# INCREDULO FINANCEIRO versao junho 2020

# rotina que roda a estrategia de preco medio

# FUNCAO DUPLAMONTECARLO

# funcao que gera dois vetores sendo um de inicio de periodo e outro com o respectivo final

# de periodo, e garantindo que exista uma comprimento (distancia) minimo de periodo

DuplaMonteCarlo=function(inicVet,fimVet,distMinEntrePontos,nSimul,meuSeed){

# gera os vetores iniciais usando sementes (seeds) diferentes para evitar repeticao de valores

set.seed(meuSeed)

amostra1=sample(inicVet:fimVet, size=nSimul, replace=TRUE)

set.seed(meuSeed+608)

amostra2=sample(inicVet:fimVet, size=nSimul, replace=TRUE)

# percorre os vetores checando se vetor de final de periodo tem valores superiores so de

# inicio de periodo, e que o comprimento dos periodos seja maior ou igual ao minimo estabelecido

for(i in 1:length(amostra1)){

# troca os valores se o inicio for maior que o final

if(amostra1[i]>amostra2[i]) {

xyz=amostra1[i]

amostra1[i]=amostra2[i]

amostra2[i]=xyz

}

# checa se o comprimento minimo foi atendido

if((amostra2[i]-amostra1[i])<distMinEntrePontos) {

amostra2[i]=amostra1[i]+distMinEntrePontos

if(amostra2[i]>fimVet) {

amostra2[i]=fimVet

amostra1[i]=amostra2[i]-distMinEntrePontos

}

}

}

return(cbind(amostra1,amostra2))

}

# inicio do codigo de preco medio

library(tidyquant)

library(jsonlite)

# le os tickers das acoes que compoem o IBOVESPA

tickers = read.csv('Ibovespa.csv',sep = ';')

# adiciona coluna que guardara os resultados de proporcao de vezes que a estrategia venceu o benchmark

tickers=cbind(tickers,Vencedores=0.0)

# data inicial e final do periodo de analise

per1="2010-01-02"

per2="2020-04-30"

#parametros iniciais

percQueda=0.09 # queda percentual do ativo para fazer a compra

percSelic=1.0 # 1.0 = 100% percentual da taxa SELIC que sera comparada com o retorno da acao

valorInvestimento=1000 # investimento a cada vez que faz o preco medio

InvestimentoInicial=0 # investimento inicial no ativo antes de comecar a estrategia

# le as cotacoes da taxa SELIC do site do Banco Central

# e constroi indice SELIC

perinicial=paste(substr(per1,9,10),'/',substr(per1,6,7),'/',substr(per1,1,4),sep = '')

perfinal=paste(substr(per2,9,10),'/',substr(per2,6,7),'/',substr(per2,1,4),sep = '')

frase=paste("https://api.bcb.gov.br/dados/serie/bcdata.sgs.12/dados?formato=json&dataInicial=",perinicial,"&dataFinal=",perfinal,sep ="")

selic=na.omit(fromJSON(frase))

selic=cbind(selic,indice=0.0)

selic[1,'indice']=valorInvestimento

novaData=array(data="",dim=length(selic))

novaData[1]=paste(substr(selic[1,'data'],7,10),'-',substr(selic[1,'data'],4,5),'-',substr(selic[1,'data'],1,2),sep="")

for(i in 2:nrow(selic)){

retornoSelic=(as.double(selic[i,'valor'])/100)\*percSelic

selic[i,'indice']=selic[i-1,'indice']\*(1+retornoSelic)

novaData[i]=paste(substr(selic[i,'data'],7,10),'-',substr(selic[i,'data'],4,5),'-',substr(selic[i,'data'],1,2),sep="")

}

# transforma em serie de tempo

novaData2=as.Date(novaData)

selicXts=xts(selic[,'indice'],order.by = novaData2)

# estabele o numero de simulacoes que serao realizada

nSimulacoes=5000

# varre os tickers e faz as simulacoes da estrategia para cada acao (ticker)

for (k in 1:nrow(tickers)) {

# le as cotacoes no Yahoo Finance

options("getSymbols.warning4.0"=FALSE)

options("getSymbols.yahoo.warning"=FALSE)

tickerYahoo=paste(tickers[k,1],".SA",sep = "")

precos=getSymbols(tickerYahoo, from = per1,to = per2,warnings = FALSE,auto.assign = FALSE)

precos=merge(precos,selicXts)

precos=na.omit(precos)

# gera periodos amostrais dentro do horizonte de tempo

amostra=DuplaMonteCarlo(1,nrow(precos),22,nSimulacoes,123)

nTotal=0

# faz as simulacoes para o ticker da vez

for(h in 1:nSimulacoes){

# prepara a matriz que guardara os resultados das simulacoes

precos2=precos[,c(4,5,6,7)]

precos2=cbind(precos2,0)

precos2=cbind(precos2,0)

colnames(precos2)=c('fechamento','invAtivo','ajustado','benchmark','qtdativos','invBench')

precos2[amostra[h,1],'invAtivo']=InvestimentoInicial

precos2[amostra[h,1],'qtdativos']=InvestimentoInicial/as.double(precos2[amostra[h,1],'fechamento']) # numero de acoes

precos2[amostra[h,1],'invBench']=InvestimentoInicial

entrou=FALSE

# varre a janela de períodos

for (i in (amostra[h,1]+1):amostra[h,2]) {

retorno=(as.double(precos2[i,'ajustado'])/as.double(precos2[i-1,'ajustado']))-1

precos2[i,'invAtivo']=as.double(precos2[i-1,'invAtivo'])\*(1+retorno)

precos2[i,'qtdativos']=as.double(precos2[i-1,'qtdativos'])

retornoBench=((as.double(precos2[i,'benchmark'])/as.double(precos2[i-1,'benchmark']))-1)\*as.double(percSelic)

precos2[i,'invBench']=as.double(precos2[i-1,'invBench'])\*(1+retornoBench)

if (retorno<=(-percQueda)) { # compra se caiu

precos2[i,'invAtivo']=as.double(precos2[i,'invAtivo'])+valorInvestimento

precos2[i,'qtdativos']=as.double(precos2[i,'qtdativos'])+(valorInvestimento/as.double(precos2[i,'fechamento']))

precos2[i,'invBench']=as.double(precos2[i,'invBench'])+valorInvestimento

# marca entrou=TRUE para saber que a estrategia foi aplicada alguma vez pois

# dependendo do periodo selecionado e do percentual de queda pode ser que a estrategia

# nao seja aplicada

entrou=TRUE

}

}

# conta se a estrategia superou o benchmark

if ((as.double(precos2[amostra[h,2],'invAtivo'])>as.double(precos2[amostra[h,2],'invBench'])) && (entrou)) {

tickers[k,'Vencedores']=tickers[k,'Vencedores']+1

}

# checa se a estrategia do preco medio foi aplicada: entrou = TRUE

if(entrou) {

nTotal=nTotal+1

}

}

# calcula a proporcao de vezes que a estrategia superio o benchmark sobre total de vezes

# que a estrategia foi aplicada

tickers[k,'Vencedores']=tickers[k,'Vencedores']/nTotal

}

# grava os resultados em um arquivo CSV

write.csv(x=tickers,file = "C:/Users/Documentos/resultados.csv")