Tarea 3 Monte-Carlo

Equipo 2

25/5/2022

```
## Loading required package: xts

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: TTR

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

##
## Attaching package: 'data.table'

## The following objects are masked from 'package:xts':
##
##     first, last

## Warning: package 'PerformanceAnalytics' was built under R version 4.1.3

##
## Attaching package: 'PerformanceAnalytics'

## The following object is masked from 'package:graphics':
##
##     legend

## Warning: package 'Deriv' was built under R version 4.1.3

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:data.table':
##
##     between, first, last

## The following objects are masked from 'package:xts':
##
##     first, last

## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

Carga de Funciones a utilizar

```
#FUNCIÓN DE INTERPOLACIÓN ALAMBRADA

talamb=function(nodos,curva,plazos) #función de interpolación de tasas por el método alamabrada
{
  n=max(ncol(plazos),1)
  m=max(ncol(nodos),1)
  TC=matrix(0,1,n)
  TL=matrix(0,1,n)
  TF=matrix(0,1,n)
  for (j in 1:n)
  {
    i=1
    repeat
    {
      if(nodos[i]<= plazos[j] && plazos[j] <=nodos[i+1])
      {
        TC[j]=curva[i]
        TL[j]=curva[i+1]
        TF[j]=((((1+TL[j]*nodos[i+1]/360)/(1+TC[j]*nodos[i]/360))^((plazos[j]-nodos[i])/(nodos[i+1]-n
        break
      }
      else if (plazos[j]<nodos[1])
      {
        TC[j]=curva[1]
        TL[j]=curva[1]
        TF[j]=curva[1]
        break
      }
      else if (plazos[j]>nodos[m])
      {
        TC[j]=curva[m]
        TL[j]=curva[m]
        TF[j]=curva[m]
        break
      }
      else
      {i=i+1}
    }
  }
  as.matrix(t(as.numeric(rbind(TF))))
}


#funciones necesarias
diagv=function(x)          #función para diagonalizar un vector
{
  n01=nrow(as.matrix(x))
  m01=ncol(as.matrix(x))
```

```
  dimmax=max(n01,m01)
  res=matrix(0,dimmax,dimmax)
  for (i in 1:dimmax)
  {
    res[i,i]=x[i]
  }
  res
}

#función de cuantil más cercano
equantile <- function(v,p=.5,ns=nrow(as.matrix(v)))
{
  if ( !is.numeric(p) || any( p<0 | p>1) )
    stop("Percentil tiene que ser 0<=p<=1")
  ranking <- order(v)
  vw=matrix(0,ns,1)
  vw[1:ns]=seq(1/ns,ns)
  sumw <- cumsum(vw[ranking])
  plist <- sumw / sumw[ length(sumw) ]
  v [ ranking [ which.max( plist >= p ) ] ]
}


wquantile <- function(v,w=rep(1,length(v)),p=.5)
{
  if ( !is.numeric(w) || length(v) != length(w) )
    stop("Los valores y los pesos tienen que tener misma longitud")
  if ( !is.numeric(p) || any( p<0 | p>1) )
    stop("Percentil tiene que ser 0<=p<=1")
  if ( min(w) < 0 ) stop("Los pesos tiene que ser mayores que 0")
  ranking <- order(v)
  sumw <- cumsum(w[ranking])
  plist <- sumw / sumw[ length(sumw) ]
  v [ ranking [ which.max( plist >= p ) ] ]
}


#CVaR con alisado
wcvar <- function(v,w=rep(1,length(v)),p=.5)
{
  if ( !is.numeric(w) || length(v) != length(w) )
    stop("Los valores y los pesos tienen que tener misma longitud")
  if ( !is.numeric(p) || any( p<0 | p>1) )
    stop("Percentil tiene que ser 0<=p<=1")
  if ( min(w) < 0 ) stop("Los pesos tiene que ser mayores que 0")
  ranking <- order(v)
  sumw <- cumsum(w[ranking])
  plist <- sumw / sumw[ length(sumw) ]
  loss= v [ ranking [ which( plist < p ) ] ]
  esc=w [ ranking [ which( plist < p ) ] ]
  sum(loss*esc)/(sum(esc))
}
```

Definir fechas de valoración y parámetros de activos a evaluar

```
fval=as.Date("20220313",format="%Y%m%d") #Fecha de valoración
itpl=0 #poner 0 si se quiere interpolación lineal o 1 si se quiere tasa alambrada
alpha=0.98 #Nivel de confianza para obtener estimaciones de riesgo
#setwd(direc)

#################ACCIONES Y DIVISAS###############################
#Cargar los símbolos de yahoo finance para EQ
Symbols<-c("AMXL.MX","GCARSOA1.MX", "WALMEX.MX")#tienen que ir en orden alfabético
pos_eq=c(-5000,1000,1200) #monto inicial invertido en acciones
#Cargar los símbolos de yahoo finance para FX
SymbolsFX<-c("EURUSD=X", "GBPUSD=X","USDMXN=X") #tienen que ir en orden alfabético
pos_fx=c(700,-600, 1500) #monto inicial invertido en divisas
nh=3660 #días de historia

#############CETES#########################
#CETES
base="ARF_2022-2_EQUIPO_2/Tarea 1/Insumos/tasa_guber.txt"
plazos_bcc=cbind( 5,    18, 72, 115,    153,    245,    850,    1500,   3000,   6400,   7500,   9958)
contratos_bcc=cbind(22000, -29000, 10000, -46000, 44000, -26000,-30000,7000,-18000,-31000,23000,-23000)
nominal_bcc=10 #CETE

############Bondes y Bonos M################
#BONDES D
btasadescst="ARF_2022-2_EQUIPO_2/Tarea 1/Insumos/tasa_guber_st.txt"
btasafondeo="ARF_2022-2_EQUIPO_2/Tarea 1/Insumos/tfondeo.txt"
plazos_bdm=cbind(707) #Vencimiento del bono
plazocupon_bdm=cbind(28) #plazos_bdm fijos de cada cupón
contratos_bdm=cbind(1000) #posición invertida
nominal_bdm=1000

#BONOS M
btasadesc_bm="ARF_2022-2_EQUIPO_2/Tarea 1/Insumos/tasa_yield.txt"
tfcupon_bm=cbind( 0.065) #Tasafija del cupón
plazos_bm=cbind( 3600) #Vencimiento del bono
plazocupon_bm=cbind( 360) #plazos_bm fijos de cada cupón
contratos_bm=cbind(1000) #posición invertida
nominal_bm=1000

#########################Forwards#################
#FORWARDS Divisas
bext="ARF_2022-2_EQUIPO_2/Tarea 1/Insumos/tasa_libor.txt"
bdom="ARF_2022-2_EQUIPO_2/Tarea 1/Insumos/tasa_fwd.txt"
SymbolsFX_ftdc<-c("USDMXN=X","GBPUSD=X") #tienen que ir en orden alfabético
plazos_fwd=cbind(5)
contratos_fwd=cbind(100)
kst_fwd=cbind(20.83)
nominal_fwd=1
yext=1 #si se carga información de yahoo en la fecha definida por fval o SymbolsFX, en caso contrario s
trlib=1 #1 si la curva libor viene a 182 0 si no.

