

Explicacion aplicacion

Willmer

25/10/2020

TABLERO PRUEBA ANALISTA DE INVERSION

El Siguiente documento pretende ilustrar una presentación básica de cómo está construido el tablero en shiny.

El tablero está dividido en tres partes

1. El procesamiento de los datos: Que abarca las descargas automáticas de los datos y la respectiva depuración y preparacion de los datos
2. El diseño de la interfaz de usuario
3. Las acciones del servidor de R haciendo reactivo cada uno de sus elementos

Procesamiento de los datos

La descarga de los datos se realiza directamente desde la web la TRM mediante servicios expuestos que tiene el gobierno nacional a través socrata y el portal www.datos.gov.co. El precio del petróleo igualmente mediante los servicios expuesto en la página kapsarc.org Y los datos de las acciones se descargan automáticamente mediante la librería BatchGetSymbols

```
# Ejemplo de descarga y procesamiento  
library(RSocrata)
```

```
## Warning: package 'RSocrata' was built under R version 3.6.3
```

```
TRM2 <- read.socrata("https://www.datos.gov.co/resource/ceyp-9c7c.csv")  
oil2 = read.csv("https://datasource.kapsarc.org/explore/dataset/opec-crude-oil-price/download/?format=csv&timezone=America/Bogota&lang=en&use_labels_for_header=true&csv_separator=%3B" ,  
               sep = ";")  
oil <- oil2  
TRM <- TRM2  
oil$Date <- as.POSIXct(oil$Date, format = "%Y-%m-%d")  
TRM <- TRM[order(as.Date(TRM$vigenciadesde, format="%Y-%m-%d")),]  
oil <- oil[order(as.Date(oil$Date, format = "%Y-%m-%d")),]  
oil <- oil[oil$Date >= '2008-01-01',]  
TRM <- TRM[TRM$vigenciadesde >= '2008-01-01',]  
copvsoil <- merge(x = TRM , y = oil , by.x = "vigenciadesde" , by.y = "Date", all = FALSE )  
copvsoil$año <- format(copvsoil$vigenciadesde , format = "%Y")  
copvsoil$mes <- format(copvsoil$vigenciadesde , format = "%Y-%m")  
copvsoil$mesnum <- format(copvsoil$vigenciadesde , format = "%m")  
  
head(copvsoil)
```

##	vigenciadesde	valor	vigenciahasta	Crude.Oil.Price	año	mes	mesnum
## 1	2008-01-03	2012.82	2008-01-03	93.67	2008	2008-01	01
## 2	2008-01-04	2013.27	2008-01-04	93.38	2008	2008-01	01
## 3	2008-01-09	2000.91	2008-01-09	91.98	2008	2008-01	01
## 4	2008-01-10	2004.70	2008-01-10	89.71	2008	2008-01	01
## 5	2008-01-11	2003.74	2008-01-11	88.36	2008	2008-01	01
## 6	2008-01-15	1949.43	2008-01-15	88.33	2008	2008-01	01

Luego de tener los datos financieros y hacer las transformaciones necesarias que pueden encontrar mas detallado en el codigo de la aplicacion nos disponer a plantear la interfaz grafica.

2. El diseño de la interfaz de usuario

La pagina fué creado de forma fluid page con dos viñetas parte uno y parte dos para dar respuestas a las dos primeras partes de la prueba el tema elegido es “cerulean” y se trabajó una malla de tal forma que pudieramos obtener el fluidRow dividido en dos grandes renglones de ubicamos 2 y 3 plot respectivamente .

En la parte superior ubicamos los respectivos inputs del modelo Indicador(petroleo,TRM), Año y mes

Al hacer clic en el nos muestra los cambios dinamicos el modelo es para todas las graficas el año es para generar las graficas discretizadas del boxplot, el scater plot y la grafica del mes y el mes para ver la linea de graficas del mes.

