**Proyecto Final**

Jorge Rubio Ocegueda

Alejandro Mercado Cuevas

Karen Arana Villalobos

Esmeralda Arriaga Hernández

El programa que se realizó es una aplicación de portafolios de inversión para principiantes, en donde se puede elegir de 1 a 5 acciones dentro del IPC y se desglosa información de la empresa e industria, 3 portafolios óptimos (máximo rendimiento, mínima varianza y máximo sharpe ratio) de las acciones elegidas por el usuario y otros 3 portafolios óptimos con las acciones con rendimiento positivo y menor correlación entre ellos y por último se hace un pronóstico a tres meses de ambos portafolios.

El link de la aplicación es: https://esmearriaga.shinyapps.io/portafolio/

Se decidió hacer el reporte en Word porque es complicado hacer un markdown con Shiny App. El funcionamiento de Shiny se divide en dos partes, una parte que ve el usuario y lo que se realiza detrás. Para comenzar se descarga las paqueterías necesarias y se crea un vector con las acciones a elegir por el usuario.

library(shiny)

library(quantmod)

library(its)

library(PortfolioAnalytics)

library(forecast)

library(plotly)

library(timeSeries)

#Acciones dentro del ipc

acciones<-c('No aplica'="",'Alfa'='ALFAA.MX','Alpek'='ALPEKA.MX','Alsea'='ALSEA.MX',

'America Movil'='AMXL.MX','Grupo Aeroportuario del Sureste'='ASURB.MX',

'Grupo Financiero Banregio'='GFREGIOO.MX','Becle Cuervo'='CUERVO.MX',

'Grupo Bimbo'='BIMBOA.MX','Cemex'='CEMEXCPO.MX','Coca-Cola Femsa'='KOFL.MX',

'Grupo Elektra'='ELEKTRA.MX','Grupo Financiero Banorte'='GFNORTEO.MX',

'Genomma Lab Internacional'='LABB.MX','Gentera'='GENTERA.MX','Grupo Mexico'='GMEXICOB.MX',

'Gruma'='GRUMAB.MX','Grupo Aeroportuario del Pacifico'='GAPB.MX',

'Grupo Carso'='GCARSOA1.MX','Grupo Financiero Inbursa'='GFINBURO.MX',

'Grupo Financiero Santander Mexico'='SANMEXB.MX','Grupa Lala'='LALAB.MX',

'Grupo Televisa'='TLEVISACPO.MX','Infraestructura Energetica Nova'='IENOVA.MX',

'Kimberly-Clark de Mexico'='KIMBERA.MX','Megacable Holdings'='MEGACPO.MX',

'Mexichem'='MEXCHEM.MX','Nemak'='NEMAKA.MX','Grupo Aeroportuario del Centro Norte'='OMAB.MX',

'Industrias Penoles'='PE&OLES.MX','Promotora y Operadora de Infraestructura'='PINFRA.MX',

'Controladora Vuela Compania de Aviacion'='VOLARA.MX','Wal-Mart de Mexico'='WALMEX.MX'

)

Lo siguiente es definir el UI que es lo que verá el usuario, primero se acomoda los datos en pestañas para que el usuario lo vea de una forma más sencilla y después se crea las opciones para que el usuario pueda seleccionar hasta 5 activos y el capital a invertir.

# Define UI for application

ui <- fluidPage(

# Application title

titlePanel("Portafolio de inversion para principiantes"),

#Tabs para informacion, rendimiento y estimacion

tabsetPanel(

tabPanel("Informacion", textOutput("error"),

tableOutput("texto")),

tabPanel("Rendimiento",tableOutput("original"),tableOutput("mejor")),

tabPanel("Estimacion",textOutput("infofor"),plotlyOutput("forecast"),plotlyOutput("forecastc"),

textOutput("infoform"),plotlyOutput("forecastm"),plotlyOutput("forecastmc"))

