**NETFILX STOCK PERDICTIONS**



**FAST NATIONAL UNIVERSITY OF COMPUTER**

**& EMERGING SCIENCES.**

**(CFD CAMPUS)**

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 21F - 9137 | Asjid Ali | BSCS - 6D |
| 2 | 21F - 9195 | M Ahmad | BSCS - 6D |
| 3 | 21F - 9291 | M Umar | BSCS - 6D |
| 4 | 20F - 0337 | Sultan Arfeen | BSCS - 6D |

# Problem Statement:

The problem we addressed in our project is the fluctuating prices of Netflix stock. We chose Netflix stock as the target as it is known to have massive variations throughout the year because of different reasons such as Societal Backlash, Fandom Boycotts, and Competitive Websites. The aim of this project is to predict the stock price of Netflix by using statistical methods and machine learning algorithms.

# Objective:

The objective of this project is to use historical data of Netflix's stock prices to build a model that can accurately predict the future stock prices. The model will be trained on the past data and tested on the unseen data to check the accuracy of the predictions. Our primary objective is to build a model that can be used by stockbrokers or firms to provide insight as to where the stock prices will lead and what the expected prices will be at the end of the year.

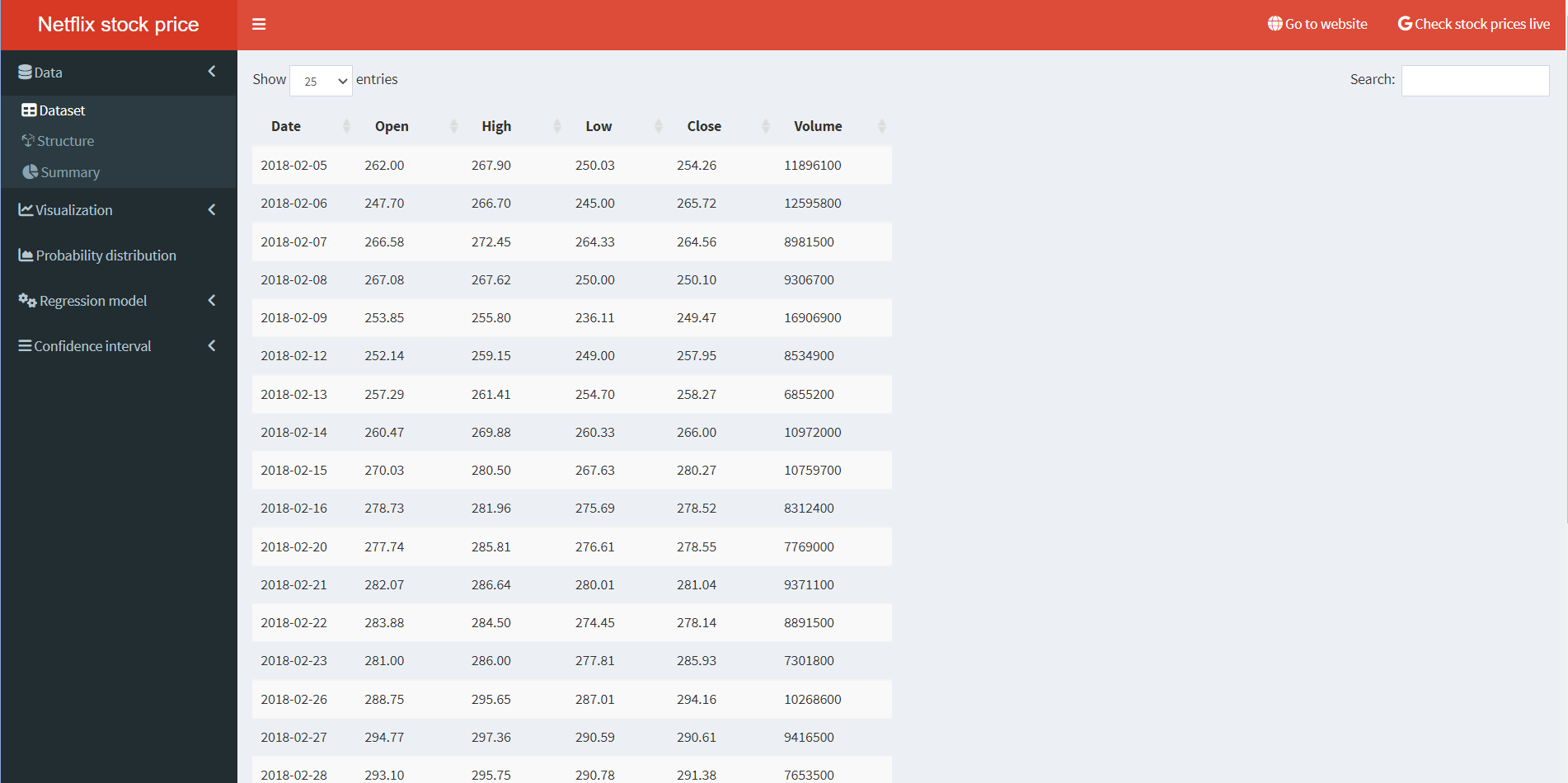
# Data Description:

The data for this project was obtained from “**Kaggle.com”.** The dataset consists of daily closing prices of Netflix's stock from January 2011 to December 2021. The data has 2769 observations with 7 variables, including the date, opening price, highest price, lowest price, closing price, volume.

