NITTE EDUCATION TRUST

N.M.A.M. INSTITUTE OF TECHNOLOGY
(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)
Nitte – 574 110, Karnataka, India

Department of Computer Science and Engineering

(B.E. Computer Science & Engineering Program Accredited by NBA from 2018-19 to 2021-25)

Report on Mini Project

Stock Market Analysis

Course Code : 21CSA31 Course

Name : R Programming

Semester: III SEM
Section: B

Submitted To:

Mr. Sunil Kumar Aithal
Assistant Professor [GD-II]
Department of Computer Science
and Engineering

Submitted By:

Ganesh Nayak (4NM21CS063)
Kritarth(4NM21CS080)
Kshitiz Mangal(4NM21CS082)

Date of submission: 21-01-2023

Signature of Course Instructor

ABSTRACT

Stock price analysis is a popular and important topic in financial and academic studies. Share Market is an untidy place for predicting since there are no significant rules to estimate or predict the price of share in the share market. Many methods like technical analysis, fundamental analysis, time series analysis and statistical analysis are all used to attempt to predict the price in the share market but none of these methods are proved as a consistently acceptable prediction tool.

Since comparing the performance of many stocks in a single visualization can be time consuming and tedious. So, in our project we decided to use R and Shiny web application by which we could easily create and track a stock portfolio to see how individual stocks perform over time.

We analyze the stock market data using the R programming language and Shiny, a web application framework for R. First, we created the plots in R using the data from yahoo finance and plotted for seven companies. Additionally, we created an interactive Shiny app that allows users to explore the data and perform their own analysis. This project demonstrates the power of R and Shiny for analyzing and understanding financial data.

TABLE OF CONTENTS

TITLE i

ABSTRACT... ii

INTRODUCTION...1

PROBLEM STATEMENT 2

OBJECTIVES 3

METHODOLOGY 4

IMPLEMENTATION:-...5

RESULTS AND DISCUSSIONS...11

CONCLUSION AND FUTURE SCOPE...12

REFERENCES..13

INTRODUCTION

“Stock” has been one of the most popular words that are tossed around every day in the financial industry. If we have to understand what the stock market analysis is, we have to understand what stocks are. A stock is a financial instrument that reflects ownership of a portion of a company.

This entitles the stockholder to a share of the corporation's assets and earnings according to the amount of stock they possess. "Shares" are the units of stock. At the stock exchanges, stocks are generally bought and sold, and this constitutes the “stock market”.

The stock market is a key driver of the global economy, with billions of dollars in trades occurring on a daily basis. Understanding how the market performs, and how individual stocks and sectors behave, is crucial for investors, analysts, and other market participants. In this project, we seek to analyze historical stock market data and gain insights into key metrics such as returns, risk, and volatility.

We use the R programming language and Shiny, a web application framework for R, to analyze and visualize the data. R is a powerful and widely-used language for statistical computing and data analysis, and Shiny allows us to build interactive, web-based applications that allow users to explore the data and perform their own analyses.

In this project, we focus on a selection of publicly-traded companies and examine how their stock performance has changed over time. We also build an interactive Shiny app that allows users to select different companies and metrics to analyze and visualize.

Overall, this project aims to demonstrate the capabilities of R and Shiny for analyzing and understanding financial data, and can serve as a useful tool for investors and analysts interested in the stock market.

PROBLEM STATEMENT

Monitoring Stock Performance

Comparing the performance of many stocks in a single visualization can be time consuming. It is particularly tedious if you want to do this over and over again. With the help of R and Shiny, you can easily create and track a stock portfolio to see how individual stocks perform over time — all in one interactive visualization.

OBJECTIVES

The aim of this project is to

- To identify factors affecting the stock market.
- To visualize a large set of data using graphs.
- To predict an approximate value of the stock price.
- To provide analysis for users through shiny web application.
- To develop trading strategies that can help traders to improve their profit.

These predicted and analyzed data can be observed by individuals to know the financial status of companies and their companies.

METHODOLOGY

The steps involved in this project are:

1. We gathered stock closing prices of Apple, Adidas, Amazon, Nike, Puma, Tesla and Tata Motors
2. We created a simple user interface using shiny web application which allows the user to select the data to be displayed
3. At the end we defined the server function.

