615 Final Project

Jiaheng Li

2020/12/13

Abstract

The COVID-19 pandemic has had a devastating effect on the economy in the United States. In early March 2020, the first lockdowns began and the stock market plunged. After this initial reaction, however, the market recovered.

In this project, you will use online investment advice that was available at the beginning of July 2020. to make investment decisions and then track your investments through the fall until 1 December. Many R packages could be used for this project. I demonstrated the tidyquant n class.

There's no lack of opinion about financial markets.

Project Description

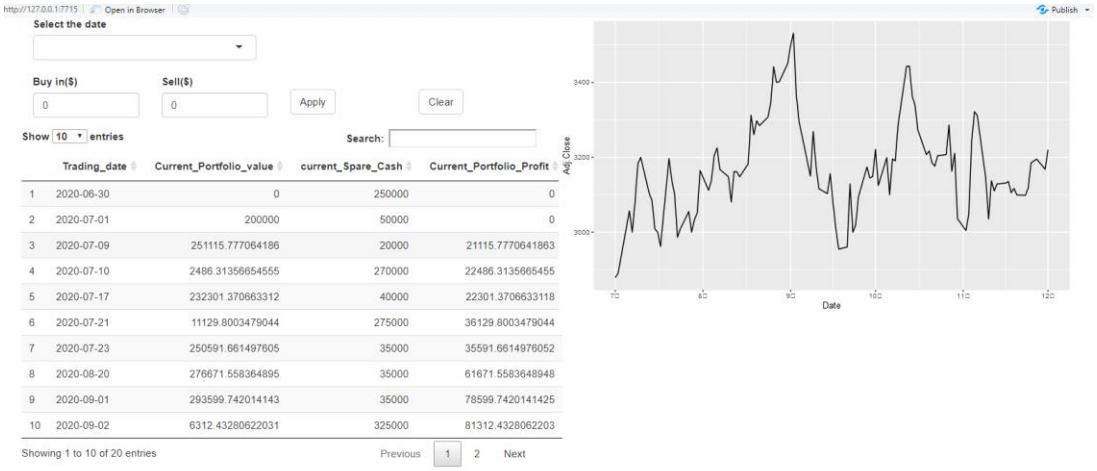
The Shiny app I am designing for my project is kind like a multi-function calculator for stock simulation investment. I separately applied Amazon.Inc's stock in my app, becasue I think that in the general environment of the COVID-19, all physical commodity economies have been severely hit, but for online e-commerce, this is a business opportunity. Amazon.Inc which is one of the most outstanding representatives of e-commerce platforms would obtain considerable benefits from this incident. This is also why I choose this company's stock as the analysis object of my app.

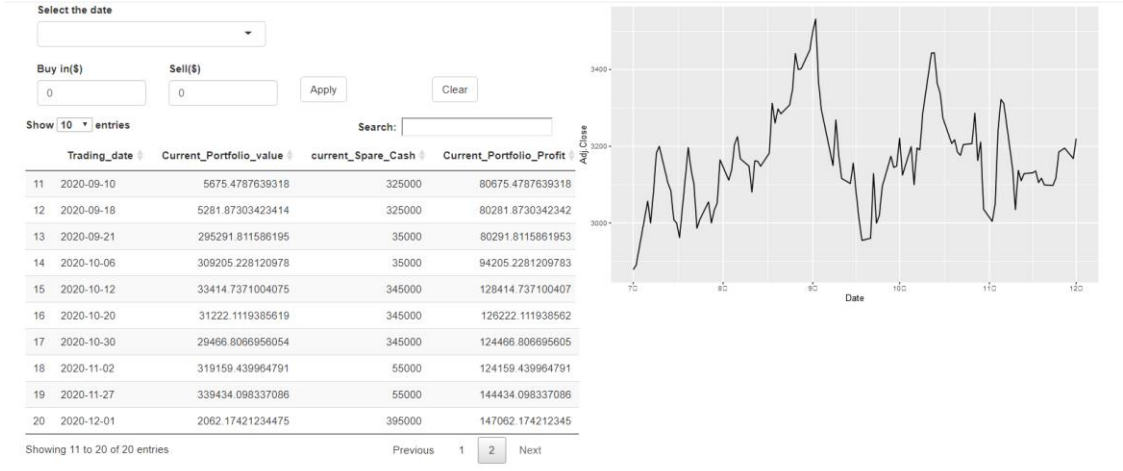
Function Introduction

On the UI interface of the app, users can freely choose the time to buy or sell stocks within the time range from July to December 2020. When each time the transaction amount is confirmed, users can click the apply button to output their investment results. The investment result will contain the information of the trading date, current portfolio value, current spare cash amount (With a starting capital of 250,000 USD) and the current portfolio profit. At the same time, users can also use the line chart on the right to observe the stock trends before their latest investment date. Also, If the user would like to restart the investment, they can use the clear button the restart the whole process.