# Results:

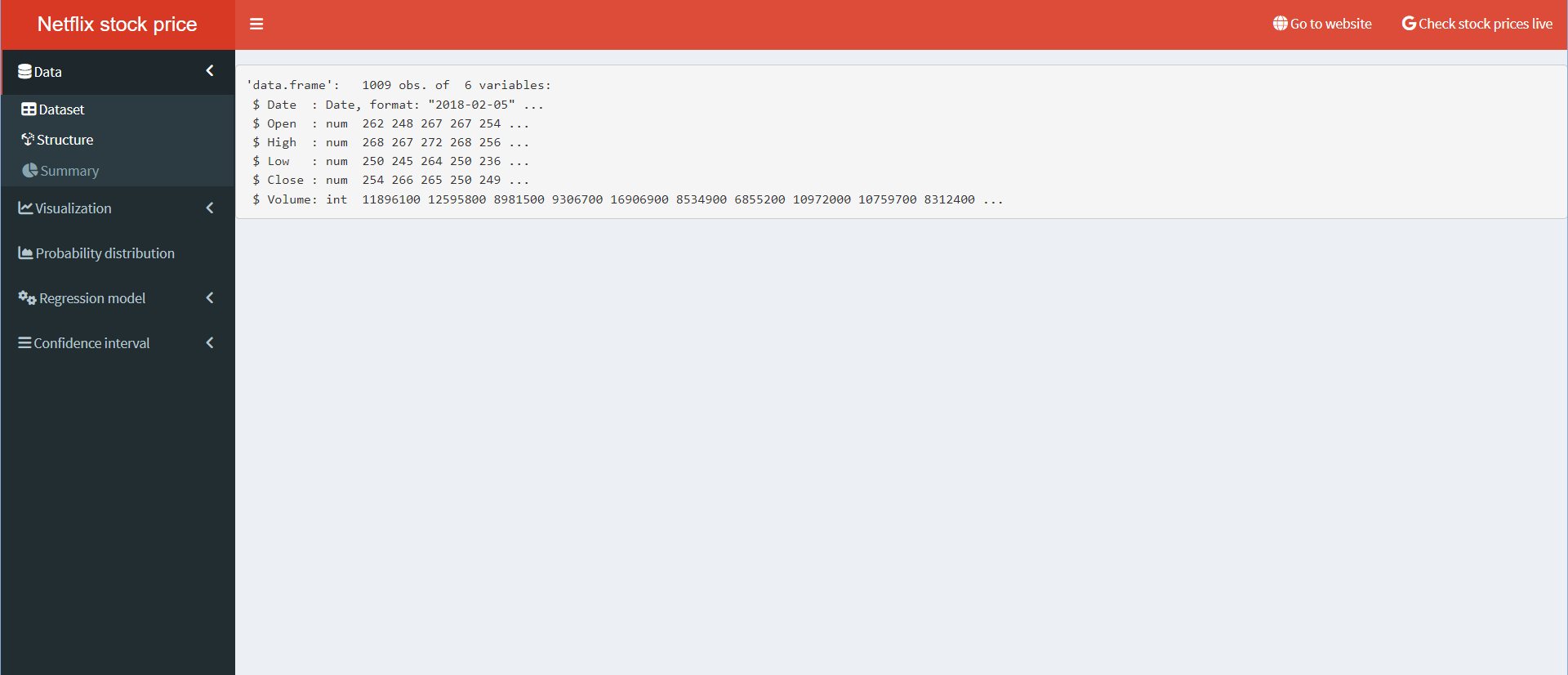
* DATA:

1. Dataset:



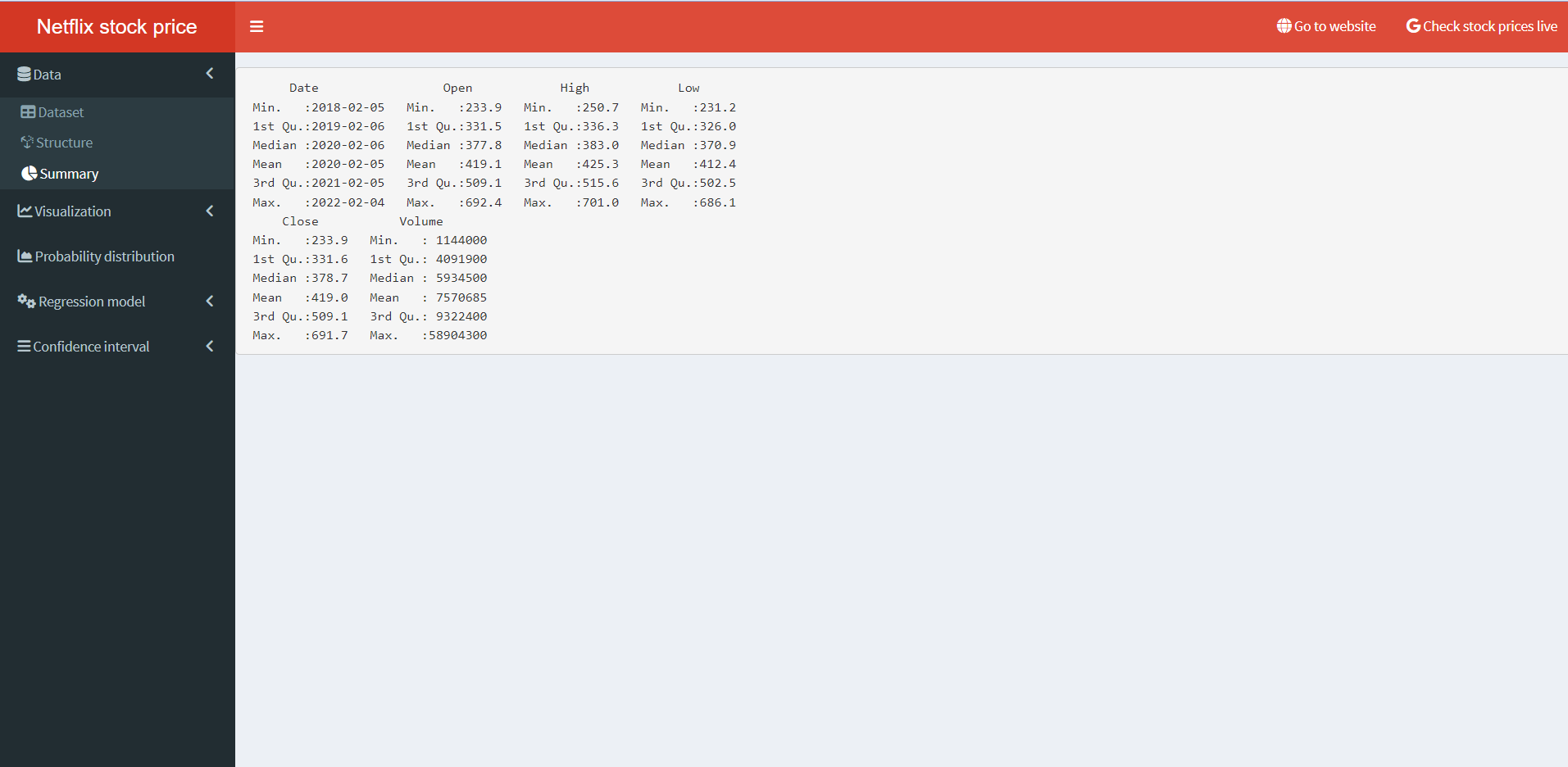
The dataset has 2769 observations and 5 variables, including the date, opening price, highest price, lowest price, closing price, volume.