IMPLEMENTATION:-

To implement our project first we need to install few libraries using the function install.packages().

```
 library(shiny)
library(shinyWidgets)
library(shinythemes)
library(plotly)
library(tidyverse)
library(tidyquant)
```

Getting the data:-

First we store the 7 companies in a variable called tickers and we use two stock indices as benchmarks. Then we use tq_get() from the library tidyquant which we had defined earlier. We are using tq_get() to get stock(closing) prices from Yahoo Finance(https://finance.yahoo.com/lookup/). The to and from arguments are used to specify the desired date range.

```
tickers <- c("TSLA","AMZN","ADDYY","AAPL","TTM","NKE","PBYI")
benchmarks <- c("^NDX","^GSPC")

prices <- tq_get(tickers,
            get  = "stock.prices",
            from = today()-months(12),
            to   = today(),
            complete_cases = F) %>%
            select(symbol,date,close)

bench <- tq_get(benchmarks,
           get  = "stock.prices",
           from = today()-months(12),
           to   = today()) %>%
           select(symbol,date,close)
```

The User Interface:-

The code specifying the user interface will be wrapped in the fluidpage function and saved in a variable that we call ui . This is demonstrated at the bottom of the article where we will put it all together.

Our application consists of three main ui components:

1. Title Panel
2. Side Bar where we can select and filter the data
3. A main panel for visualizing the data.

The Title Panel:-

```
titlePanel("Stock Market Analysis")
```

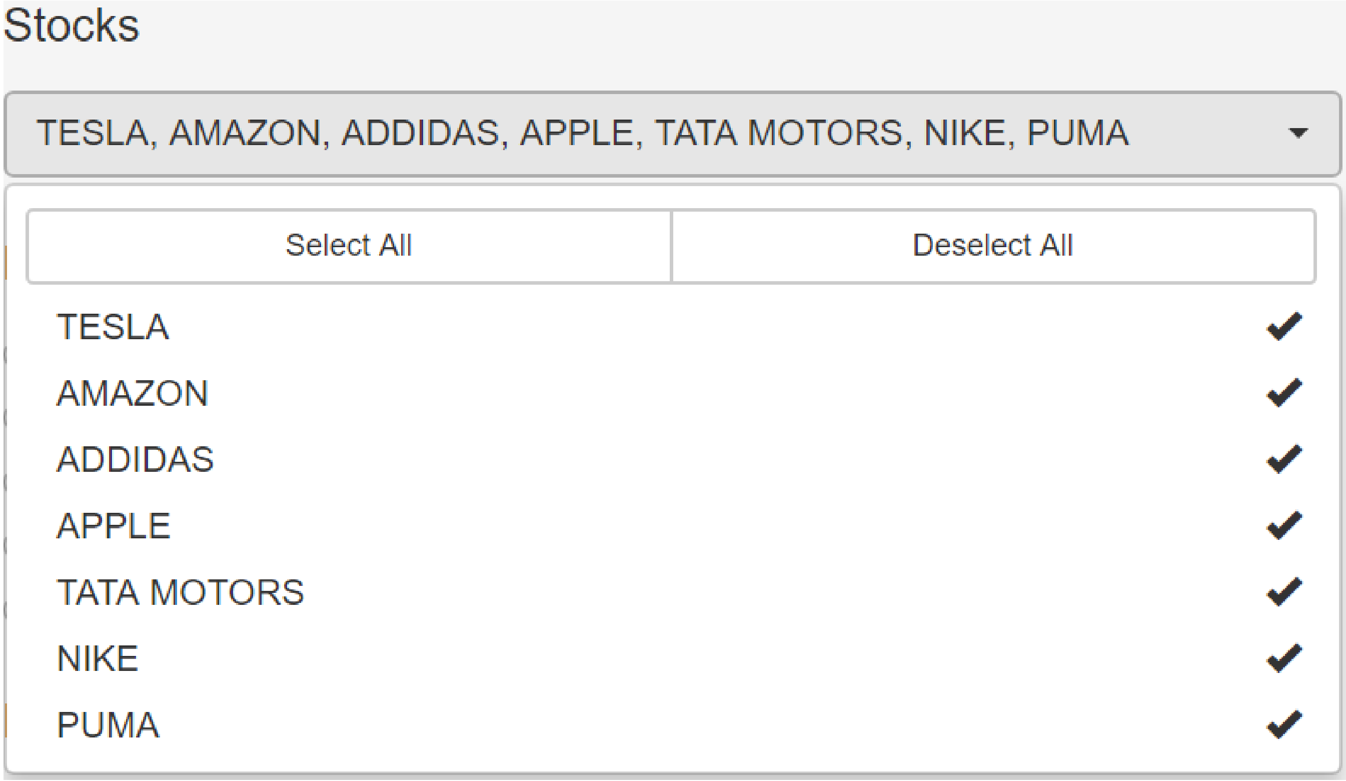
The SideBar:-

The sidebar allows the user to select the required companies from a particular time period and it allows the user to add the benchmark or not.