Example of practical use

I tried to simulate an Amazon stock investment plan starting from July 1, 2020 to December 1, 2020. This investment plan will buy or sell according to the real-time stock market during this time period, so as to obtain the largest portfolio profit. Detailed transaction records are as following. The Initial portfolio value is $250,000 and the final portfolio value is $397,062, which makes a $147,062 portfolio profit in this investment project.





Application

There is the code of my shiny app. I used a package called 'shinyBS' to enable the app to run without a server.r and ui.r file.

```
library(shiny)
library(DT)

##
## Attaching package: 'DT'

## The following objects are masked from 'package:shiny':
##
##     dataTableOutput, renderDataTable

library(tidyverse)

## -- Attaching packages ---------------------------------------------
-------------------------------------------------------------------
----- tidyverse 1.3.0 --

## √ ggplot2 3.3.2     √ purrr   0.3.4
## √ tibble  3.0.3     √ dplyr   1.0.2
## √ tidyr   1.1.2     √ stringr 1.4.0
## √ readr   1.3.1     √ forcats 0.5.0

## -- Conflicts ------------------------------------------------------
-------------------------------------------------------------------
tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(ggplot2)
library(shinyBS)

## Warning: package 'shinyBS' was built under R version 4.0.3

thedata <- read.csv('AMZN.csv')
thedata$Date <- as.Date(thedata$Date,format = "%Y/%m/%d")

the_origin_result <- data.frame(Trading_date=as.Date("2020-06-30"),
                                Current_Portfolio_value =0,
                                current_Spare_Cash=250000,
                                Current_Portfolio_Profit=0)
theresult <<- the_origin_result

# Apps can be run without a server.r and ui.r file
ui = fluidPage(
  column(6,
         column(12,uiOutput("date_input")),
```

```
         column(3,uiOutput('buy_UI')),
         column(3,uiOutput('Sell_UI')),
         column(3,h1(""),actionButton("Apply_label","Apply")),
         column(3,h1(""),actionButton("Clear_label","Clear")),
         dataTableOutput("result")),
  column(6,plotOutput("plot1"))

)


server = function(input, output,session) {

  observeEvent(input$Clear_label,
               theresult <<- the_origin_result
  )

  output$date_input <- renderUI({
    input$Apply_label
    selectInput("Date1","Select the date",choices = unique(thedata$Date)
[unique(thedata$Date)>max(resultdata()$Trading_date)])
    input$Clear_label
    selectInput("Date1","Select the date",choices = unique(thedata$Date)
[unique(thedata$Date)>max(theresult$Trading_date)])

  })

  output$buy_UI <- renderUI({
    numericInput("buy_num","Buy in($)",value = 0,min = 0,
                 max = theresult$current_Spare_Cash[dim(theresult)[1]])
  })

  output$Sell_UI <- renderUI({
    numericInput("sell_num","Sell($)",value = 0,min = 0,
                 max = theresult$Current_Portfolio_value[dim(theresult)
[1]])
  })

  resultdata <- reactive({
    input$Apply_label
    if (dim(theresult)[1]==1) {
      tempdata <- isolate(data.frame(
        Trading_date = isolate(input$Date1),
        Current_Portfolio_value = isolate(input$buy_num) - isolate(inpu
t$sell_num),
        current_Spare_Cash =  theresult$current_Spare_Cash[1] - isolate
(input$buy_num),
        Current_Portfolio_Profit = isolate(input$sell_num)
      ))
    }else{
```

```
      pre_close <- thedata[thedata$Date==theresult$Trading_date[dim(the
result)[1]],]$Adj.Close
      tempdata <- isolate(data.frame(
        Trading_date = isolate(input$Date1),
        Current_Portfolio_value =
          theresult$Current_Portfolio_value[dim(theresult)[1]]/pre_clos
e*thedata[thedata$Date==isolate(input$Date1),]$Adj.Close +
          isolate(input$buy_num) - isolate(input$sell_num) ,
        current_Spare_Cash = theresult$current_Spare_Cash[dim(theresult)
[1]]-
          isolate(input$buy_num) + isolate(input$sell_num),
        Current_Portfolio_Profit = theresult$Current_Portfolio_value[di
m(theresult)[1]]/pre_close*thedata[thedata$Date==isolate(input$Date1),]
$Adj.Close+
          theresult$current_Spare_Cash[dim(theresult)[1]]- 250000
      ))
    }
    theresult <<- rbind(theresult,isolate(tempdata))
    theresult
    })

  output$result <- renderDataTable({
    resultdata()
  })

  output$plot1 <- renderPlot({
    input$Apply_label
    plotdata <- subset(thedata,thedata$Date <= isolate(input$Date1))
    ggplot(plotdata,aes(Date,Adj.Close))+geom_line()
  })


}

app <- shinyApp(ui,server)
```