#FORWARDS DE IPC
#Descontamos con gubernamental
```

```
base="ARF_2022-2_EQUIPO_2/Tarea 1/Insumos/tasa_guber.txt"
SymbolsEQ_find<-SymbolsEQ_find<-c("^MXX", "GCARSOA1.MX") #tienen que ir en orden alfabético
#tienen que ir en orden alfabético
plazos_fwd_ind=cbind(53)
contratos_fwd_ind=cbind(50)
kst_fwd_ind=cbind(49525)
nominal_fwd_ind=1

###################SWAPS##########
#SWAP
btasadesc_sw="ARF_2022-2_EQUIPO_2/Tarea 1/Insumos/tasa_TIIE_SW_OP.txt"
btasacupvar_sw="ARF_2022-2_EQUIPO_2/Tarea 1/Insumos/tasa_DIRS_SW_OP.txt"
tasafija_sw=cbind(0.066,.059) #se establece la tasa fija a pagar para cada swap
plazos_sw=cbind(588,270) #se establece el número de días que vivirá el swap
plazocupon_sw=cbind(28,28) #se establece el número de días que se pagará cada cupón
contratos_sw=cbind(1,1) #se establece el número de contratos_sw de cada swap
nominal_sw=cbind(16000000,12000000) #se establece el nominal_sw de cada swap
por_sw=cbind(0,1) #se establece 0 si se paga tasa fija y 1 si se paga tasa variable

######################Opciones#########
#OPCIONES
btasadesc_oir="ARF_2022-2_EQUIPO_2/Tarea 1/Insumos/tasa_TIIE_SW_OP.txt"
btasaspot_oir="ARF_2022-2_EQUIPO_2/Tarea 1/Insumos/tasa_DIRS_SW_OP.txt"
bvolspot_oir="ARF_2022-2_EQUIPO_2/Tarea 1/Insumos/tvoltiie_opc.txt"
plazos_oir=cbind( 1700, 700) #T-t
pr_oir=28 #plazo de referencia
dct_oir=360 #d_base
cp_oir=cbind(1,0) #si es call (cap) o put (floor)
K_oir=cbind( 0.058, 0.06)
contratos_oir=cbind(1000, 500)
nominal_oir=1
cs_oir=1 #1 si es continua la tasa 0 si es simple
```

Carga de los datos

###Acciones

```
## [1] "AMXL.MX"     "GCARSOA1.MX" "WALMEX.MX"
```

```
## [1] "EURUSD=X" "GBPUSD=X" "USDMXN=X"
```

```
##                      EURUSD.X GBPUSD.X USDMXN.X AMXL.MX GCARSOA1.MX WALMEX.MX
## 2022-05-19 19:00:00 21.07716 24.82909 19.92360   21.20       72.15     73.70
## 2022-05-22 19:00:00 20.98048 24.81984 19.84942   21.35       73.18     72.45
## 2022-05-23 19:00:00 21.27646 25.03199 19.91796   21.66       74.41     71.22
## 2022-05-24 19:00:00 21.30417 24.88862 19.84270   22.02       76.25     71.61
## 2022-05-25 19:00:00 21.17894 24.94338 19.81820   22.25       78.08     72.56
## 2022-05-26 19:00:00 21.19962 24.92164 19.75190   22.16       80.37     73.34
```

Cetes

```
data<-read.table(base)
n<-nrow(data)
m_gov=ncol(data)
#x_orig_gov=data.frame(data[2:n,1:m_gov])
```

```
x_orig_gov=as.data.table(mutate(data[2:n,1:m_gov],Date=as.Date(V1,format="%Y%m%d")))
x_orig_gov=x_orig_gov%>%select(-V1)
nodos_gov=data.frame(data[1,2:m_gov])
```

Bonos M

```
data1<-read.table(btasadesc_bm)
n<-nrow(data1)
m_tybm=ncol(data1)
X_orig_tybm=as.data.table(mutate(data1[2:n,1:m_tybm],Date=as.Date(V1,format="%Y%m%d")))
X_orig_tybm=X_orig_tybm%>%select(-V1)
nodos_tybm=data.frame(data1[1,2:m_tybm])
```

Bondes D

```
##CARGA DE DATOS DE BONDE D
#carga de datos
#carga de tasas de descuento
data1<-read.table(base)
n<-nrow(data1)
m_bd=ncol(data1)
##X_orig_bd=as.data.table(mutate(data1[2:n,1:m_tybm],Date=as.Date(V1,format="%Y%m%d")))
#X_orig_bd%>%select(-V1)

X1_orig=mutate(data.frame(data1[2:n,1:m_bd]), V1=as.Date(V1,format="%Y%m%d"))
#nodos=data.frame(data1[1,2:m_bd])
n=n-1

data3<-read.table(btasadescst)
n3<-nrow(data3)
m3_bd=ncol(data3)
X3_orig_bd=as.data.table(mutate(data3[2:n,1:m3_bd],Date=as.Date(V1,format="%Y%m%d")))
X3_orig_bd=X3_orig_bd%>%select(-V1)

X3a_orig=mutate(data.frame(data1[2:n,1:m_bd]), V1=as.Date(V1,format="%Y%m%d"))
nodos3_bd=data.frame(data3[1,2:m3_bd])
n3=n3-1

data2<-read.table(btasafondeo)
n2<-nrow(data2)
X2_orig_bd=data.frame(data2[2:n2,1:2])
X2_orig_bd=mutate(X2_orig_bd, V1=as.Date(V1,format="%Y%m%d"), Date=as.Date(V1,format="%Y%m%d"), V2=as.nu
tfh=seq(min(X2_orig_bd$V1), max(X2_orig_bd$V1), "days") #sucesión de dias para tasa fondeo
tfhd=data.frame(ID=1:count(tfh),fecha=tfh)

#Cruzar la sucesión de todos los días versus el de tasa de fondeo
tfhd=setDT(tfhd)[, Date := tfh][order(-Date)]
X2_orig_bd=setDT(X2_orig_bd)[, Date := V1][order(-Date)]
# rolling join unión por rolling, rellena las fechas que faltaban con el último valor conocido "roll=In
X2_orig_bd=X2_orig_bd[tfhd, on = .(Date), roll = Inf]
#buscar fecha de valuación en tfondeo
tf_act=X2_orig_bd[fecha==fval,]$V2/100
tf_int=X2_orig_bd[fecha<=fval & fecha>=(fval-plazocupon_bdm[1])]$V2/100
```