The function pickerInput lets us pick between stocks and select/deselect all.

All the companies are selected by default using the selected argument. It is important to specify the inputId in order to refer the selected stock in our server logic

```
pickerInput(
          inputId = "stocks",
          label = h4("Stocks"),
          choices = c(
            "TESLA"      = tickers[1],
            "AMAZON"   = tickers[2],
            "ADIDAS"     = tickers[3],
            "APPLE"       = tickers[4],
            "TATA MOTORS"= tickers[5],
            "NIKE"     = tickers[6],
            "PUMA"      = tickers[7]),
          selected = tickers,
          options = list(`actions-box` = TRUE),
          multiple = T
         )
```



Stocks

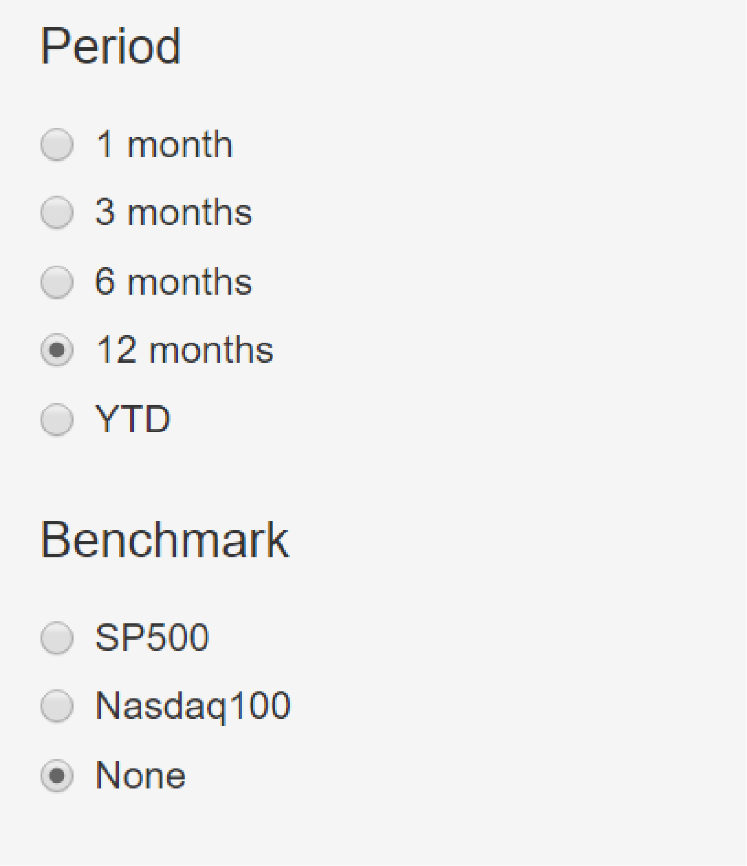
TESLA, AMAZON, ADDIDAS, APPLE, TATA MOTORS, NIKE, PUMA

TESLA ✓
AMAZON ✓
ADDIDAS ✓
APPLE ✓
TATA MOTORS ✓
NIKE ✓
PUMA ✓

The radioButtons() which allows the user to select the different time ranges to choose from as well as whether to include a benchmark index in the visualization.

```
#Time period
 radioButtons("period", label = h4("Period"),
 choices = list("1 month" = 1, "3 months" = 2, "6 months" = 3, "12 months" =  4, "YTD" = 5),
  selected = 4 )

 #Benchmark
 radioButtons("benchmark", label = h4("Benchmark"),
 choices = list("SP500" = 1, "Nasdaq100" = 2,"None" = 3),
 selected = 3
```