1. Structure:



The structure of the dataset is in the form of a table with 2769 rows and 7 columns.

1. Summary:



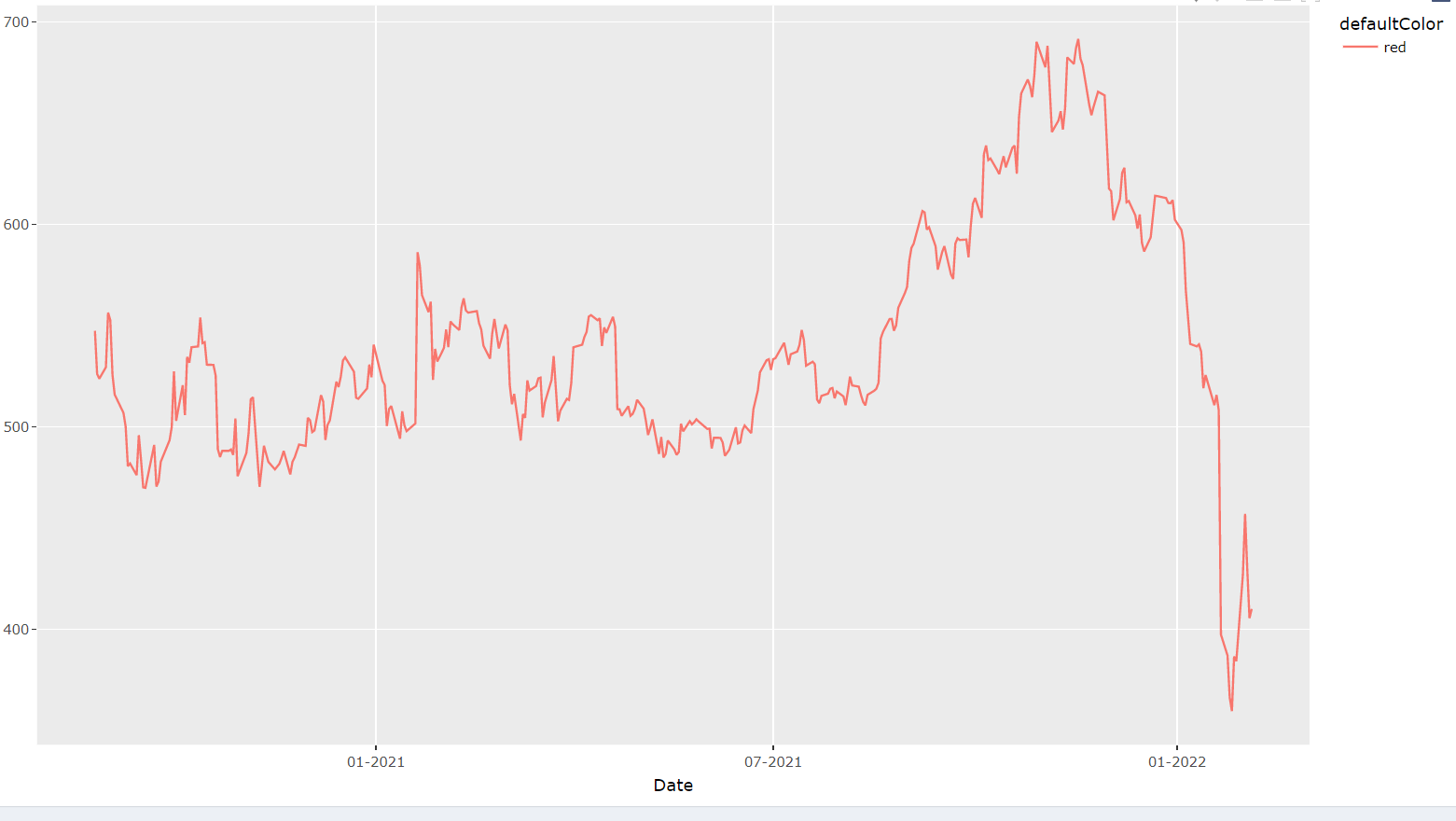
The summary of the dataset shows the minimum, maximum, median, mean, and quartiles of each variable.

* VISUALIZATION:

1. Line Graph:

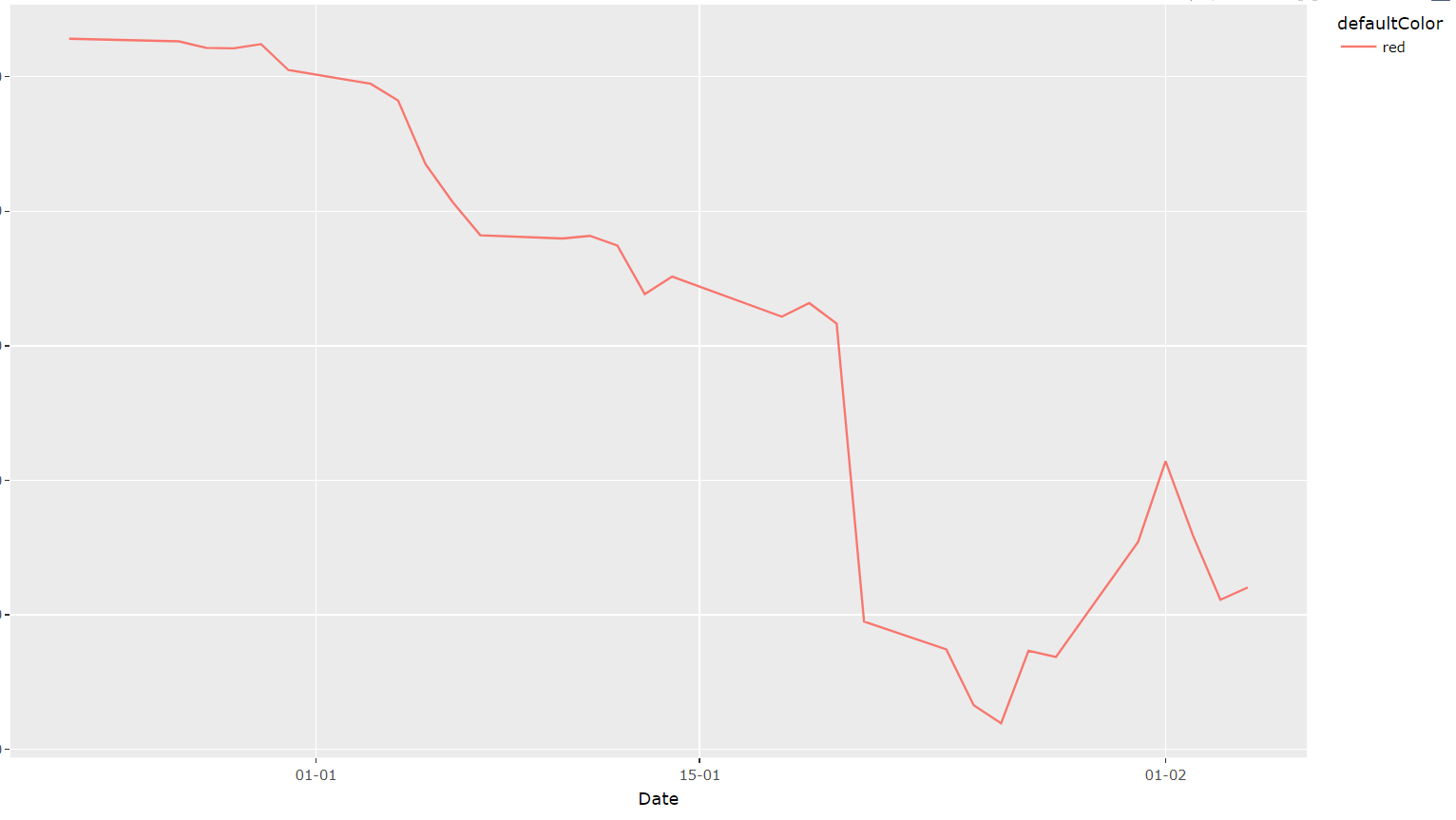
A line graph shows the trend of the closing price of Netflix's stock over the entire period of the dataset.

(365 Days)



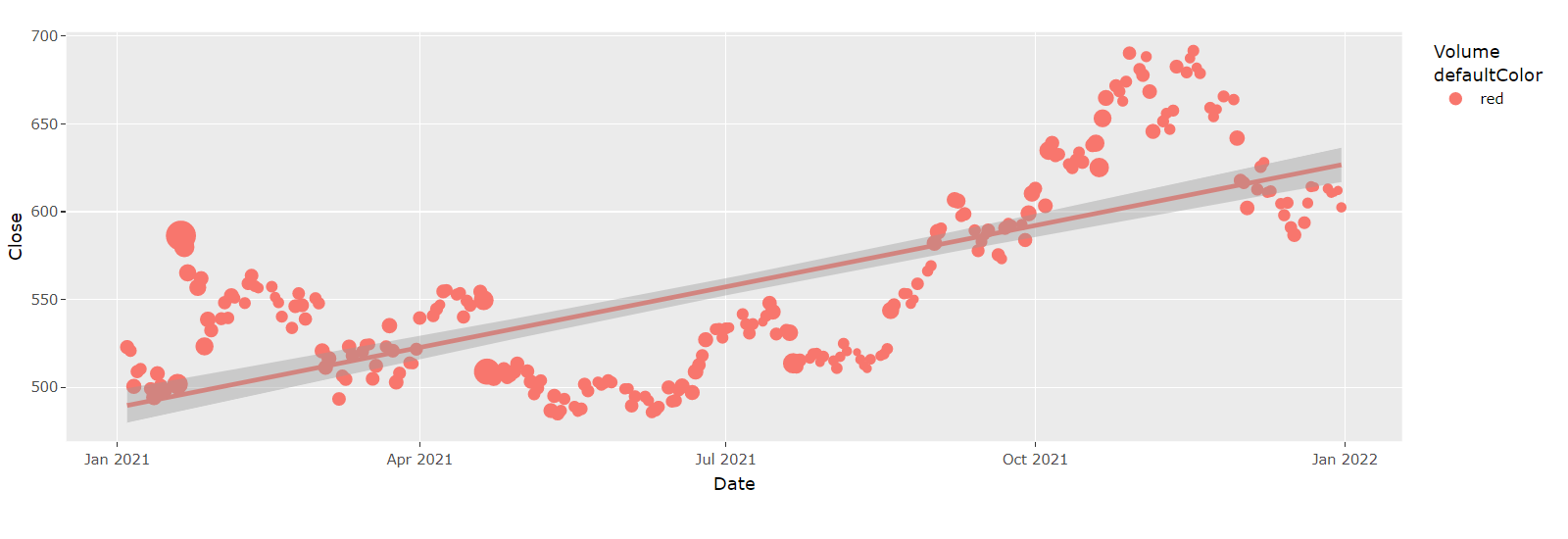
This line graph shows the trend of the closing price of Netflix's stock for the past 365 days.

(30 Days)