```
X1_orig=setDT(X1_orig)[, Date:= V1][order(-Date)] #Para alinear con valor presente y tasa de fondeo.
```

Forwards de Divisas

```
#datas
#data<-read.table("tasa_tiie.txt")
data1<-read.table(bext)
data2<-read.table(bdom)

########minimos para parametrizar
n1=nrow(data1)
n2=nrow(data2)
m1_ftdc=ncol(data1)
m2_ftdc=ncol(data2)
n=min(n1,n2)-1
###NODOS###
nodos1_ftdc=data.frame(data1[1,2:m1_ftdc])
nodos2_ftdc=data.frame(data2[1,2:m2_ftdc])
####MATRICES DEL MISMO TAMAÑO MENOS DOLAR

x1_ftdc=as.data.table(mutate(data1[2:n,1:m1_ftdc],Date=as.Date(V1,format="%Y%m%d")))
x1_ftdc=x1_ftdc%>%select(-V1)
x2_ftdc=as.data.table(mutate(data2[2:n,1:m2_ftdc],Date=as.Date(V1,format="%Y%m%d")))
x2_ftdc=x2_ftdc%>%select(-V1)

###Para Dolar

if (yext==1)
{
  #Cargar los símbolos de yahoo finance para FX
  start_date=fval-3660 #fecha inicial

  #Creación del objeto para guardar los datos
  dataEnvFX<-new.env()

  #obtener los datos
  getSymbols.yahoo(SymbolsFX_ftdc,env=dataEnvFX,from=start_date, to=(fval))
  #limpiarlos, alinearnos y quedarnos con el precio de cierre
  bt.prep(dataEnvFX,align='remove.na',fill.gaps=T)

  #muestra de datos
  head(dataEnvFX$prices[,2])

  #Nos quedamos con los precios
  X3_ftdc=data.table(Date=as.Date(index(dataEnvFX$prices[,2])),coredata(dataEnvFX$prices[,2]))
} else
{
data3<-read.table(btsp)
print(head(data3))
n3<-nrow(data3)
m3<-ncol(data3)
X3=data.table(as.matrix(as.double(as.matrix(data3[2:(n+1),m3]))))
X3_find=as.data.table(mutate(data3[2:(n+1),1:m3],Date=as.Date(V1,format="%Y%m%d")))
```

```
}

#tail(X3_find)
```

Forwards IPC

```
# CARGA DE DATOS DE  FORWARD DE IPC

data<-read.table(base)
n<-nrow(data)
m_gov=ncol(data)
#x_orig_gov=data.frame(data[2:n,1:m_gov])
x_orig_gov=as.data.table(mutate(data[2:n,1:m_gov],Date=as.Date(V1,format="%Y%m%d")))
x_orig_gov=x_orig_gov%>%select(-V1)
nodos_gov=data.frame(data[1,2:m_gov])

  #Cargar los símbolos de yahoo finance para EQ
  start_date=fval-nh #fecha inicial

  #Creación del objeto para guardar los datos
  dataEnvEQ<-new.env()

  #obtener los datos
  getSymbols.yahoo(SymbolsEQ_find,env=dataEnvEQ,from=start_date, to=(fval))
```

```
## [1] "^MXX"        "GCARSOA1.MX"
```

```
  #limpiarlos, alinearnos y quedarnos con el precio de cierre
  bt.prep(dataEnvEQ,align='remove.na',fill.gaps=T)

  #muestra de datos
#  head(dataEnvEQ$prices)

  #Nos quedamos con los precios
  X3_find=data.table(Date=as.Date(index(dataEnvEQ$prices[,2])),coredata(dataEnvEQ$prices[,2]))

tail(X3_find)
```

```
##          Date      MXX
## 1: 2022-03-04 53322.05
## 2: 2022-03-07 52312.61
## 3: 2022-03-08 53288.23
## 4: 2022-03-09 53911.76
## 5: 2022-03-10 53387.62
## 6: 2022-03-11 53300.70
```

Swaps

```
#CARGA DE DATOS PARA SWAP
  data1<-read.table(btasadesc_sw)
  n1<-nrow(data1)
  m1_orig_sw=ncol(data1)
  X1_orig_sw=data.table(mutate(data1[2:n1,1:m1_orig_sw],Date=as.Date(V1,format="%Y%m%d")))
  X1_orig_sw=X1_orig_sw%>%select(-V1)
```

```
 nodos1_sw=data.frame(data1[1,2:m1_orig_sw])

 data2<-read.table(btasacupvar_sw)
 n2<-nrow(data2)
 m2_orig_sw=ncol(data2)
 X2_orig_sw=data.table(mutate(data2[2:n1,1:m1_orig_sw],Date=as.Date(V1,format="%Y%m%d")))
 X2_orig_sw=X2_orig_sw%>%select(-V1)
 nodos2_sw=data.frame(data2[1,2:m2_orig_sw])
```

Opciones Tasa de interés

```
#CARGA DE DATOS PARA OPCIONES DE TASA DE INTERÉS
 #carga de datos
 #carga de rho
 data1<-read.table(btasadesc_oir)
 n<-nrow(data1)
 m1_orig_oir=ncol(data1)
 x1_orig_oir=data.table(mutate(data1[2:n,1:m1_orig_oir],Date=as.Date(V1,format="%Y%m%d")))
 x1_orig_oir=x1_orig_oir%>%select(-V1)
 nodos1_oir=data.frame(data1[1,2:m1_orig_oir])