),

hr(),

#Para elegir de 1 a 5 acciones y capital a invertir

fluidRow (

column(4,selectInput(

inputId="activo1", label="Elige tu accion",

choices=acciones, multiple = FALSE,

width = NULL, size = NULL),

selectInput(inputId="activo2", label="Elige tu accion",

choices=acciones, multiple = FALSE,

width = NULL, size = NULL)),

column(4,selectInput(inputId="activo3", label="Elige tu accion",

choices=acciones, multiple = FALSE,

width = NULL, size = NULL),

selectInput(inputId="activo4", label="Elige tu accion",

choices=acciones, multiple = FALSE,

width = NULL, size = NULL)),

column(4,selectInput(inputId="activo5", label="Elige tu accion",

choices=acciones, multiple = FALSE,

width = NULL, size = NULL),

numericInput(inputId="dinero", label="Dinero a invertir",

value=0, min=0, max=NA,step=NA,width=NULL),

actionButton("do", "Click Me")

)

)

)

Después se realiza el procedimiento detrás de los resultados en donde hay una serie de funciones:

1. La descarga de los activos con getsymbols
2. Se obtiene los rendimientos de los precios descargados (cuando es mas de 1)
3. El rendimiento, riesgo y títulos de un solo activo
4. Una función para volver una matriz de correlación a un vector
5. La selección de activos, primero se pasa por un filtro de rendimiento positivo y correlación menor de 0.3 de los activos, si alguno de ellos no lo cumple se elimina y se agrega otro dentro del IPC y se vuelve a pasar por las condiciones hasta que el n número de acciones ingresado por el usuario cumplan los filtros.
6. El rendimiento, riesgo, participación y títulos de un portafolio.
7. La descarga de un archivo en drive donde se encuentra la información de cada activo del IPC almacenada.
8. La predicción de un activo con el modelo ARIMA
9. Se crea un solo vector de precios de un conjunto de activos, tomando como proporciones el portafolio de máximo sharpe.

Por último, se tiene el programa separado en dos partes, lo que debe hacer cuando es solo un activo que es descargar la información, desplegar la información del Excel, su rendimiento y riesgo, así como su pronóstico; cuando es un portafolio debe descargar los precios de cada activo crear el portafolio que pidió el usuario así como el portafolio mejorado por el programa, crear un solo precio ponderado con el portafolio de máximo sharpe y desplegar las estimaciones.

# Define server logic

server <- function(input, output) {

observeEvent(

eventExpr = input$do,handlerExpr = {

fechasf<-seq(as.Date("2017-11-02"), as.Date("2018-02-01"), "days")

## Descarga de activos

descarga<-function(activos){

dataEnv <- new.env()

getSymbols(activos,src="yahoo",from="2015-11-01", to="2017-11-01", env=dataEnv)

plist <- eapply(dataEnv, Ad)

pframe <- do.call(merge, plist)

pframe <- as.timeSeries.xts(pframe)

pframe <- na.locf.default(pframe,na.rm=TRUE,fromLast=TRUE)

pframe<-as.data.frame(pframe)

return(pframe)

}

## Rendimiento de activos

rendimientos<-function(precios){

rendimientos <- log(precios[2:nrow(precios), ] / precios[1:(nrow(precios)-1), ])

return(rendimientos)

}

## rendimiento, riesgo y titulos de 1 activo

meansd<-function(rend,capital,ultimo){

meanma<-data.frame(nrow=1,ncol=3)

meanma[1,1]<-paste("$",round(mean(rend)\*capital,2))

meanma[1,2]<-paste("$",round(sd(rend,na.rm=TRUE)\*capital,2))

meanma[1,3]<-round(capital/ultimo)

colnames(meanma)<-c("Rendimiento ($)","Riesgo ($)","Titulos")

return(meanma)

}

## matriz

flattenSquareMatrix <- function(m) {

if( (class(m) != "matrix") | (nrow(m) != ncol(m))) stop("Must be a square matrix.")

if(!identical(rownames(m), colnames(m))) stop("Row and column names must be equal.")

ut <- upper.tri(m)

data.frame(i = rownames(m)[row(m)[ut]],

j = rownames(m)[col(m)[ut]],

cor=t(m)[ut],

p=m[ut])

}

## seleccion de activos

seleccion<-function(activos){

activos<-sort(activos)

des<-descarga(activos)

des<-des[ , order(names(des))]

rendi<-rendimientos(des)

rendprom<-colMeans(rendi)

numeros<-c()

si<-c()

cor<-c()

for(i in 1:length(rendprom)){

if(rendprom[i]<0){

numeros<-c(numeros,i)

}else

si<-c(si,i)

