.frame(price=price)
    write.table(new_row, file = file_name, sep = ",", append = TRUE,
                quote = FALSE, col.names = FALSE, row.names = TRUE)
  },
  error=function(e){
    print(e)
    price=c(price)
    price.data <- data.frame(price)
    write.csv(file_name,price.data)
  })
}

while(TRUE){
  Sys.sleep(5)
  n=readline("Read prices (0=NO):")
  if(n=='0'){
    break
  }else{
    ReadPrices()
  }
}

}
```

Time Series Analysis

{

```
timeSeries=function(){
  name=(readline("Enter the product name:"))
  start=c()
  start[1]=as.integer(readline("starting year:"))
  start[2]=as.integer(readline("starting month:"))
  file_name=paste0("C:/Users/Thej Venkat/Desktop/Thej/sem4/r
programming/",name,".csv")
  if (file.exists(file_name)) {
    book=read.csv(file_name,header=FALSE)
    n=nrow(book)
    print(paste("Readings available for first",n,"Months."))
    end=c()
    end[1]=as.integer(readline("ending year:"))
    end[2]=as.integer(readline("ending month:"))
    ts <- ts(book, start = start, end = end, frequency = 12)
    plot(ts, type = "o", col = "blue", main = "Time Series Plot", xlab = "Time", ylab =
"Price")

  } else {
    print("Product file does not exist")
  }
}
timeSeries()
```

}

ALGORITHM

Amazon Price Checker

Get Price{

1. Get the link of the product
2. Get the html page source of the product using the link using Rvest
3. Find the class of the price web element
4. Select the web element using class
5. Reduce the text to price
6. Return the Price

}

Book Ex{

1. Read the excel worksheet for the list of products and prices
2. For each product in list:
    a. Get the name of the product
    b. Get the price of the product using GetPrice
    c. Write the price of the product in the designated file using WriteFile

}

WriteFile{

1. Get the name of the product
2. Create the csv file with the same name if not present
3. Write the price into the csv file

}

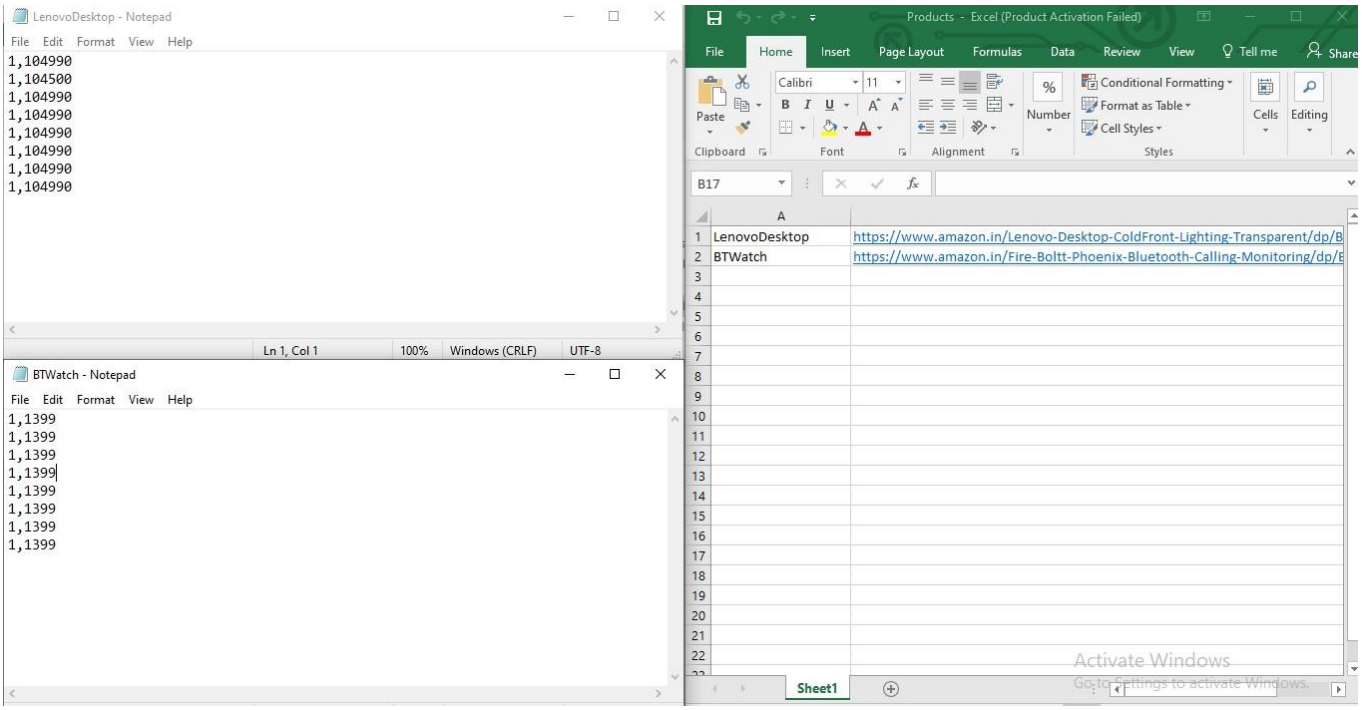
While (True):

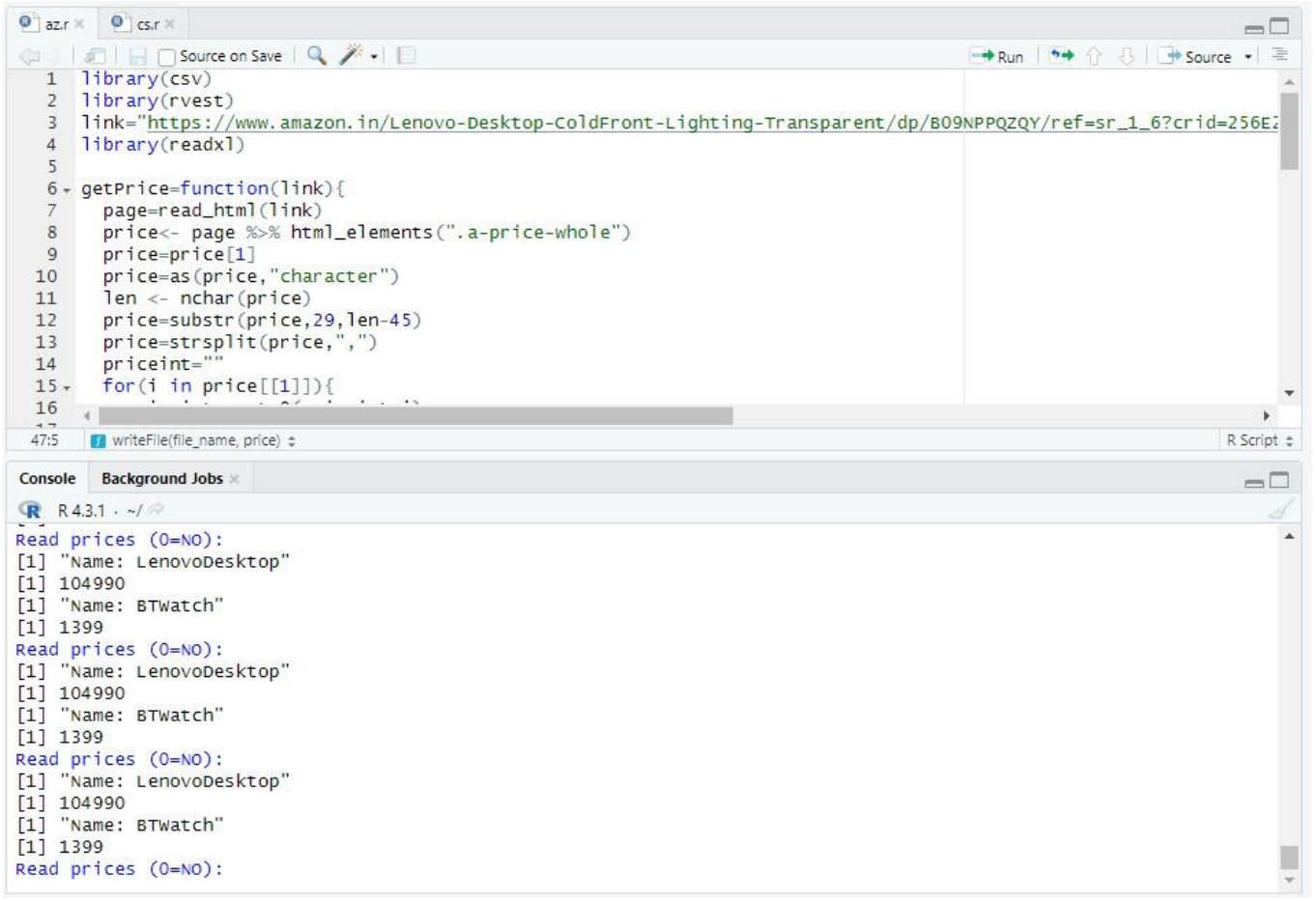
ReadPrices()

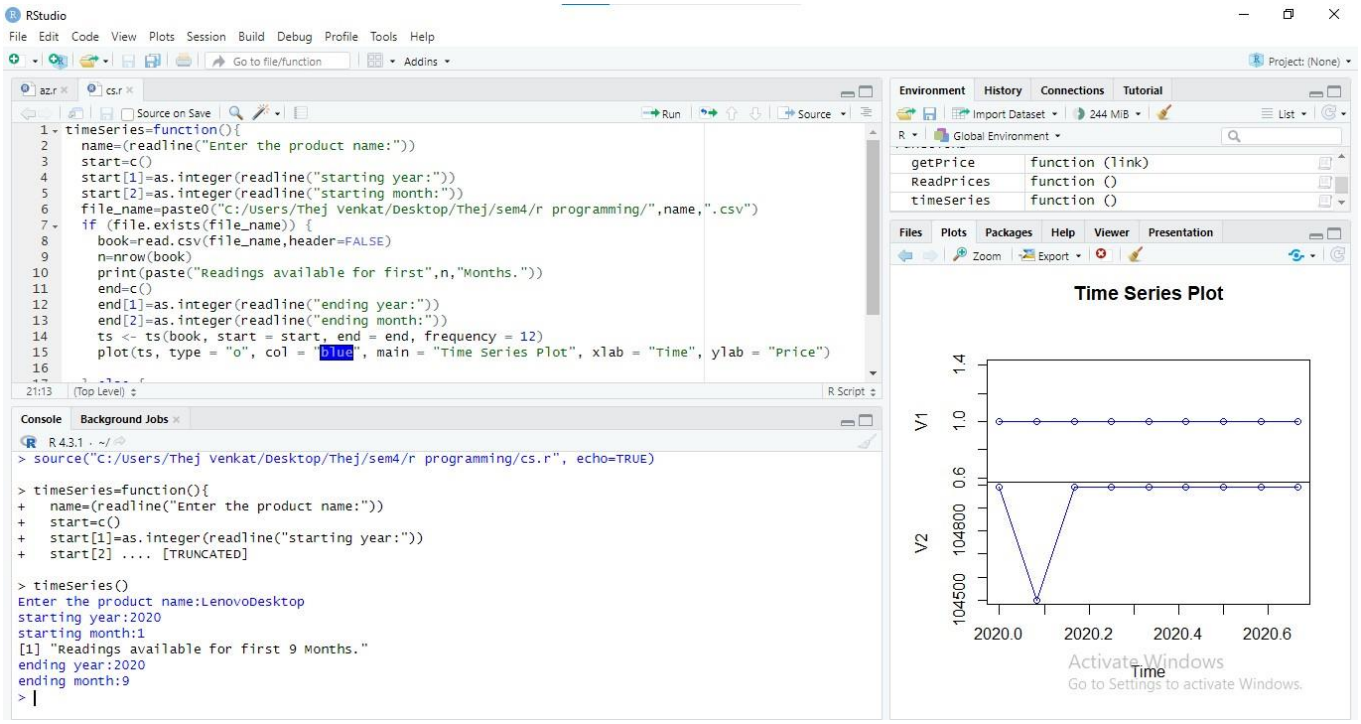
Time Series Analysis

1. Enter the Product name
2. Check if the product file exists
3. Enter the Starting Year and Month (2020,1)
4. Print the number of available observations of the prices of Product
5. Enter the Ending Year and Month
6. Plot the Time Series Analysis Graph

OUTPUT


LenovoDesktop - Notepad
File Edit Format View Help
1,104990
1,104500
1,104990
1,104990
1,104990
1,104990
1,104990
1,104990
Ln 1, Col 1 100% Windows (CRLF) UTF-8
BTWatch - Notepad
File Edit Format View Help
1,1399
1,1399
1,1399
1,1399
1,1399
1,1399
1,1399
1,1399
Products - Excel (Product Activation Failed)
File Home Insert Page Layout Formulas Data Review View Tell me Share
Paste Clipboard Calibri 11 Font Alignment Number Conditional Formatting Format as Table Cell Styles Styles Cells Editing
B17
A
1 LenovoDesktop https://www.amazon.in/Lenovo-Desktop-ColdFront-Lighting-Transparent/dp/B
2 BTWatch https://www.amazon.in/Fire-Boltt-Phoenix-Bluetooth-Calling-Monitoring/dp/B
Sheet1
Activate Windows
Go to Settings to activate Windows.