 #data1[1:3,]
 #carga de tasas spot
 data2<-read.table(btasaspot_oir)
 n<-nrow(data2)
 m2_orig_oir=ncol(data2)
 x2_orig_oir=data.table(mutate(data2[2:n,1:m2_orig_oir],Date=as.Date(V1,format="%Y%m%d")))
 x2_orig_oir=x2_orig_oir%>%select(-V1)
 nodos2_oir=data.frame(data2[1,2:m2_orig_oir])
 #carga de volatilidades de spot
 data3<-read.table(bvolspot_oir)
 n<-nrow(data3)
 m3_orig_oir=ncol(data3)
 x3_orig_oir=data.table(mutate(data3[2:n,1:m3_orig_oir],Date=as.Date(V1,format="%Y%m%d")))
 x3_orig_oir=x3_orig_oir%>%select(-V1)

 nodos3_oir=data.frame(data3[1,2:m3_orig_oir])
```

Integración de insumos

```
#INTERSECCIÓN DE FECHAS DE TODOS LOS INSUMOS

lin_gub=data.table(Date=as.Date(aux2[x_orig_gov,on=.(Date),nomatch=0]$Date)) #Fechas acciones, equity y
lin_gub=unique(lin_gub)
lin_gub_bmy=data.table(Date=as.Date(lin_gub[X_orig_tybm,on=.(Date),nomatch=0]$Date)) #Fechas acciones,
lin_gub_bmy=unique(lin_gub_bmy)
lin_gub_bmybdst=data.table(Date=as.Date(lin_gub_bmy[X3_orig_bd,on=.(Date),nomatch=0]$Date)) #Fechas acc
lin_gub_bmybdst=unique(lin_gub_bmybdst)
lin_gub_bmybdst_flib=data.table(Date=as.Date(lin_gub_bmybdst[x1_ftdc,on=.(Date),nomatch=0]$Date)) #Fech
lin_gub_bmybdst_flib=unique(lin_gub_bmybdst_flib)
lin_gub_bmybdst_flibfwd=data.table(Date=as.Date(lin_gub_bmybdst_flib[x2_ftdc,on=.(Date),nomatch=0]$Date
lin_gub_bmybdst_flibfwd=unique(lin_gub_bmybdst_flibfwd)
```

```
lin_gub_bmybdst_flibfwdsp=data.table(Date=as.Date(lin_gub_bmybdst_flibfwd[X3_ftdc,on=.(Date),nomatch=0]$
lin_gub_bmybdst_flibfwdsp=unique(lin_gub_bmybdst_flibfwdsp)
lin_gub_bmybdst_flibfwdspind=data.table(Date=as.Date(lin_gub_bmybdst_flibfwdsp[X3_find,on=.(Date),nomat
lin_gub_bmybdst_flibfwdspind=unique(lin_gub_bmybdst_flibfwdspind)
lin_gub_bmybdst_flibfwdspind_swcup=data.table(Date=as.Date(lin_gub_bmybdst_flibfwdspind[X1_orig_sw,on=.
lin_gub_bmybdst_flibfwdspind_swcup=unique(lin_gub_bmybdst_flibfwdspind_swcup)
lin_gub_bmybdst_flibfwdspind_swcupvp=data.table(Date=as.Date(lin_gub_bmybdst_flibfwdspind_swcup[X2_orig
lin_gub_bmybdst_flibfwdspind_swcupvp=unique(lin_gub_bmybdst_flibfwdspind_swcupvp)
lin_gub_bmybdst_flibfwdspind_swcupvp_oirs=data.table(Date=as.Date(lin_gub_bmybdst_flibfwdspind_swcupvp[
lin_gub_bmybdst_flibfwdspind_swcupvp_oirs=unique(lin_gub_bmybdst_flibfwdspind_swcupvp_oirs)
lin_gub_bmybdst_flibfwdspind_swcupvp_oirsvp=data.table(Date=as.Date(lin_gub_bmybdst_flibfwdspind_swcupv
lin_gub_bmybdst_flibfwdspind_swcupvp_oirsvp=unique(lin_gub_bmybdst_flibfwdspind_swcupvp_oirsvp)
lin_gub_bmybdst_flibfwdspind_swcupvp_oirsvpvol=data.table(Date=as.Date(lin_gub_bmybdst_flibfwdspind_swcu
lin_gub_bmybdst_flibfwdspind_swcupvp_oirsvpvol=unique(lin_gub_bmybdst_flibfwdspind_swcupvp_oirsvpvol)

n=nrow(lin_gub_bmybdst_flibfwdspind_swcupvp_oirsvpvol) #Historia de todos
print(n)
```

```
## [1] 241
```

```
#historia de acciones y divisas
stock_prices_EQFX=lin_gub_bmybdst_flibfwdspind_swcupvp_oirsvpvol[aux2,on=.(Date),nomatch=0][order(-Date)
stock_prices_EQFX=stock_prices_EQFX%>%select(-Date)

#historia de curva gubernamental
x_orig_gov=lin_gub_bmybdst_flibfwdspind_swcupvp_oirsvpvol[x_orig_gov,on=.(Date),nomatch=0][order(-Date)]
x_orig_gov=x_orig_gov%>%select(-Date)

#historia de curva yield
X_orig_tybm=lin_gub_bmybdst_flibfwdspind_swcupvp_oirsvpvol[X_orig_tybm,on=.(Date),nomatch=0][order(-Date
X_orig_tybm=X_orig_tybm%>%select(-Date)


#Historia de curvas de bonde
#CONSIDERAR LA CURVA GUBERNAMENTAL X1_ORIG_GOV
X3_orig_bd=lin_gub_bmybdst_flibfwdspind_swcupvp_oirsvpvol[X3_orig_bd,on=.(Date),nomatch=0][order(-Date)]
X3_orig_bd=X3_orig_bd%>%select(-Date)


#historia de curvas de forward tdc
x1_ftdc=lin_gub_bmybdst_flibfwdspind_swcupvp_oirsvpvol[x1_ftdc,on=.(Date),nomatch=0][order(-Date)]
x1_ftdc=x1_ftdc%>%select(-Date)/100
x2_ftdc=lin_gub_bmybdst_flibfwdspind_swcupvp_oirsvpvol[x2_ftdc,on=.(Date),nomatch=0][order(-Date)]
x2_ftdc=x2_ftdc%>%select(-Date)/100
X3_ftdc=lin_gub_bmybdst_flibfwdspind_swcupvp_oirsvpvol[X3_ftdc,on=.(Date),nomatch=0][order(-Date)]
X3_ftdc=X3_ftdc%>%select(-Date)

#historia de curvas de forward ind
#CONSIDERAR LA CURVA GUBERNAMENTAL X1_ORIG_GOV
X3_find=lin_gub_bmybdst_flibfwdspind_swcupvp_oirsvpvol[X3_find,on=.(Date),nomatch=0][order(-Date)]
X3_find=X3_find%>%select(-Date)

#historia de swaps
X1_orig_sw=lin_gub_bmybdst_flibfwdspind_swcupvp_oirsvpvol[X1_orig_sw,on=.(Date),nomatch=0][order(-Date)]
```

```
X1_orig_sw=X1_orig_sw%>%select(-Date)/100
X2_orig_sw=lin_gub_bmybdst_flibfwdspind_swcupvp_oirsvpvol[X2_orig_sw,on=.(Date),nomatch=0][order(-Date)]
X2_orig_sw=X2_orig_sw%>%select(-Date)/100