}

if(length(numeros)!=0){

rendprom<-rendprom[-numeros]

nombres<-activos[-numeros]

}else{

nombres<-activos}

if(length(nombres)!=1){

rendimientosn<-rendi[,si]

colnames(rendimientosn)<-c(nombres)

corrend<-flattenSquareMatrix(cor(rendimientosn, method="spearman" ))

buscar<-which(corrend[3]>.3)#

if(length(buscar)!=0){

for (i in 1:length(buscar)){

ac1<-corrend[buscar[i],1]

ac2<-corrend[buscar[i],2]

cor<-c(cor,which(nombres==ac1))

cor<-c(cor,which(nombres==ac2))

}

doble<-cor[!duplicated(cor)]

rendcov<-rendprom[-doble]

nombres<-nombres[-doble]

}

}else{

rendcov<-rendprom}

return(nombres)}

## rendimiento, riesgo, participaciones y titulo de un portafolio

portafolio<-function(acciones,rendimiento,minimo,maximo,capital,info){

Port1 <- portfolio.spec(assets=acciones)

Port1 <- add.constraint(portfolio=Port1,

type="full\_investment")

# Restriccion 2: Limites superior e inferior para el valor de los pesos individuales

Port1 <- add.constraint(portfolio=Port1,

type="box",

min=minimo, max=maximo)

# Restricci?n 3: Objetivo de rendimiento

Port1 <- add.objective(portfolio=Port1, type="return", name="mean")

Port1 <- optimize.portfolio(R=rendimiento, portfolio=Port1, optimize\_method="random",

trace=TRUE, search\_size=5000)

Portafolios <- vector("list", length = length(Port1$random\_portfolio\_objective\_results))

for(i in 1:length(Port1)) {

Portafolios[[i]]$Pesos <- Port1$random\_portfolio\_objective\_results[[i]]$weights

Portafolios[[i]]$Medias <- Port1$random\_portfolio\_objective\_results[[i]]$objective\_measures$mean

Portafolios[[i]]$Vars <- var.portfolio(R = Port1$R, weights = Portafolios[[i]]$Pesos)

names(Portafolios[[i]]$Medias) <- NULL

}

df\_Portafolios <- data.frame(matrix(nrow=length(Port1),

ncol=2, data = 0))

colnames(df\_Portafolios) <- c("Rendimiento ($)","Riesgo ($)")

for(i in 1:length(Port1)) {

df\_Portafolios[i,1] <- round(Portafolios[[i]]$Medias\*252,4)\*capital

df\_Portafolios[i,2] <- round(sqrt(Portafolios[[i]]$Vars)\*sqrt(252),4)\*capital

for(k in 1:length(acciones)) {

df\_Portafolios[i,paste("Peso de ", acciones[k],sep="")] <- Portafolios[[i]]$Pesos[k]

df\_Portafolios[i,paste("Titulos de ", acciones[k],sep="")] <-

(capital\*Portafolios[[i]]$Pesos[k])%/%info[1,k]

}

}

# Portafolio con m?ximo rendimiento esperado

Port\_1 <- df\_Portafolios[which.max(df\_Portafolios[,1]),]

# Portafolio con m?nima varianza

Port\_2 <- df\_Portafolios[which.min(df\_Portafolios[,2]),]

# Tasa libre de riesgo

rf <- 0.008

# Rendimiento de portafolio

rp <- df\_Portafolios[,2]

# Varianza de portafolio

sp <- df\_Portafolios[,2]

# Indice de sharpe

sharpe <- (rp-rf)/sp

# Portafolio con m?ximo Sharpe ratio

Port\_3 <- df\_Portafolios[which.max(sharpe),]

Ports <- cbind(rbind(Port\_1, Port\_2, Port\_3),

"Portafolio" = c("Maximo Rendimiento","Minima Varianza","Maximo Sharpe Ratio"))

for(i in 1:3){

Ports[i,1]<-paste("$",Ports[i,1])

Ports[i,2]<-paste("$",Ports[i,2])

}

return(Ports)

}

## Excel de informacion

saexcel<-function(activos){

id <- "0B2v9Xt\_MUZ4qWGhGeENFYnRlSzQ" # google file ID

excel<-read.csv(sprintf("https://docs.google.com/uc?id=%s&export=download", id))

igual<-c()

for(i in 1:length(activos)){

sel<-which(activos[i]==excel[,3])

igual<-c(igual,sel)

}

infoa<-excel[igual,2:5]

infoa<-data.frame(infoa)

return(infoa)}

## forecast de solo un activo

foreuno<-function(precios){

arma<-auto.arima(precios)

exac<-accuracy(arma)

pred<-forecast(arma,h=92)

pron<-c(pred$mean)

mov<-pron[63]-pron[1]

if (exac[1]>0.5){

if(mov>0){#alcista

imp<-"Existe una probabilidad que se siga un movimiento alcista. Se recomienda comprar."