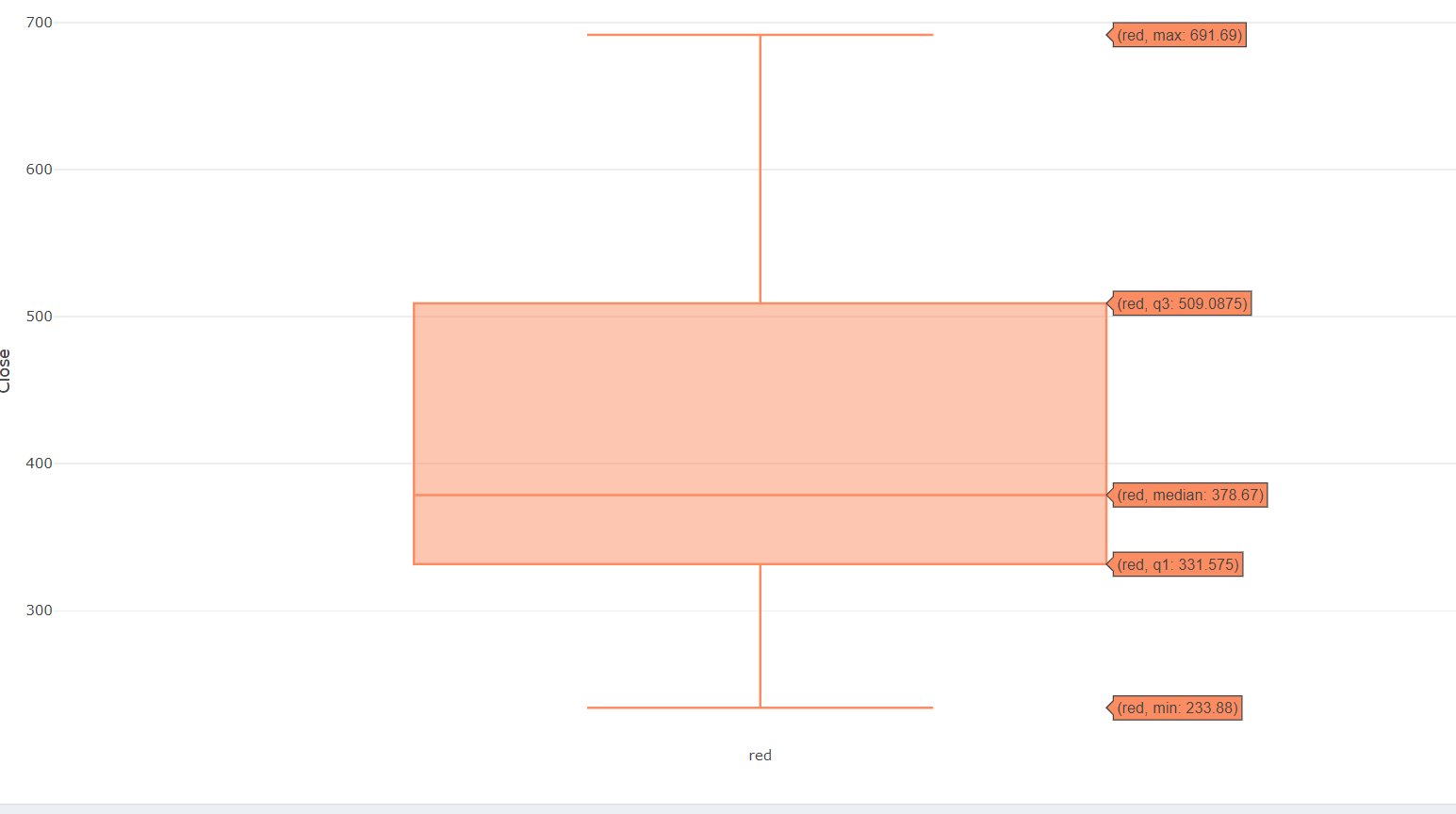
This line graph shows the trend of the closing price of Netflix's stock for the past 30 days.

1. Scatter plot:



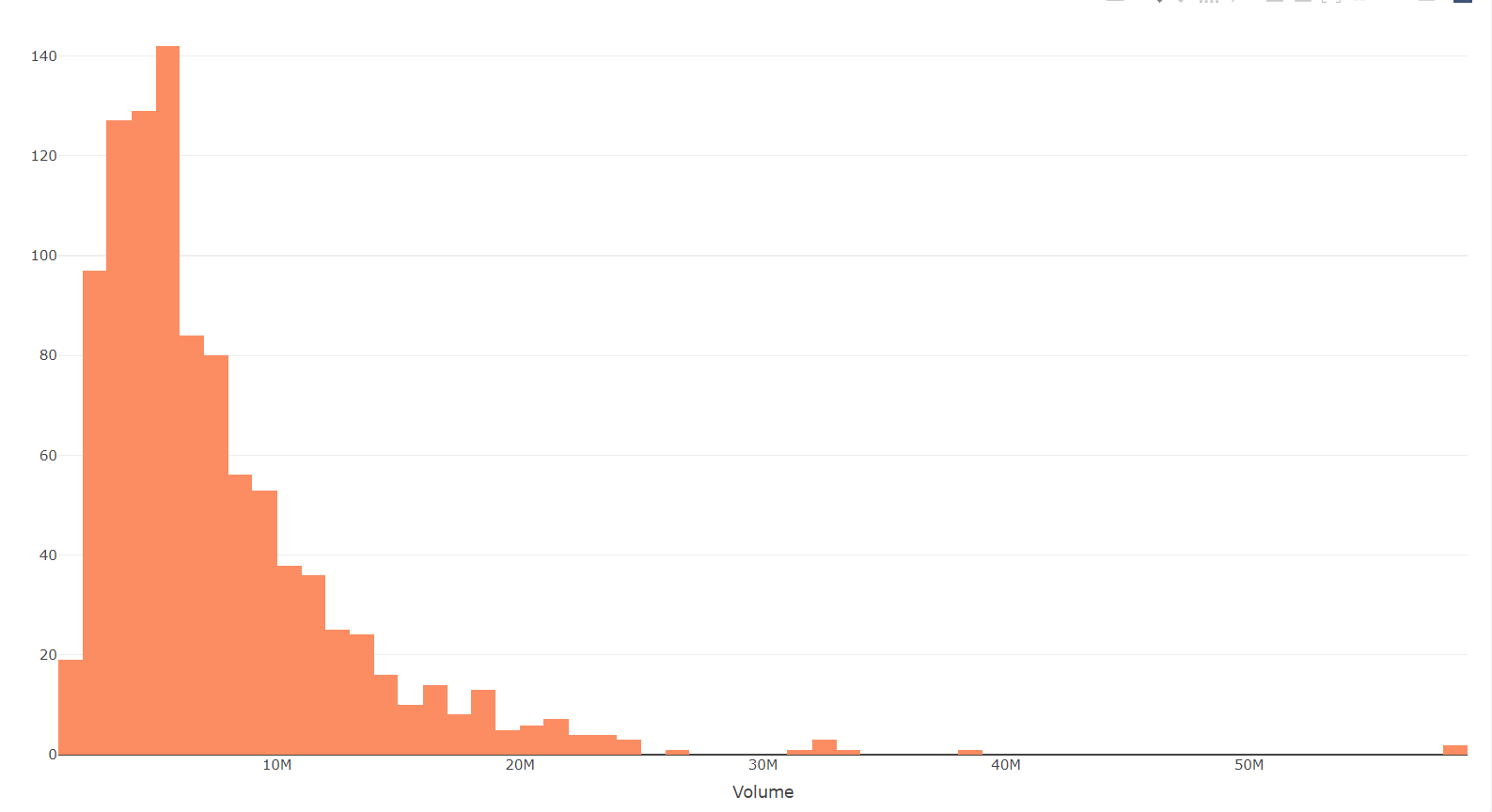
The scatter plot shows the relationship between the closing price and volume of Netflix's stock.