az.r
cs.r
Source on Save
Run
Source
1 library(csv)
2 library(rvest)
3 link="https://www.amazon.in/Lenovo-Desktop-ColdFront-Lighting-Transparent/dp/B09NPPQZQY/ref=sr_1_6?crid=256E
4 library(readxl)
5
6 getPrice=function(link){
7 page=read_html(link)
8 price<- page %>% html_elements(".a-price-whole")
9 price=price[1]
10 price=as(price,"character")
11 len <- nchar(price)
12 price=substr(price,29,len-45)
13 price=strsplit(price,",")
14 priceint=""
15 for(i in price[[1]]){
16
47:5 writeFile(file_name, price) R Script
Console Background Jobs
R 4.3.1 · ~/
Read prices (0=NO):
[1] "Name: LenovoDesktop"
[1] 104990
[1] "Name: BTWatch"
[1] 1399
Read prices (0=NO):
[1] "Name: LenovoDesktop"
[1] 104990
[1] "Name: BTWatch"
[1] 1399
Read prices (0=NO):
[1] "Name: LenovoDesktop"
[1] 104990
[1] "Name: BTWatch"
[1] 1399
Read prices (0=NO):


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
az.r
cs.r
timeSeries=function(){
name=(readline("Enter the product name:"))
start=c()
start[1]=as.integer(readline("starting year:"))
start[2]=as.integer(readline("starting month:"))
file_name=paste0("C:/Users/Thej Venkat/Desktop/Thej/sem4/r programming/",name,".csv")
if (file.exists(file_name)) {
book=read.csv(file_name,header=FALSE)
n=nrow(book)
print(paste("Readings available for first",n,"Months."))
end=c()
end[1]=as.integer(readline("ending year:"))
end[2]=as.integer(readline("ending month:"))
ts <- ts(book, start = start, end = end, frequency = 12)
plot(ts, type = "o", col = "blue", main = "Time Series Plot", xlab = "Time", ylab = "Price")
Console
Background Jobs
R 4.3.1 · ~/
> source("C:/Users/Thej Venkat/Desktop/Thej/sem4/r programming/cs.r", echo=TRUE)
> timeSeries=function(){
+ name=(readline("Enter the product name:"))
+ start=c()
+ start[1]=as.integer(readline("starting year:"))
+ start[2] .... [TRUNCATED]
> timeSeries()
Enter the product name:LenovoDesktop
starting year:2020
starting month:1
[1] "Readings available for first 9 Months."
ending year:2020
ending month:9
Environment
History
Connections
Tutorial
getPrice function (link)
ReadPrices function ()
timeSeries function ()
Files
Plots
Packages
Help
Viewer
Presentation
Time Series Plot
V1
V2
1.4
1.0
0.6
104800
104500
2020.0
2020.2
2020.4
2020.6
Time
Activate Windows
Go to Settings to activate Windows.

CONCLUSION

In conclusion, this report has presented the results of a comprehensive R project focused on time series analysis using web scraping techniques. Throughout the project, we have demonstrated the power and versatility of R in collecting and analyzing time series data from various online sources.

Key findings and takeaways from this project include:

Data Collection: We successfully scraped time series data from websites, APIs, and other online sources, demonstrating the ability to retrieve data efficiently and effectively for analysis.

Data Cleaning and Preprocessing: We emphasized the importance of data cleaning and preprocessing in time series analysis, highlighting techniques such as handling missing values, smoothing noisy data, and converting timestamps to appropriate formats.

Visualization: We utilized R's powerful visualization libraries to create informative plots and charts that facilitate data exploration and help identify underlying patterns and trends.

Time Series Analysis: Various time series analysis techniques were applied, including decomposition, autocorrelation, and moving averages, to gain insights into the data and extract meaningful information.

Web Scraping Automation: We discussed strategies for automating web scraping tasks, ensuring that data collection remains up-to-date and relevant.

Challenges and Considerations: It is important to note that web scraping can present challenges related to website structure, data availability, and ethical considerations. We addressed these challenges and highlighted the importance of adhering to ethical scraping practices.

Future Directions: To further enhance the project, future work could explore more advanced modeling techniques, incorporate machine learning algorithms, and consider alternative data sources. Additionally, the automation of data collection and analysis pipelines can be improved for scalability and efficiency.

In summary, this R project demonstrates the valuable insights that can be gained through time series analysis using web scraping. The combination of data collection, preprocessing, analysis, and visualization in R provides a powerful toolkit for understanding and forecasting time-dependent data. As businesses and researchers continue to seek data-driven insights, the skills and techniques showcased in this project will remain highly relevant and useful in a wide range of applications.

REFERENCES

1. **Online Tutorials and Documentation:**
   - R Project's official website: https://www.r-project.org/
   - ggplot2 documentation: https://ggplot2.tidyverse.org/
2. **Blogs and Websites:**
   - R-bloggers: https://www.r-bloggers.com/
   - RStudio blog: https://blog.rstudio.com/
   - R Views: https://rviews.rstudio.com/
   - Geeks for Geeks: https://www.geeksforgeeks.org/
3. **Forums and Q&A:**
   - Stack Overflow's R tag: https://stackoverflow.com/questions/tagged/r