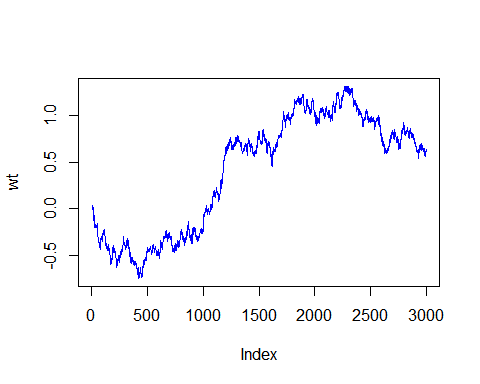
HAIRO ULISES MIRANDA BELMONTE

Miranda Belmonte Hairo

7 de diciembre de 2016

library(quantmod)

brow <- function(r,sigma,T,n,s0){  
 dt=T/n  
 t=seq(0,T,by=dt)  
 zt=rnorm(n,mean=0,sd=1)  
 s=c(s0,r\*dt+sigma\*sqrt(dt)\*zt)  
 st=cumsum(s)  
}  
  
  
dons <- function(r,sigma,T,n,s0){  
 dt=T/n  
 t=seq(0,T,by=dt)  
 zt=rnorm(n,mean=0,sd=1)  
 s=c(s0,r\*dt+sigma\*sqrt(dt)\*zt)  
 st=cumsum(s)/sqrt(n)  
 sf=sum(s)+s0  
}  
  
dons2 <- function(n){  
 zt <- rnorm(n)  
 wt <- cumsum(zt/sqrt(n))  
 plot(wt, type='l', col = "blue")  
}  
dons2(3000)



library(quantmod)  
  
##EJERCICIO2  
  
brow.a<- function(r,sigma,T,n,s0){  
 dt=T/n  
 t=seq(0,T,by=dt)  
 zt=rnorm(n,mean=0,sd=1)  
 s=c(s0,r\*dt+sigma\*sqrt(dt)\*zt)  
 st=cumsum(s)  
   
}  
  
#A  
x1<-brow.a(0,1,1/12,200,10)  
n=200  
  
pp<-x1<0  
spp<-sum(pp)  
p<-spp/n  
p

## [1] 0

pp1<-x1<1  
spp1<-sum(pp1)  
p1<-spp1/n  
p1

## [1] 0

pp2<-x1<9  
spp2<-sum(pp2)  
p2<-spp2/n  
p2

## [1] 0

pp3<-x1<10  
spp3<-sum(pp3)  
p3<-spp3/n  
p3

## [1] 0.9

##EJERCICIO 3  
library(quantmod)  
  
#EJERCICIO 3   
  
GB <- function(r,sigma,T,n,s0){  
 dt=T/n  
 t=seq(0,T,by=dt)  
 bt=rnorm(n,mean=0,sd=1)  
 s=c(s0\*(exp(r-sigma^2/2)\*dt)+sigma\*bt)  
 st=cumsum(s)  
}  
  
x2<-GB(0,1,1/12,200,10)  
m<-10  
n<-200  
x1

ppgm<-x1<m  
sppgm<-sum(ppgm)  
pgm<-sppgm/n  
pgm

## [1] 0.9

library(quantmod)  
  
#EJERCICIO 4  
  
n4<-30  
fprob<-function(x,n){  
 sb1<-na.omit(diff(x))  
 sb<-sb1[1:30]  
 ppsb<-sb<0  
 pb<-sum(ppsb)/n4  
 ps<-1-pb  
 pb  
 ps  
 matrix(c(pb,ps),nrow=1,ncol=2,dimnames = list(c("prob"),c("ProbBajar","ProbSubir")))  
}  
  
  
xa<-GB(0,1,1/365,10000,100)  
fprob(xa,30)

## ProbBajar ProbSubir  
## prob 0.4 0.6

xb<-GB(0,.2,1/365,10000,100)  
fprob(xb,30)

## ProbBajar ProbSubir  
## prob 0.4333333 0.5666667

xc<-GB(.1,.2,1/365,10000,100)  
fprob(xc,30)

## ProbBajar ProbSubir  
## prob 0.5 0.5

xd1<-GB(0,1,1/180,10000,100)  
fprob(xd1,30)

## ProbBajar ProbSubir  
## prob 0.4333333 0.5666667

xd2<-GB(0,.2,1/180,10000,100)  
fprob(xd2,30)

## ProbBajar ProbSubir  
## prob 0.6666667 0.3333333

xd3<-GB(.1,.2,1/180,10000,100)  
fprob(xd3,30)

## ProbBajar ProbSubir  
## prob 0.4666667 0.5333333

library(quantmod)  
#EJERCICIO 5  
#ENCONTRAMOS FUNCI??N SIN EMBARGO NO SE ASEGURA DE QUE SEA CORRECTA   
genlattice <- function(X0, u, d, N) {  
 X <- c()  
 X[1] <- X0  
 count <- 2  
   
 for (i in 1:N) {  
 for (j in 0:i) {  
 X[count] <- X0 \* u^j \* d^(i-j)  
 count <- count + 1  
 }  
 }  
 return(X)  
}  
genlattice(1,2,.5,6)

## [1] 1.000000 0.500000 2.000000 0.250000 1.000000 4.000000 0.125000  
## [8] 0.500000 2.000000 8.000000 0.062500 0.250000 1.000000 4.000000  
## [15] 16.000000 0.031250 0.125000 0.500000 2.000000 8.000000 32.000000  
## [22] 0.015625 0.062500 0.250000 1.000000 4.000000 16.000000 64.000000

#EJERCICIO 6  
  
YF<-function(x){  
 z<-x[,1]  
 row<-sample(1:length(z), 1)  
 rand<-x[row,]  
 Adjt<-rand[,6]  
 Adjt2<-x[row+1,6]  
 AdjtDF<-as.matrix(Adjt)  
 Adjt2DF<-as.matrix(Adjt2)  
 p<-AdjtDF<Adjt2  
 p  
}  
#PROBEMAS CON LA FUNCION  
f2 <- function(n.rep, x){  
 sar <- replicate(n.rep, YF(x))  
}  
  
r1<-function(x){  
 replicados <- sample(f2(500,x), size = 10000, replace=TRUE)  
 repis <- sum(replicados)/10000  
 repis  
}  
  
  
  
getSymbols("OIL")

## [1] "OIL"

x<-OIL  
r1(x)

## [1] 0.4694

getSymbols("GOLD")

## [1] "GOLD"

x2<-GOLD  
r1(x2)

## [1] 0.4775

getSymbols("CL")

## [1] "CL"

x3<-CL  
r1(x3)

## [1] 0.4816

getSymbols("SILVER")

## [1] "SILVER"

x4<-SILVER  
r1(x4)

## [1] 0.5415

getSymbols("GSPC")

## [1] "GSPC"

x5<-GSPC  
r1(x5)

## [1] 0.49177