#historia de opciones
x1_orig_oir=lin_gub_bmybdst_flibfwdspind_swcupvp_oirsvpvol[x1_orig_oir,on=.(Date),nomatch=0][order(-Date
x1_orig_oir=x1_orig_oir%>%select(-Date)/100
x2_orig_oir=lin_gub_bmybdst_flibfwdspind_swcupvp_oirsvpvol[x2_orig_oir,on=.(Date),nomatch=0][order(-Date
x2_orig_oir=x2_orig_oir%>%select(-Date)/100
x3_orig_oir=lin_gub_bmybdst_flibfwdspind_swcupvp_oirsvpvol[x3_orig_oir,on=.(Date),nomatch=0][order(-Date
x3_orig_oir=x3_orig_oir%>%select(-Date)
```

Tenemos 241 factores de riesgo

Medición del riesgo Por método de simulación Monte Carlo

Para todos los instrumentos hay que definir y calcular los siguientes elementos:

1.Historico de factores de riesgo

2.Vector de precios actual

3.Valoración al día actual

Histórico de factores de riesgo

Acciones y Divisas

```
x0_acc_div=stock_prices_EQFX[1,]
V0_acc_div=cbind(t(pos_fx),t(pos_eq))*x0_acc_div
m_fx=length(pos_fx)
m_acc=length(pos_eq)
```

Cetes

```
#CETE CÁLCULO
#paso para interpolar las tasas
m=ncol(plazos_bcc)
x_bcc=matrix(0,n,m)
for (i in 1:(n))
{
  #x[i,]=approx(nodos_gov,x_orig_gov[i,],plazos_bcc)$y
  x_bcc[i,]=if(itpl==0){approx(nodos_gov,x_orig_gov[i,],plazos_bcc,rule=2)$y}else{talamb(nodos_gov,x_ori
}

x0_bcc=x_bcc[1,]

#función
bonocupcero = function(i,t)
{
  1/(1+i*t/360)
}

V0_bcc=as.matrix(bonocupcero(x0_bcc,plazos_bcc))*contratos_bcc*nominal_bcc #Valor actual de cada bono
```

```
VT0_bcc=sum(V0_bcc) #Valor total del portafolio al tiempo 0
```

Bono cupón cero

```
#BONO M  CÁLCULO
#Posición inicial
#interpolación de tasas y volatilidades
m=ncol(plazos_bm)
X_bm=matrix(0,n,m)
for (i in 1:(n))
{
  X_bm[i,]=if(itpl==0){approx(nodos_tybm,X_orig_tybm[i,],plazos_bm,rule=2)$y}else{talamb(nodos_tybm,X_o
}



#Función de valoración por tasa yield
bonoMyield=function(x, plazos_bm, plazocupon_bm, tfcupon_bm, nominal_bm, contratos_bm) #valoración bono
{
  N=as.integer(plazos_bm/plazocupon_bm)+1
  a=(1-(1+x*plazocupon_bm/360)^(-N))/(plazocupon_bm*x/360)
  p1=plazos_bm-plazocupon_bm*(N-1)
  ((contratos_bm*nominal_bm*tfcupon_bm*plazocupon_bm/360)*a+(contratos_bm*nominal_bm)/((1+x*plazocupon_
}

x0_bm=t(as.matrix(X_bm[1,])) #tasas de descuento valor actual
V0_bm=bonoMyield(x0_bm,plazos_bm, plazocupon_bm, tfcupon_bm, nominal_bm, contratos_bm)
```

Bondes D

```
#BONDE D CÁLCULO

X2_pr=lin_gub_bmybdst_flibfwdspind_swcupvp_oirsvpvol[X2_orig_bd, on = .(Date),nomatch=0][order(-Date)]
m=ncol(plazos_bdm)


N_bd=as.integer(plazos_bdm/plazocupon_bdm)+1 #número de cupones a pagar
VTplazos_bdm=matrix(0,1,sum(N_bd)) #vector de todos los plazos_bdm de todos los contratos_bdm
contratos_bdmT=matrix(0,1,sum(N_bd)) #vector de todos los contratos_bdm de todos los flujos de todos lo
nominal_bdmT=matrix(0,1,sum(N_bd)) #vector de todos los nominal_bdmes de todos los flujos de todos los
plazocupon_bdmT=matrix(0,1,sum(N_bd)) #vector de todos los plazos_bdmcupon de todos los flujos de todos
tasafijaT_bd=matrix(0,1,sum(N_bd)) #vector de tasas fijas de todos los flujos de todos los contratos_bd
ulNomT_bd=matrix(0,1,sum(N_bd)) #vector de contratos_bdm a final de flujo


plazini_bd=plazos_bdm-plazocupon_bdm*(N_bd-1) #vector de plazos_bdm iniciales
ddv=plazocupon_bdm-plazini_bd #dias trasncurridos del cupón vigente
tfcupon=matrix(0,1,m)  #El primero cupón de cada bono
tfcupondev=matrix(0,1,m) #cupón de los días devengados
tfcupgen=((1+tf_act/360)^(plazocupon_bdm[1])-1)*360/plazocupon_bdm[1] #el segundo al último cupón de to
#calcula cupones de bonos
for (j in (1:m))
```

```
{
  tfcupondev[j]=(prod(1+tf_int[(1:ddv[j])]/360)-1)*360/ddv[j]
  tfcupon[j]=((1+tfcupondev[j]*ddv[j]/360)*(1+tf_act/360)^(plazocupon_bdm[1]-ddv[j])-1)*360/plazocupon_
}


for (j in (1:m))
{
  if (j==1)
  {
    VTplazos_bdm[,1:sum(N_bd[1:j])]=seq(plazini_bd[j],plazos_bdm[j], by=plazocupon_bdm[j])
    contratos_bdmT[,1:sum(N_bd[1:j])]=seq(contratos_bdm[j],contratos_bdm[j])
    plazocupon_bdmT[,1:sum(N_bd[1:j])]=seq(plazocupon_bdm[j],plazocupon_bdm[j])
    ulNomT_bd[,sum(N_bd[1:j])]=contratos_bdm[j]
    tasafijaT_bd[,1]=tfcupon[j]
    tasafijaT_bd[,2:sum(N_bd[1:j])]=seq(tfcupgen,tfcupgen)
  }
  else
  {
    VTplazos_bdm[,(sum(N_bd[1:j-1])+1):sum(N_bd[1:j])]=seq(plazini_bd[j],plazos_bdm[j], by=plazocupon_b
    contratos_bdmT[,(sum(N_bd[1:j-1])+1):sum(N_bd[1:j])]=seq(contratos_bdm[j],contratos_bdm[j])
    plazocupon_bdmT[,(sum(N_bd[1:j-1])+1):sum(N_bd[1:j])]=seq(plazocupon_bdm[j],plazocupon_bdm[j])
    tasafijaT_bd[,(sum(N_bd[1:j-1])+1)]=tfcupon[j]
    tasafijaT_bd[,(sum(N_bd[1:j-1])+2):sum(N_bd[1:j])]=seq(tfcupgen,tfcupgen)
    ulNomT_bd[,sum(N_bd[1:j])]=contratos_bdm[j]
  }
}