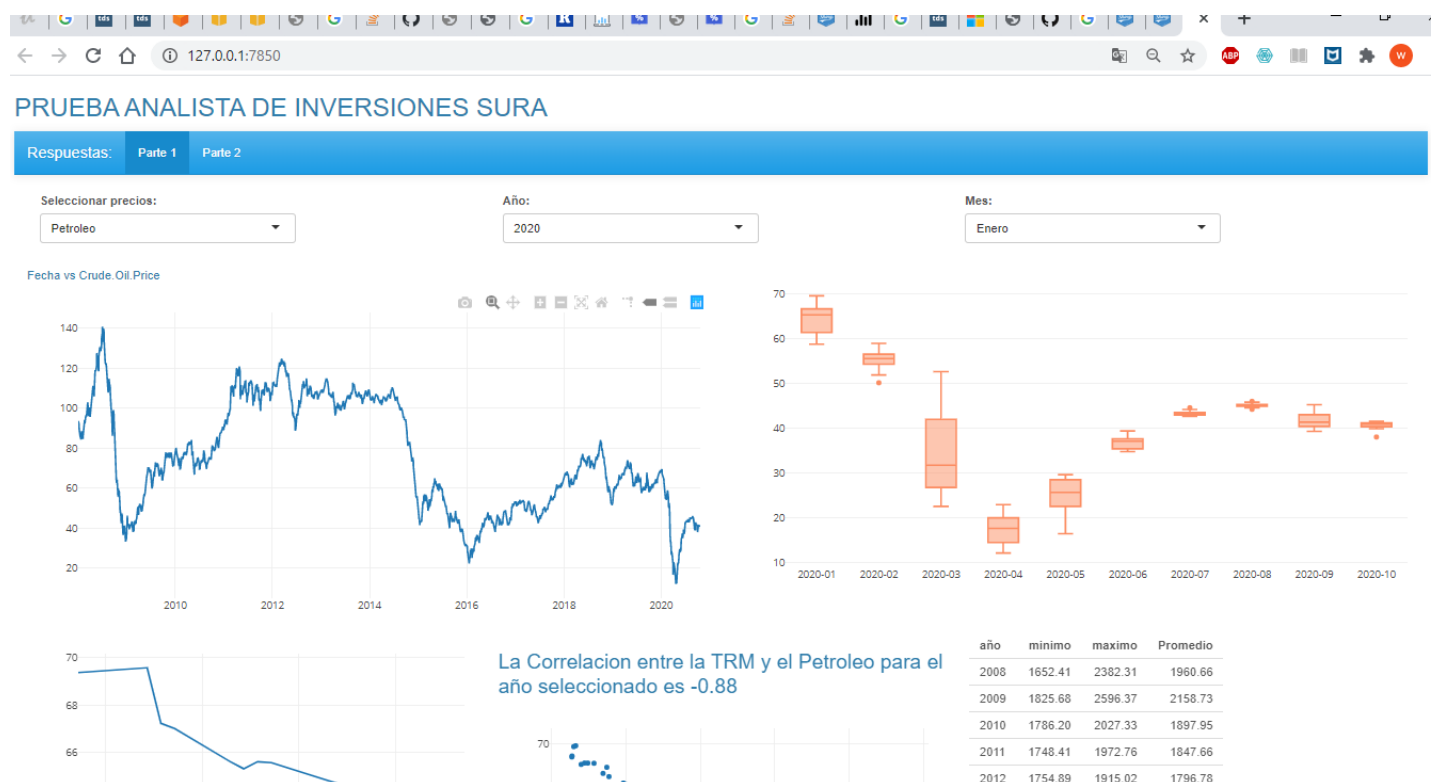


Imagen general

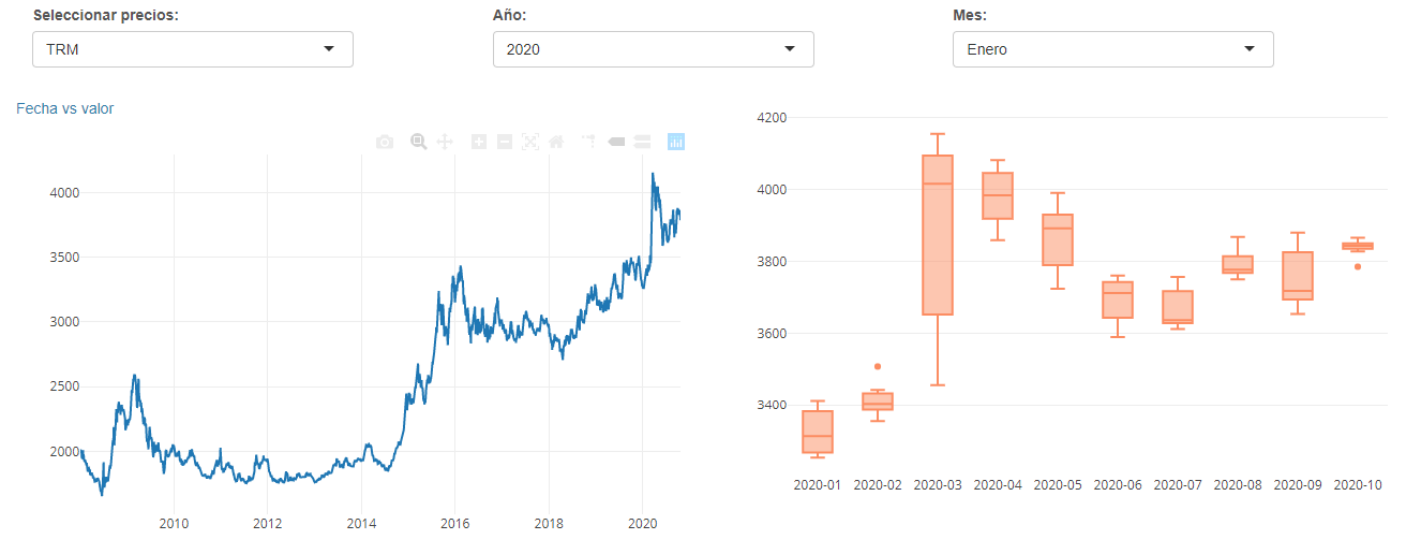
De esta forma damos respuesta a las pregunta dadas en los numerales