}else{

imp<-"Existe una probabilidad que se siga un movimiento bajista. Se recomienda no comprar o mantener."

}}else{

if(mov>0){#alcista

imp<-"Existe una alta probabilidad que se siga un movimiento alcista. Se recomienda comprar."

}else{

imp<-"Existe una alta probabilidad que se siga un movimiento bajista. Se recomiendo no comprar o mantener."

}}

list(imp,pred,pron)

#plot(pred)

}

## crear rendimientos diarios para un portafolio proporciones de max sharpe

juntar<-function(portef,acciones,precios){

proporciones<-c()

suma<-data.frame()

i=1

for (j in 1:length(acciones)){

proporciones<-c(proporciones,portef[3,i+2])

i=i+2

}

for(j in 1:length(precios[,1])){

for(k in 1:length(acciones)){

suma[j,k]<-precios[j,k]\*proporciones[k]

}}

dport<-rowSums(suma)

}

activos<-c(input$activo1,input$activo2,input$activo3,input$activo4,input$activo5)

activos<-activos[activos!=""]

lana<-input$dinero

info<-descarga(activos)

info<-na.omit(info)

j<-as.POSIXct(row.names(info))

precio<-tail(info, n=1)

if(length(activos)==1){

if(lana<precio){

num<-round(as.numeric(precio[1]),2)

output$error<-renderText(

(print(paste("La inversion debe ser mayor a",num,sep =," $")))

)

}else{

withProgress(message = 'Cargando informacion',detail = 'Porfavor espere...', {

rend<-rendimientos(info)

rendsd<-meansd(rend,lana,precio)

ex<-saexcel(activos)

predi<-foreuno(info)

})

output$texto<-renderTable(

{ex} ,striped = TRUE, bordered = TRUE,

spacing = 'm'

#informacion de activo, industria y precio van aqui

#pronostico

)

output$original<-renderTable(

{rendsd} ,striped = TRUE, bordered = TRUE,

spacing = 'm')

output$infofor<-renderText(

predi[[1]]

)

output$forecast<-renderPlotly({

plot\_ly() %>%

add\_lines(x= j, y= info[,1],

color= I("black"), name="Observado") %>%

layout(title='Precios historicos (Nov 2015- Nov 2017) Portafolio Original',xaxis=list(title='Fecha'),yaxis=list(title='Precio'))

})

output$forecastc<-renderPlotly({

plot\_ly() %>%

add\_ribbons(x=fechasf, ymin=predi[[2]]$lower[,2],

ymax=predi[[2]]$upper[,2], color=I("gray95"), name="95% confianza") %>%

add\_ribbons(x=fechasf,ymin=predi[[2]]$lower[,1],

ymax=predi[[2]]$upper[,1],color=I("gray88"), name="88% confianza") %>%

add\_lines(x=fechasf, y=predi[[2]]$mean, color=I("blue"), name="Prediccion") %>%

layout(title='Pronostico (Dic 2017- Marzo 2017)',xaxis=list(title='Fecha'),yaxis=list(title='Precio'))

})

}}else{

info<-descarga(activos)

info<-na.omit(info)

j<-as.POSIXct(row.names(info))

precio<-as.numeric(c(tail(info, n=1)))

precio<-sum(precio)

if(lana<precio){

num<-round(as.numeric(precio[1]),2)

output$error<-renderText(

print(paste("La inversion debe ser mayor a",num,sep = " $"))