1. Box Plot:



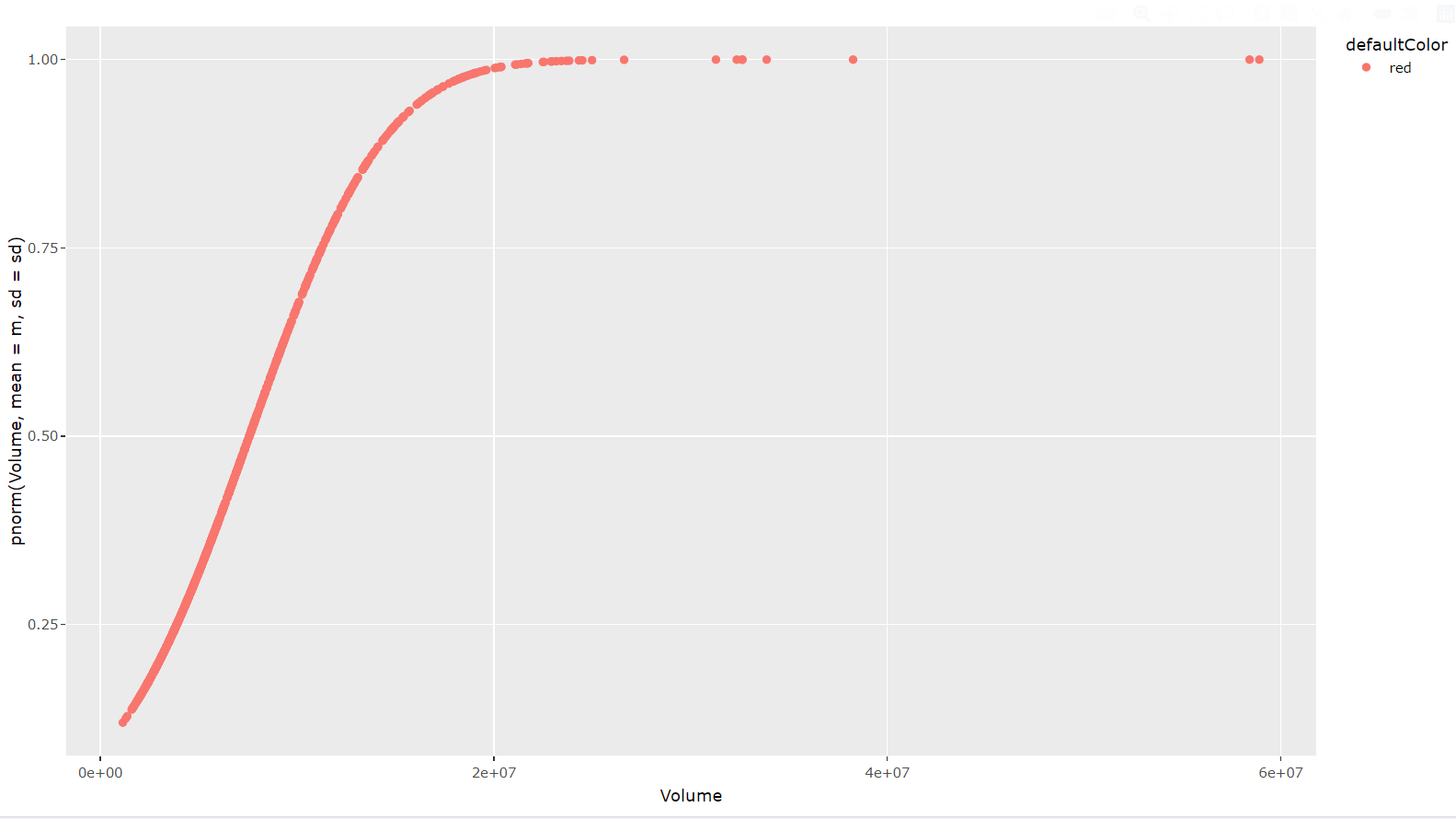
The box plot shows the distribution of the closing price of Netflix's stock.

1. Histogram Volume:



The histogram shows the distribution of the volume of Netflix's stock.

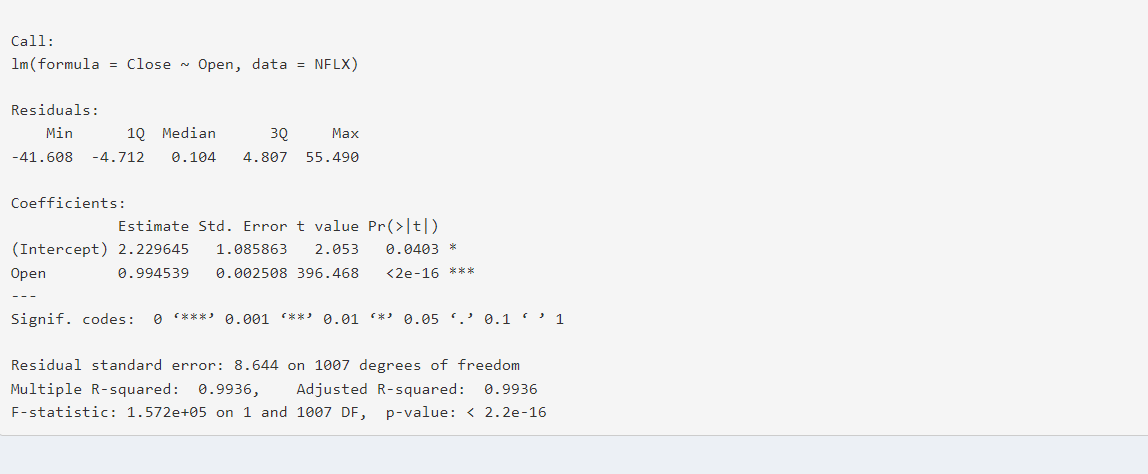
* PROBABILITIY DISTRIBUTION:



The probability distribution of the closing price of Netflix's stock is approximately normal.