Xvp_bd=matrix(0,n,ncol(VTplazos_bdm))
Xst_bd=matrix(0,n,ncol(VTplazos_bdm))

for (i in (1:n))
{
  Xvp_bd[i,]=if(itpl==0){approx(nodos_gov,x_orig_gov[i,],VTplazos_bdm,rule=2)$y}else{talamb(nodos_gov,x_
  Xst_bd[i,]=if(itpl==0){approx(nodos3_bd,X3_orig_bd[i,],VTplazos_bdm,rule=2)$y}else{talamb(nodos3_bd,X3

}

X_bd_tc=matrix(1,n,ncol(contratos_bdmT))*X2_pr$V2/100
X_bd_ext=cbind(X_bd_tc,as.matrix(Xvp_bd),as.matrix(Xst_bd))

bondeD=function(contratos_bdmT,  nominal_bdm, tf_act, plazocupon_bdmT, VTplazos_bdm, Xvp, Xst, N,ddv)
{
  tfcupon=matrix(0,1,m)  #El primero cupón de cada bono
  tfcupondev=matrix(0,1,m) #cupón de los días devengados
  tfcupgen=((1+tf_act/360)^(plazocupon_bdm[1])-1)*360/plazocupon_bdm[1] #el segundo al último cupón de
  tasafijaT=matrix(0,1,sum(N))
  #calcula cupones de bonos
  for (j in (1:m))
  {
    tfcupondev[j]=(prod(1+tf_int[(1:ddv[j])]/360)-1)*360/ddv[j]
    tfcupon[j]=((1+tfcupondev[j]*ddv[j]/360)*(1+tf_act/360)^(plazocupon_bdm[1]-ddv[j])-1)*360/plazocupo
  }
```

```
  for (j in (1:m))
  {
    if (j==1)
    {
      tasafijaT[,1]=tfcupon[j]
      tasafijaT[,2:sum(N[1:j])]=seq(tfcupgen,tfcupgen)
    }
    else
    {
      tasafijaT[,(sum(N[1:j-1])+1)]=tfcupon[j]
      tasafijaT[,(sum(N[1:j-1])+2):sum(N[1:j])]=seq(tfcupgen,tfcupgen)
    }
  }


  V0=matrix(0,1,count(N))
  V0f=((((contratos_bdmT*(tasafijaT)*(plazocupon_bdmT/360))+ulNomT_bd)/(1+(Xvp+Xst)*VTplazos_bdm/360)))
  for (j in (1:count(N)))
  {
    if(j==1)
    {
      V0[j]=sum(V0f[j:N[j]])
    }
    else
    {
      V0[j]=sum(V0f[(sum(N[1:j-1])+1):(sum(N[1:j]))])
    }
  }
  V0
}



V0_bd=bondeD(contratos_bdmT, nominal_bdm, tf_act, plazocupon_bdmT, VTplazos_bdm, Xvp_bd[1,], Xst_bd[1,]
```

Forwards ripo de cambio

```
#FORWARDS Y/O FUTUROS DE TIPO DE CAMBIO CÁLCULO

#################MATRICES DE INTERPOLACION LINEAL ####################

m=ncol(plazos_fwd)
X1_fwtdc=matrix(0,n,m)
X2_fwtdc=matrix(0,n,m)

for (j in 1:n)
{
  X1_fwtdc[j,]=if(itpl==0){approx(nodos1_ftdc,x1_ftdc[j,],plazos_fwd,rule=2)$y}else{talamb(nodos1_ftdc,
  X2_fwtdc[j,]=if(itpl==0){approx(nodos2_ftdc,x2_ftdc[j,],plazos_fwd,rule=2)$y}else{talamb(nodos2_ftdc,
  if(trlib==1){X1_fwtdc[j,]=((1+X1_fwtdc[j,])^(plazos_fwd/180)-1)*360/plazos_fwd} #transformación de ac
}
```

```
futuroTC = function(t,tl,tn,s,k) #t=dias por vencer, tn=tasa nacional para tipo de cambio forward, tl=
{
  f=s*((1+tn*t/360)/(1+tl*t/360)) #Se obtiene el tipo de cambio forward
  t(as.numeric((f-k)/(1+t*tn/360))) #Se obtiene el valor del payoff a valor presente con el valor z que
}

X3_ftdc=as.matrix(X3_ftdc)
X_futtdc=cbind(X1_fwtdc,X2_fwtdc,X3_ftdc)

V0_fwtdc=futuroTC(plazos_fwd,X1_fwtdc[1,],X2_fwtdc[1,],X3_ftdc[1,],kst_fwd)*contratos_fwd*nominal_fwd
```

Forwards acciones o índices

```
#FORWARDS Y/O FUTUROS DE ÍNDICES CÁLCULO

##################MATRICES DE INTERPOLACION LINEAL ####################

m_ind=ncol(plazos_fwd_ind)
X1_fwind=matrix(0,n,m_ind) #DIVIDENDOS
X2_fwind=matrix(0,n,m_ind)

for (j in 1:n)
{
  #X1_fwind[j,]=if(itpl==0){approx(nodos1_,x1_ftdc[j,],plazos_fwd)$y}else{talamb(nodos1_ftdc,x1_ftdc[j,
  X2_fwind[j,]=if(itpl==0){approx(nodos_gov,x_orig_gov[j,],plazos_fwd_ind,rule=2)$y}else{talamb(nodos_g
  #if(trlib==1){X1_fwtdc[j,]=((1+X1_fwtdc[j,])^(plazos_fwd/180)-1)*360/plazos_fwd} #transformación de a
}

X3_find=as.matrix(X3_find)
X_futind=cbind(X1_fwind,X2_fwind,matrix(X3_find,n,ncol(X1_fwind)))

V0_fwind=futuroTC(plazos_fwd_ind,X1_fwind[1,],X2_fwind[1,],X3_find[1,],kst_fwd_ind)*contratos_fwd_ind*n
```

SWAPS

```
##SWAP TASA FIJA VS TASA VARIABLE CÁLCULO

##Interpolamos

nodosvp=nodos1_sw
nodostc=nodos2_sw
curvavp=as.matrix(X1_orig_sw)
curvatc=X2_orig_sw
n1=nrow(curvavp)
n2=nrow(curvatc)


m=max(ncol(plazos_sw),1) #número de contratos_sw swap a valorar
N=matrix(0,1,m) #es un vector de m valores donde se cargarán los m número de cupones a pagar para cada
for (j in (1:m))
{
```