Period

○ 1 month

○ 3 months

○ 6 months

● 12 months

○ YTD

Benchmark

○ SP500

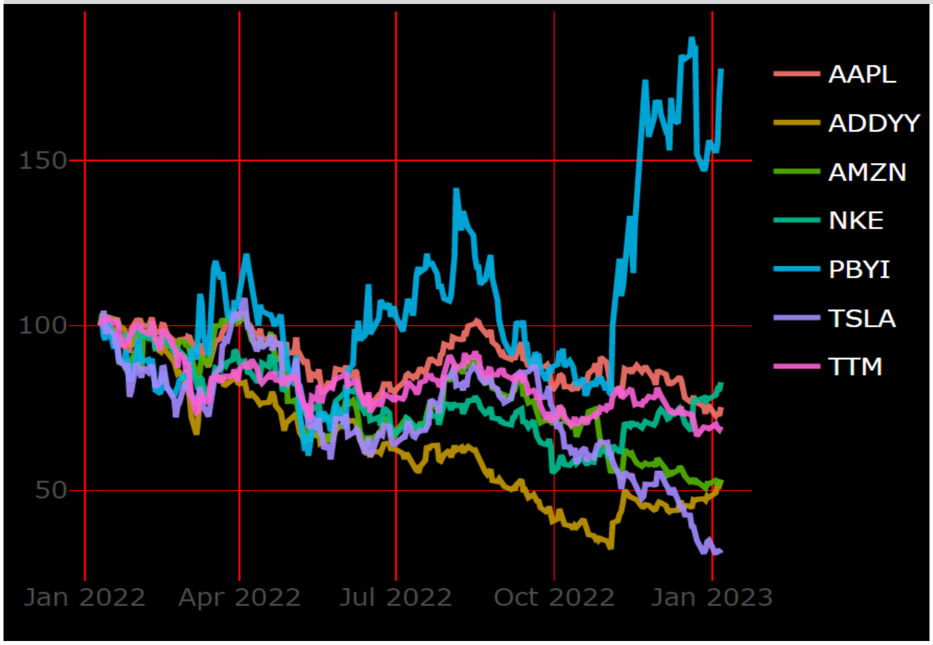
○ Nasdaq100

● None

The Main Panel:-

In the main panel we display the plot.

```
mainPanel(plotlyOutput(“plot” , height = 600))
```



Server Functions:-

We must specify how our app should behave based on user input. All of this is completed in our code's server section. We wrap the code because we want our app to respond to user input and change the visuals using the function
observeEvent .

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

  # server logic based on user input
  observeEvent(c(input$period,input$stocks,input$benchmark), {

    prices <- prices %>%
      filter(symbol %in% input$stocks)

    if (input$period == 1) {
      prices <- prices %>%
        filter(
          date >= today()-months(1)) }

    if (input$period == 2) {
      prices <- prices %>%
        filter(date >= today()-months(3)) }

    if (input$period == 3) {
      prices <- prices %>%
        filter(date >= today()-months(6)) }

    if (input$period == 5) {
      prices <- prices %>%
        filter(year(date) == year(today())) }

    if (input$benchmark == 1) {
      bench <- bench %>%
        filter(symbol=="^GSPC",
               date >= min(prices$date))
      prices <- rbind(prices,bench) }