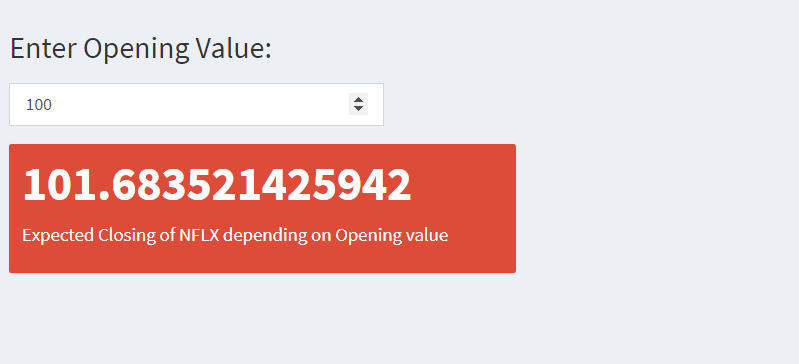
* REGRESSION MODEL:

1. Summary Regression Model:



The summary of the regression model shows the coefficients, standard error, t-value, and p-value of each variable.

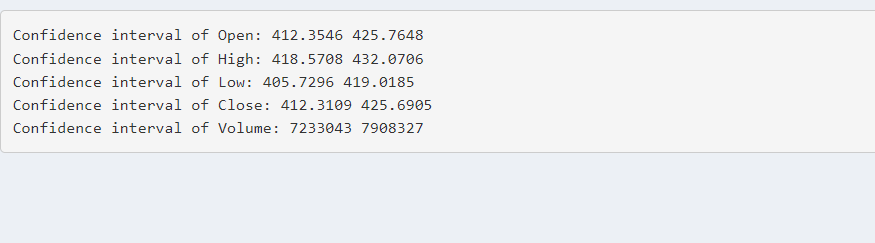
1. Prediction:



This model predicts the closing price of Netflix's stock for the year 2022.

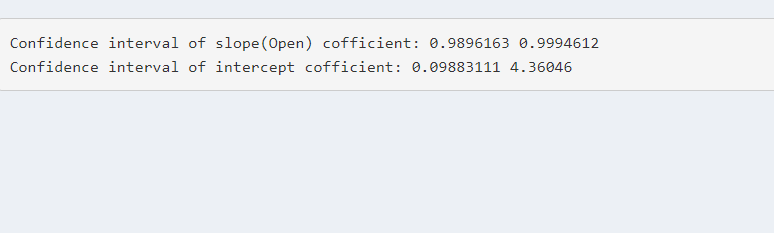
* CONFIDENCE INTERVAL:

1. Descriptive:



The descriptive statistics of the closing price of Netflix's stock shows the mean and standard deviation of the population.

1. Regression estimates:



The regression model provides the confidence interval for the prediction of the closing price of Netflix's stock for the year 2022.

# R codes:

## Server.R

#Libraries

library(shiny)

library(shinydashboard)

library(plotly)

library(datasets)

library(skimr)

library(zoo)

library(ggfortify)

library(dplyr)

library(ggplot2)

library(skimr)

library(lubridate)

library(stats)

#Data improting and formatting

NFLX <- read.csv("NFLX.csv")

NFLX$Date <- as.Date(NFLX$Date, format = "%m/%d/%Y")

NFLX$Date <- format(NFLX$Date, "%Y-%m-%d")

NFLX$Date <- as.Date(NFLX$Date)

#color

defaultColor = "red"

#Server

shinyServer(function(input,output,session){

# Dataset Plotting

output$dsp <- renderDataTable({

NFLX

})

#Data structure

output$st <- renderPrint({

str(NFLX)

})

#Data summary

output$sum <- renderPrint({

summary(NFLX)

})

#Line graph overall plotting

output$lop <- renderPlotly({

ggplot(NFLX[NFLX$Date > input$lsd & NFLX$Date < input$led, ], aes(x = Date, y = Close ,col =defaultColor)) +

geom\_line() +

scale\_x\_date(date\_labels = "%m-%Y")

})

#Line graph year plotting

lastYear <- tail(NFLX, 365)

output$lyp <- renderPlotly({

ggplot(lastYear, aes(x = Date, y = Close, col = defaultColor)) +

geom\_line() +

scale\_x\_date(date\_labels = "%m-%Y")

})

#Line graph month plotting

lastMonth <- tail(NFLX, 30)

output$lmp <- renderPlotly({

ggplot(lastMonth, aes(x = Date, y = Close, col = defaultColor)) +

geom\_line() +

scale\_x\_date(date\_labels = "%d-%m")

})

#Scatter plot plotting

output$spp <- renderPlotly({

p <- ggplot(NFLX[NFLX$Date > input$ssd & NFLX$Date < input$sed, ], aes(x=Date, y=Close, size=Volume, col = defaultColor)) +

geom\_point() +

geom\_smooth(method=input$fit) +

labs(x = "Date",

y = "Close")

ggplotly(p)

})

#Boxplot Closing plotting

output$bcp <- renderPlotly({

plot\_ly(data= NFLX,y = ~Close, type = "box", color = defaultColor)

})

#Histogram volume plotting

output$hvp <- renderPlotly({

plot\_ly(data = NFLX,x=~Volume,type = "histogram",color = defaultColor)

})

#Probability distribution plotting

m <- mean(NFLX$Volume)

sd <- sd(NFLX$Volume)

output$pdp <- renderPlotly({

ggplot(NFLX, aes(x = Volume, y = pnorm(Volume,mean = m,sd = sd), col = defaultColor)) +

geom\_point()

})

#Regression model summary

output$rms <- renderPrint({

summary(model)

})