```
  N[j]=as.integer(plazos_sw[j]/plazocupon_sw[j])+1 #número de cupones a pagar
}
VTplazos_sw=matrix(0,1,sum(N)) #vector de todos los plazos_sw de todos los contratos_sw
contratos_swT=matrix(0,1,sum(N)) #vector de todos los contratos_sw de todos los flujos de todos los con
nominal_swT=matrix(0,1,sum(N)) #vector de todos los nominal_swes de todos los flujos de todos los contr
por_swT=matrix(0,1,sum(N)) #vector de todos los dummy si paga o recibe de todos los flujos de todos los
plazocupon_swT=matrix(0,1,sum(N)) #vector de todos los plazos_swcupon de todos los flujos de todos los
tasafija_swT=matrix(0,1,sum(N)) #vector de tasas fijas de todos los flujos de todos los contratos_sw
VTplazos_swc=matrix(0,1,sum(N)) #vector de todos los plazos_sw cortos de todos los contratos_sw


plazini=plazos_sw-plazocupon_sw*(N-1) #vector de plazos_sw iniciales

for (j in (1:m))
{
  if (j==1)
  {
    VTplazos_sw[,1:sum(N[1:j])]=seq(plazini[j],plazos_sw[j], by=plazocupon_sw[j])
    VTplazos_swc[,1:sum(N[1:j])]=c(0,VTplazos_sw[,1:(sum(N[1:j])-1)])
    contratos_swT[,1:sum(N[1:j])]=seq(contratos_sw[j],contratos_sw[j])
    nominal_swT[,1:sum(N[1:j])]=seq(nominal_sw[j],nominal_sw[j])
    por_swT[,1:sum(N[1:j])]=seq(por_sw[j],por_sw[j])
    plazocupon_swT[,1:sum(N[1:j])]=seq(plazocupon_sw[j],plazocupon_sw[j])
    tasafija_swT[,1:sum(N[1:j])]=seq(tasafija_sw[j],tasafija_sw[j])
  }
  else
  {
    VTplazos_sw[,(sum(N[1:j-1])+1):sum(N[1:j])]=seq(plazini[j],plazos_sw[j], by=plazocupon_sw[j])
    VTplazos_swc[,(sum(N[1:j-1])+1):sum(N[1:j])]=c(0,VTplazos_sw[,(sum(N[1:j-1])+1):(sum(N[1:j])-1)])
    contratos_swT[,(sum(N[1:j-1])+1):sum(N[1:j])]=seq(contratos_sw[j],contratos_sw[j])
    nominal_swT[,(sum(N[1:j-1])+1):sum(N[1:j])]=seq(nominal_sw[j],nominal_sw[j])
    por_swT[,(sum(N[1:j-1])+1):sum(N[1:j])]=seq(por_sw[j],por_sw[j])
    plazocupon_swT[,(sum(N[1:j-1])+1):sum(N[1:j])]=seq(plazocupon_sw[j],plazocupon_sw[j])
    tasafija_swT[,(sum(N[1:j-1])+1):sum(N[1:j])]=seq(tasafija_sw[j],tasafija_sw[j])
  }
}


Xvp=matrix(0,n,ncol(VTplazos_sw))
Xtc=matrix(0,n,ncol(VTplazos_sw))
Xtcc=matrix(0,n,ncol(VTplazos_sw))
XtfwdT=matrix(0,n,ncol(VTplazos_sw))

for (i in (1:n))
{
  Xvp[i,]=if(itpl==0){approx(nodosvp,curvavp[i,],VTplazos_sw,rule=2)$y}else{talamb(nodosvp,curvavp[i,],
  Xtc[i,]=if(itpl==0){approx(nodostc,curvatc[i,],VTplazos_sw,rule=2)$y}else{talamb(nodostc,curvatc[i,],
  Xtcc[i,]=if(itpl==0){approx(nodostc,curvatc[i,],VTplazos_swc, rule=2)$y}else{talamb(nodostc,curvatc[i

  XtfwdT[i,]=((1+Xtc[i,]*VTplazos_sw/360)/(1+Xtcc[i,]*VTplazos_swc/360)-1)*360/plazocupon_swT
  for (j in (1:ncol(VTplazos_sw)))
  {
    if (VTplazos_sw[j]<= plazocupon_swT[j])
```

```
    {
      XtfwdT[i,j]=Xtc[i,j]
    }
    else
    {
      j=sum(N[1:j])
    }
  }
}

X_sw=cbind(XtfwdT,Xvp)

swap=function(por_swT, contratos_swT, nominal_swT, XtfwdT, tasafija_swT, plazocupon_swT, VTplazos_sw, X
{
  V0=matrix(0,1,ncol(N))
  V0f=(((contratos_swT*(XtfwdT-tasafija_swT)*(plazocupon_swT/360)))/(1+Xvp*VTplazos_sw/360))*nominal_sw
  for (j in (1:ncol(N)))
  {
    if(j==1)
    {
      V0[j]=sum(V0f[j:N[j]])
    }
    else
    {
      V0[j]=sum(V0f[(sum(N[1:j-1])+1):(sum(N[1:j]))])
    }
  }
  V0
}

V0_sw=swap(por_swT, contratos_swT, nominal_swT, XtfwdT[1,], tasafija_swT, plazocupon_swT, VTplazos_sw, X
```

Opciones de tasa de interés

```
##opciones de tasa de interés, con inicio el día de la valuación CÁLCULO
#Posición inicial

#interpolación de tasas y volatilidades
m=ncol(plazos_oir)
x1=matrix(0,n,m)
x2tc=matrix(0,n,m)
x2tl=matrix(0,n,m)
x2=matrix(0,n,m)
x3=matrix(0,n,m)
for (i in 1:(n))
{
  x1[i,]=if(itpl==0){approx(nodos1_oir,x1_orig_oir[i,],plazos_oir,rule=2)$y}else{talamb(nodos1_oir,x1_o
  x2tc[i,]=if(itpl==0){approx(nodos2_oir,x2_orig_oir[i,],plazos_oir,rule=2)$y}else{talamb(nodos2_oir,x2
  x2tl[i,]=if(itpl==0){approx(nodos2_oir,x2_orig_oir[i,],(plazos_oir+pr_oir),rule=2)$y}else{talamb(nodos
  x3[i,]=if(itpl==0){approx(nodos3_oir,x3_orig_oir[i,],plazos_oir,rule=2)$y}else{talamb(nodos3_oir,x3_o
  x2[i,]=((1+x2tl[i,]*(plazos_oir+pr_oir)/360)/(1+x2tc[i,]*(plazos_oir)/360)-1)*360/pr_oir
}
```