PRUEBA ANALISTA DE INVERSIONES SURA

Respuestas:

Parte 1

Parte 2

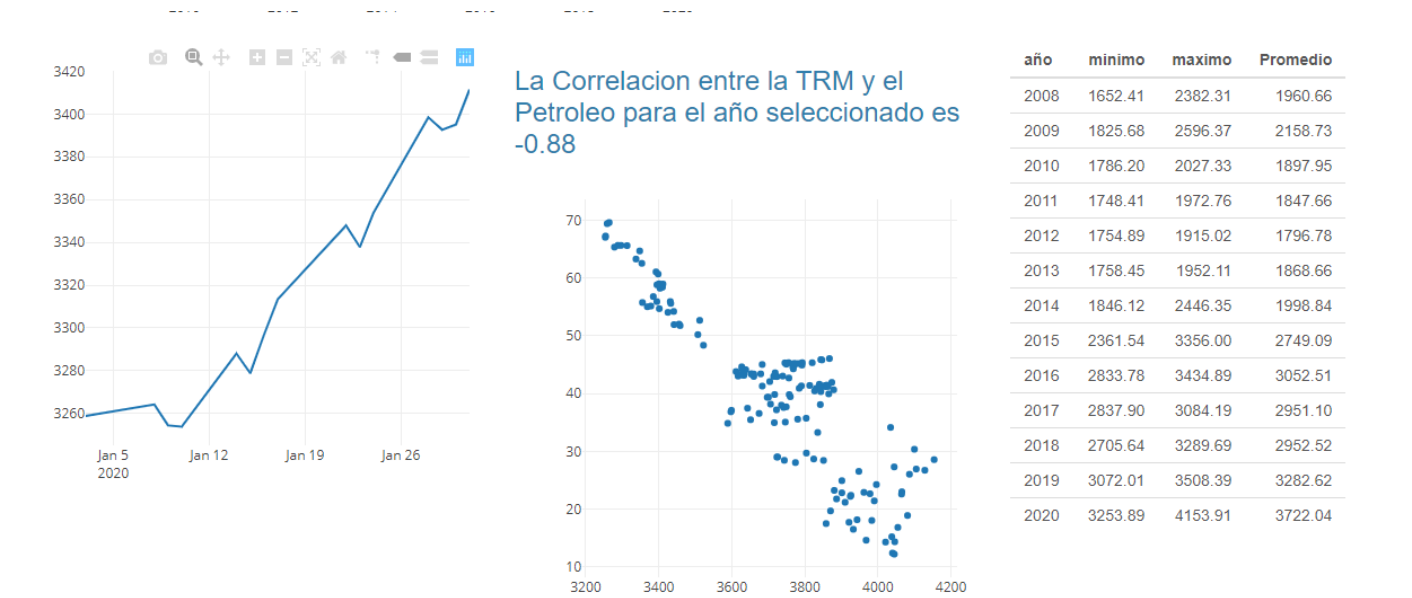


Fecha vs valor





Imagen general



Y mediante el navebar podremos seleccionar le parte 2 donde seleccionamos algunas de las acciones mas importantes de S&P500

PRUEBA ANALISTA DE INVERSIONES SURA

Respuestas: Parte 1 Parte 2

Seleccione una acción:

FB

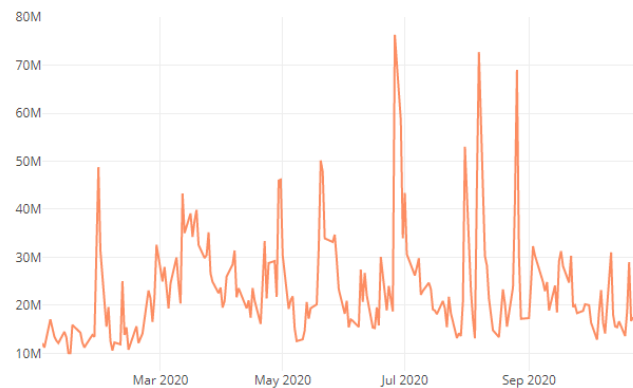
Año:

2020

Precios de cierre por año de la accion FB



Volumen



##3. Las acciones del servidor

En este paso configuramos todos nuestros input reactivos configuramos nuestras graficas con la libreria seleccionada para hacer las graficas que fue plotly ejemplo:

```
bolsa <- reactive({ financiero_data[(financiero_data$ticker == input$accion) & (financiero_data$
año == input$year),]
```

```
output$finanza <- renderPlotly({ plot_ly(bolsa(), y = bolsas$price.close, x = bolsas$ref.date, type = 'scatter', mode =
'lines')
```

```
})
```

En este ejemplo de código que se repite muchas veces seleccionamos un dataframe configurados por nuestros input y lo guardamos con la función `bolsa()` luego renderizamos nuestra gráfica en plotly con el dataframe que se va creando a medida que el usuario marque cada una de las opciones disponible en este caso acciones y año