    if (input$benchmark == 2) {
      bench <- bench %>%
        filter(symbol=="^NDX",
               date >= min(prices$date))
      prices <- rbind(prices,bench) }
```

Finally, we can plot our data combining ggplot2 and plotly .

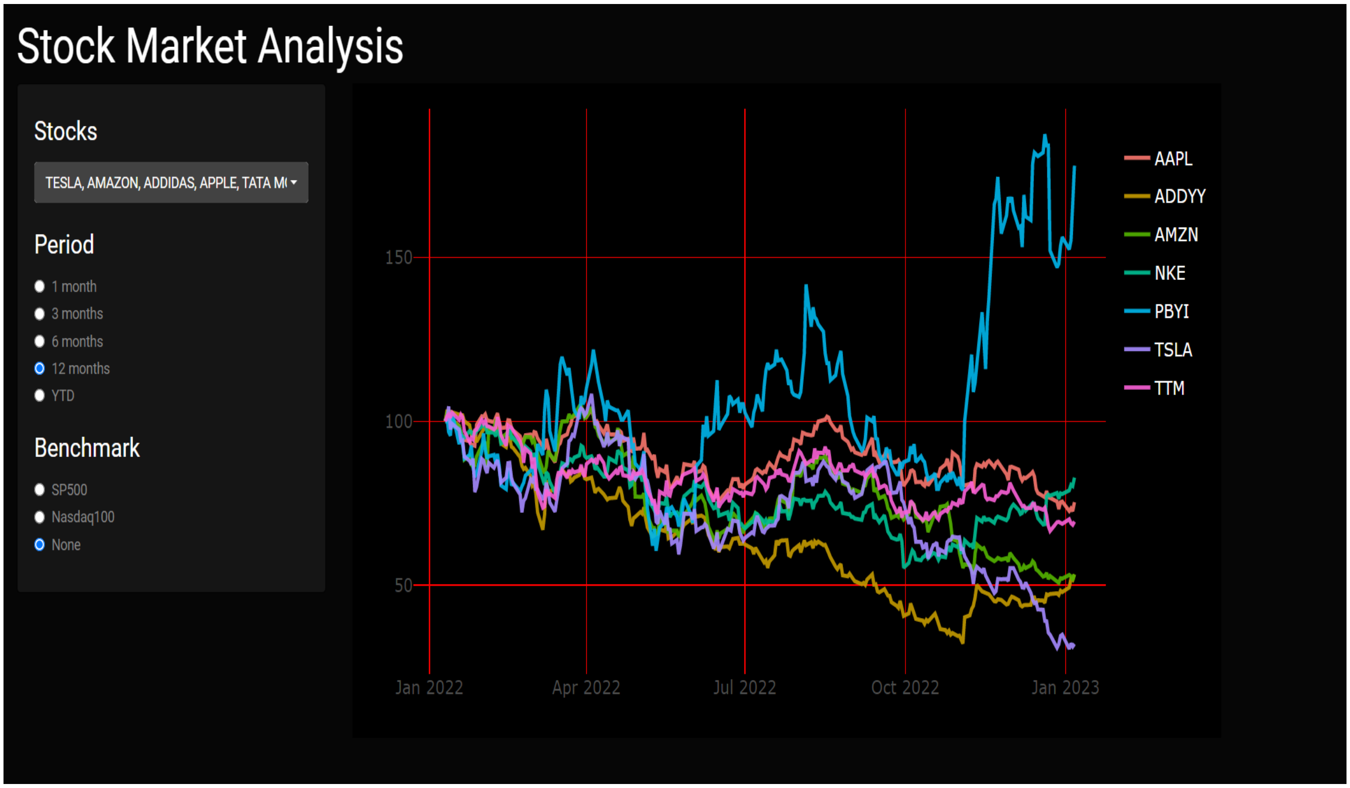
```
 # Create plot
output$plot <- renderPlotly({
  print(
    ggplotly(prices %>%
            group_by(symbol) %>%
            mutate(init_close = if_else(date == min(date),close,NA_real_)) %>%
            mutate(value = round(100 * close / sum(init_close,na.rm=T),1)) %>%
            ungroup() %>%
            ggplot(aes(date, value,color = symbol)) +
            geom_line(size = 1, alpha = .9)+
            # uncomment the line below to show area under curves
            #geom_area(aes(fill=symbol),position="identity",alpha=.2) +
            theme_minimal(base_size=16) +
             theme(axis.title=element_blank(),
                   plot.background = element_rect(fill = "black"),
                   panel.background = element_rect(fill="black"),
                  # panel.grid = element_blank())
                 panel.grid.major = element_line(color = "red", size = 0.5),
                  panel.grid.minor = element_line(color = "red",size=0.5),
            legend.text = element_text(color="white"))
```

To Access the full code please scan the QR code

Or   Visit: Click Here

RESULTS AND DISCUSSIONS

Combining the above codes and modifying some functions we get the following output.



Stock Market Analysis

Stocks

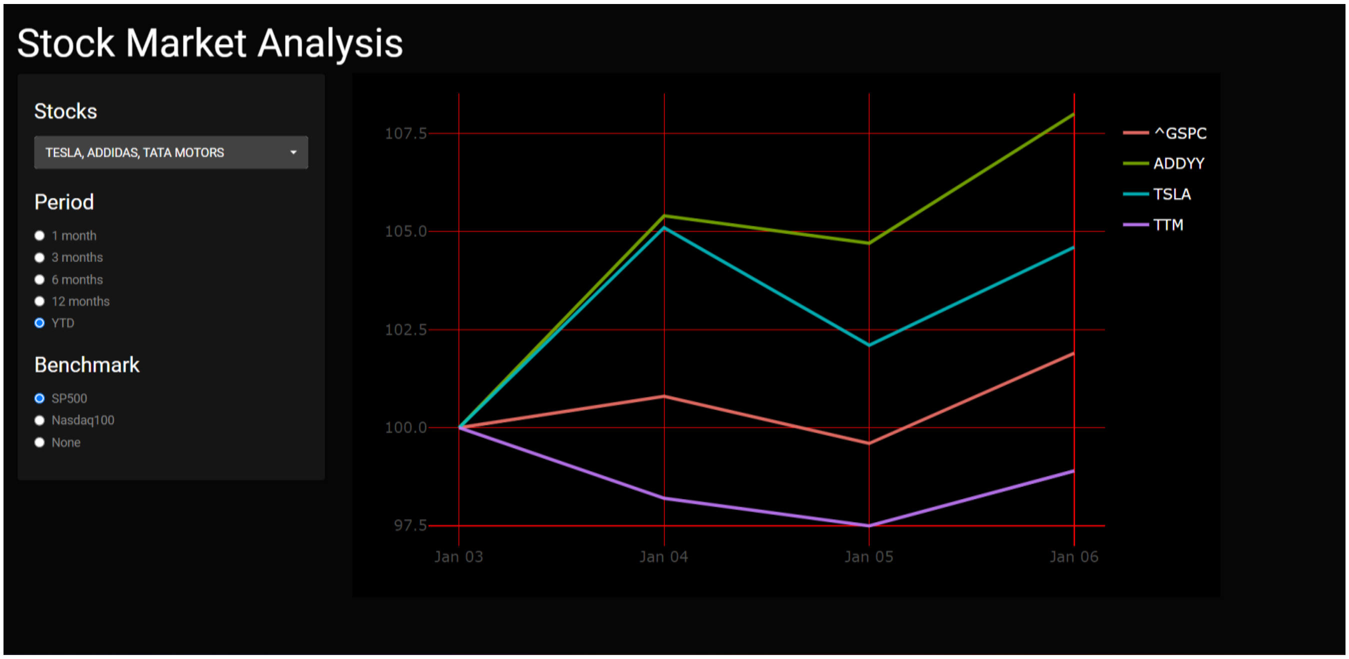
TESLA, AMAZON, ADDIDAS, APPLE, TATA M(

Period

○ 1 month
○ 3 months
○ 6 months
● 12 months
○ YTD

Benchmark

○ SP500
○ Nasdaq100
● None



Stock Market Analysis

Stocks

TESLA, ADDIDAS, TATA MOTORS

Period

○ 1 month
○ 3 months
○ 6 months
○ 12 months
● YTD

Benchmark

● SP500
○ Nasdaq100
○ None

CONCLUSION AND FUTURE SCOPE