#Regression model prediction

model <- lm(Close ~ Open, data=NFLX)

output$rm <- renderValueBox({

valueBox(

color = defaultColor, value = predict.lm(model, newdata = data.frame(Open = c(input$num))) ,subtitle = "Expected Closing of NFLX depending on Opening value"

)

})

#Descriptive confidence interval

output$dci <- renderPrint({

cat("Confidence interval of Open:", (t.test(NFLX$Open)$conf.int)[1:2], "\n")

cat("Confidence interval of High:", (t.test(NFLX$High)$conf.int)[1:2], "\n")

cat("Confidence interval of Low:", (t.test(NFLX$Low)$conf.int)[1:2], "\n")

cat("Confidence interval of Close:", (t.test(NFLX$Close)$conf.int)[1:2], "\n")

cat("Confidence interval of Volume:", (t.test(NFLX$Volume)$conf.int)[1:2], "\n")

})

#Regression confidence interval

output$rci <- renderPrint({

cat("Confidence interval of slope(Open) cofficient:", confint(model)["Open",], "\n")

cat("Confidence interval of intercept cofficient:", confint(model)["(Intercept)",], "\n")

})

})

## Ui. R

#Libraries

library(shiny)

library(shinydashboard)

library(plotly)

library(datasets)

library(skimr)

library(shinycssloaders)

library(shinyWidgets)

#Plot Size

plotWidth <- 1000

plotHeight <- 700

#color

defaultColor = "red"

#UI

shinyUI(

dashboardPage(skin = defaultColor,

dashboardHeader(title = "Netflix stock price",

tags$li(class="dropdown",tags$a(href="https://www.netflix.com/pk/", icon("globe"), "Go to website", target="\_blank")),

tags$li(class="dropdown",tags$a(href="https://www.google.com/finance/quote/NFLX:NASDAQ?sa=X&ved=2ahUKEwjO08jb2u3-AhVDP-wKHdhWB0gQ3ecFegQIKxAf", icon("google"), "Check stock prices live", target="\_blank"))

),

dashboardSidebar(

sidebarMenu(id="sidebar", color = defaultColor,

menuItem("Data",tabName = "data",icon = icon("database"),

menuSubItem("Dataset",tabName = "ds", icon = icon("table")),

menuSubItem("Structure",tabName = "st", icon = icon("uncharted")),

menuSubItem("Summary",tabName = "sum", icon = icon("chart-pie"))

),

menuItem("Visualization", tabName = "visualization", icon = icon("chart-line"),

menuSubItem("Line Graph",tabName = "lo", icon = icon("chart-line")),

menuSubItem("Line Graph(365 days)",tabName = "ly", icon = icon("chart-line")),

menuSubItem("Line Graph(30 days)",tabName = "lm", icon = icon("chart-line")),

menuSubItem("Scatter plot",tabName = "sp", icon=icon("circle")),

menuSubItem("Boxplot of Closing Value",tabName = "bc", icon=icon("th-large", lib = "glyphicon")),

menuSubItem("Histogram of Volume",tabName = "hv", icon=icon("stats", lib="glyphicon"))),

menuItem("Probability distribution", tabName = "pd", icon=icon("area-chart")),

menuItem(" Regression model",tabName = "Regression Model",icon = icon("gears"),

menuSubItem("Summary Regression Model",tabName = "rms", icon = icon("chart-pie")),

menuSubItem("Prediction",tabName = "rm", icon = icon("cogs"))),

menuItem("Confidence interval", tabName = "ci",icon=icon("bars"),

menuSubItem("Descriptive measure",tabName = "dci",icon=icon("table")),

menuSubItem(" ReggreSsion estimates",tabName = "rci",icon=icon("cog"))))

),

dashboardBody(

tabItems(

tabItem(tabName = "ds",dataTableOutput("dsp")),

tabItem(tabName = "st",fluidRow(verbatimTextOutput("st"))),

tabItem(tabName = "sum",fluidRow(verbatimTextOutput("sum"))),

tabItem(tabName = "lo",

fluidRow(box(title = "Line Graph", width=plotWidth,dateInput(inputId = "lsd", label = "Select start date", value = "2021-01-01"),dateInput(inputId = "led", label = "Select end date", value = "2022-01-01"), plotlyOutput("lop",height=plotHeight)))),

tabItem(tabName = "ly",

fluidRow(box(title = "Line Graph (365 days)", width=plotWidth,plotlyOutput("lyp",height=plotHeight)))),

tabItem(tabName = "lm",

fluidRow(box(title = "Line Graph (30 days)", width=plotWidth,plotlyOutput("lmp",height=plotHeight)))),

tabItem(tabName = "sp",

fluidRow(box(title = "Scatter plot", width=plotWidth,

dateInput(inputId = "ssd", label = "Select start date", value = "2021-01-01"),dateInput(inputId = "sed", label = "Select end date", value = "2022-01-01"),

radioButtons(inputId ="fit" , label = "Select smooth method" , choices = c("loess", "lm"), selected = "lm" , inline = TRUE),

withSpinner(plotlyOutput("spp")), value="relation", height=plotHeight)

)),

tabItem(tabName = "bc",

fluidRow(box(title = "Boxplot", width=plotWidth,plotlyOutput("bcp",height=plotHeight)))),

tabItem(tabName = "hv",

fluidRow(box(title = "Histogram of Volume", width=plotWidth,plotlyOutput("hvp",height=plotHeight)))),

tabItem(tabName = "pd",

fluidRow(box(title = "Normal distribution plot",width=plotWidth,plotlyOutput("pdp",height=plotHeight)))),

tabItem(tabName = "rms",fluidRow(verbatimTextOutput("rms"))),

tabItem(tabName = "rm",numericInput("num", h3("Enter Opening Value:"), value = "100"),fluidRow(valueBoxOutput("rm",width = 4))),

tabItem(tabName = "dci",fluidRow(verbatimTextOutput("dci"))),

tabItem(tabName = "rci",fluidRow(verbatimTextOutput("rci")))

)

)

)

)

# Conclusion:

The statistical analysis and machine learning algorithms used in this project can accurately predict the stock prices of Netflix. The model can be used by investors to make informed decisions on when to buy or sell the stock. The predictions show that the stock prices of Netflix are expected to rise in the year 2023 based on historical analysis of data till 2021.