)}else{

withProgress(message = 'Cargando informacion',detail = 'Porfavor espere...', { # loading

rend<-rendimientos(info)

longitud<-length(activos)

minimo<-c(rep(.1,longitud))

maximo<-c(rep(.7,longitud))

portuno<-portafolio(activos,rend,minimo,maximo,lana,info)

jun<-as.data.frame(juntar(portuno,activos,info))

predi1<-foreuno(jun)

ex<-saexcel(activos)

uno<-c()

uno<-seleccion(activos)

eliminar<-c()

rest<-0

cont<-2

for(i in 1:length(activos)){

otro<-which(activos[i]==acciones)

eliminar<-c(eliminar,otro)

opciones<-acciones[-eliminar]

}

while(length(uno)!=length(activos)){

cont<-cont+rest

rest<-length(activos)-length(uno)

resta<-rest+cont-1

nuevoacc<-array(opciones[cont:resta])

nuevoacc<-c(uno,nuevoacc)

uno<-seleccion(nuevoacc)

}

infodos<-descarga(uno)

infodos<-na.omit(infodos)

j1<-as.POSIXct(row.names(infodos))

rendo<-rendimientos(info)

portdos<-portafolio(uno,rendo,minimo,maximo,lana,infodos)

junn<-as.data.frame(juntar(portdos,uno,infodos))

predi2<-foreuno(junn)

})

#rendimiento y riesgo de portafolio, info y pronostico

output$texto<-renderTable(

{ex} ,striped = TRUE, bordered = TRUE,

spacing = 'm'

)

output$original<-renderTable(

{ portuno} ,striped = TRUE, bordered = TRUE,

spacing = 'm'

)

output$forecast<-renderPlotly({

plot\_ly() %>%

add\_lines(x= j, y= jun[,1],

color= I("black"), name="Observado") %>%

layout(title='Precios historicos (Nov 2015- Nov 2017) Portafolio Mejorado',xaxis=list(title='Fecha'),yaxis=list(title='Precio'))

})

output$forecastc<-renderPlotly({

plot\_ly() %>%

add\_ribbons(x=fechasf, ymin=predi1[[2]]$lower[,2],

ymax=predi1[[2]]$upper[,2], color=I("gray95"), name="95% confianza") %>%

add\_ribbons(x=fechasf,ymin=predi1[[2]]$lower[,1],

ymax=predi1[[2]]$upper[,1],color=I("gray88"), name="88% confianza") %>%

add\_lines(x=fechasf, y=predi1[[2]]$mean, color=I("blue"), name="Prediccion") %>%

layout(title='Pronostico (Dic 2017- Marzo 2017)',xaxis=list(title='Fecha'),yaxis=list(title='Precio'))

})

output$infofor<-renderText(

predi1[[1]]

)

output$infoform<-renderText(

predi2[[1]]

)

output$mejor<-renderTable(

{portdos} ,striped = TRUE, bordered = TRUE,

spacing = 'm'

)

output$forecastm<-renderPlotly({

plot\_ly() %>%

add\_lines(x= j1, y= junn[,1],

color= I("black"), name="Observado") %>%

layout(title='Precios historicos (Nov 2015- Nov 2017)',xaxis=list(title='Fecha'),yaxis=list(title='Precio'))

})

output$forecastmc<-renderPlotly({

plot\_ly() %>%

add\_ribbons(x=fechasf, ymin=predi2[[2]]$lower[,2],

ymax=predi2[[2]]$upper[,2], color=I("gray95"), name="95% confianza") %>%

add\_ribbons(x=fechasf,ymin=predi2[[2]]$lower[,1],

ymax=predi2[[2]]$upper[,1],color=I("gray88"), name="88% confianza") %>%

add\_lines(x=fechasf, y=predi2[[2]]$mean, color=I("blue"), name="Prediccion") %>%

layout(title='Pronostico (Dic 2017- Marzo 2017)',xaxis=list(title='Fecha'),yaxis=list(title='Precio'))

})

}}

}#termina boton

)

}

# Run the application

shinyApp(ui = ui, server = server)

En conclusión, la aplicación busca ser una herramienta fácil y accesible a personas que desconozcan el tema de invertir en acciones dentro de la BMV, a partir de temas complejos se trató de llegar a respuestas básicas para que la persona común que tenga dinero extra y busque invertir pueda explorar más allá de bonos libres de riesgo a un plazo y capital definido. Los retos encontrados fueron principalmente adquirir conocimiento de shiny app y generalizar funciones que sirvieran para n caso de acciones.