The project's goal was to create interactive visualizations to explore stock market data, the conclusion is that we achieved the visualizations and the value of the Shiny web application in facilitating data exploration. Our project provides a simple and interactive way for users to view and compare the performance of selected stocks over a specified time period, with the option to include a benchmark. It can be used by investors, traders and anyone interested in keeping track of the stock market.

Future Scope of our project: -

Our project is a basic model to track stock markets, in the future we can add more interactive features such as creating multiple tabs to showcase multiple analysis.

We could add more visualizations such as candlestick charts, etc.

In the future we could also add Machine Learning models to predict future stock prices

REFERENCES

R Studio : RStudio integrates with R as an IDE to provide further functionality. RStudio combines a source code editor, build automation tools and a debugger.

2.) **Shiny** : Shiny is an R package that makes it easy to build interactive web apps straight from R.(https://shiny.rstudio.com)

3.) **Yahoo Finance** : It's the place for insight-driven investors, financial professionals and business leaders who take their money seriously.(https://finance.yahoo.com)

4.) **GitHub** :GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere. (https://github.com/)

5.) **R Pubs** : Rpubs is an open publishing platform for HTML documents produced using RMarkdown from within RStudio, it provides a simple way to host and share your presentations and documents. Documents stored on RPubs are available publically and allow other to comment on your content. (https://rpubs.com)

6.) **Tutorials Point** : Tutorials point is a dedicated website to provide quality online education in the domains of Computer Science, Information Technology, Programming Languages, and other Engineering as well as Management subjects. (https://www.tutorialspoint.com/index.htm)