```
x01=x1[1,] #tasas de descuento
x02=x2[1,] #tasas spot
x03=x3[1,] #volatilidades

X_oir=cbind(x1,x2,x3)

opctint = function(d,S,K_oir,vol,t,cp_oir,cs_oir,pr_oir,dct_oir)    #función de una opción europea
{
  d1=if(cs_oir==1){(log(S/K_oir)+vol^2*t/(365*2))*(1/(vol*sqrt(t/365)))}else{(log(S/K_oir)+vol^2*t/(360
  d2=if(cs_oir==1){(log(S/K_oir)-vol^2*t/(365*2))*(1/(vol*sqrt(t/365)))}else{(log(S/K_oir)-vol^2*t/(360
  vp=if(cs_oir==1){log(1+d*t/360)*365/t}else{d}
  (if(cs_oir==1){(S*pnorm(d1*(-1)^cp_oir)-K_oir*pnorm(d2*(-1)^cp_oir))*(exp(-vp*t/365))*(-1)^cp_oir}else
}


V0_oir=opctint(x01,x02,K_oir,x03,plazos_oir,cp_oir,cs_oir,pr_oir,dct_oir)*contratos_oir*nominal_oir #Va
```

Integración de factores y cálculo de riesgo en conjunto, y aplicación de simulación

Por Cholesky Empírico

```
#DIMENSION DE TODOS LOS INSTRUMENTOS
#Son 8 instrumentos financieros (9 si separamos acciones y divisas)
n_if=matrix(0,8,1)
n_if[1]=ncol(stock_prices_EQFX) #acciones y divisas
n_if[2]=ncol(x_bcc) #cetes
n_if[3]=ncol(X_bm) #bonos m
n_if[4]=ncol(X_bd_ext) #bonde
n_if[5]=ncol(X_futtdc) #fut tdc
n_if[6]=ncol(X_futind) #fut ind
n_if[7]=ncol(X_sw) #swaps
n_if[8]=ncol(X_oir) #opciones tasa de interés

#valor del portafolios

V0_port=cbind(V0_acc_div,V0_bcc, V0_bm, V0_bd, V0_fwtdc, V0_fwind, V0_sw, V0_oir) #contrato
V0T_port=sum(V0_port)



#INTEGRACIÓN DE TODOS LOS FACTORES DE RIESGO EN UNA MATRIZ
X_port=cbind(stock_prices_EQFX,x_bcc,X_bm,X_bd_ext,X_futtdc,X_futind,X_sw,X_oir) #Factores de riesgo de

#Cálculo de variaciones Delta_X DEL PORTAFOLIOS
DeltaX_port=as.matrix(X_port[1:(n-1),]/X_port[2:(n),]-1)
DeltaX_port[is.nan(DeltaX_port)] <- 0 #quitamos NaN
DeltaX_port[is.na(DeltaX_port)] <- 0 #quitamos Na
DeltaX_port[is.infinite(DeltaX_port)] <- 0 #quitamos Na


Ns=10000 #Definimos número de simulaciones
alpha=0.98 #Nivel de Confianza para las medidas de riesgo
ChCP=0  #0 Cholesky, 1 Componentes Principales
```

```
normempi=1 #0 normal, 1 empírico
eta=.85 #Varianza explicada  (sólo aplica para Componentes Principales)
mT=ncol(DeltaX_port)


VarDeltaX=if(ChCP==0){cor(DeltaX_port)}else{var(DeltaX_port)*(ncol(DeltaX_port)/(ncol(DeltaX_port)-1))}
```

```
## Warning in cor(DeltaX_port): the standard deviation is zero
```

```
if(ChCP==0) {
  CVarDeltaX=as.matrix(chol(VarDeltaX,pivot=TRUE))
  CVarDeltaX[is.nan(CVarDeltaX)] <- 0 #quitamos NaN
  CVarDeltaX[is.na(CVarDeltaX)] <- 0 #quitamos Na
  print(ncol(CVarDeltaX))
  print(ncol(VarDeltaX))
  M=matrix(0, Ns,mT)
  if(normempi==1)
  {
    for (i in 1:mT)
    {
      M[,i]=quantile(DeltaX_port[,i],runif(Ns))
    }
    Met='Cholesky Empírico'
  } else    {
    for (i in 1:mT)
    {
      M[,i]=rnorm(Ns)*sd(DeltaX_port[,i])*(mT/(mT-1))+mean(DeltaX_port[,i])
    }
    Met='Cholesky Normal'
  }
  DeltaX_s=M%*%CVarDeltaX  #Factores de riesgo simulados
} else { #componentes principales
  G<-eigen(VarDeltaX)      #eigenvectores
  g1<-G$values                  #eigenvalores
  g1p=g1/sum(g1)

  g1pac=cumsum(g1p) #suma acumulada de varianza

  k=which.max(g1pac>eta) #valor mínimo k

  y=as.matrix((DeltaX_port-colMeans(DeltaX_port)))%*%G$vectors[,1:k] #componentes principales
  #/sqrt(diag(var(DeltaX))))
  M=matrix(0, Ns,k)
  if(normempi==0)
  {
    for (i in 1:k)
    {
      M[,i]=rnorm(Ns)*sd(y[,i])+mean(y[,i])
    }
    Met='Componentes Principales Normales'
  } else    {
    for (i in 1:k)
    {
      M[,i]=quantile(y[,i],runif(Ns))
```

```
    }
    Met='Componentes Principales Empíricos'
  }
  DeltaX_s=M%*% t(G$vectors[,1:k]) #Factores de riesgo simulados
}
```

```
## Warning in chol.default(VarDeltaX, pivot = TRUE): the matrix is either rank-
## deficient or indefinite
```

```
## [1] 173
## [1] 173
```

```
#Met
#DeltaX_port[1:5,]


#Met


Ms1='Se explica el'
Ms2= 'de varianza'
Ms3= 'de suma de var-cov'

Ms1
```

```
## [1] "Se explica el"
```

```
print(sum(diag(var(DeltaX_s)))/sum(diag(var(DeltaX_port)))) #Comprobación de matriz de varianza-cov vs
```

```
## [1] 0.6101215
```

```
Ms2
```

```
## [1] "de varianza"
```

```
Ms1
```

```
## [1] "Se explica el"
```

```
print(matrix(1,1,ncol(DeltaX_s))%*%var(DeltaX_s)%*%t(matrix(1,1,ncol(DeltaX_s)))/matrix(1,1,ncol(DeltaX
```

```
##           [,1]
## [1,] 0.430026
```

```
Ms3
```

```
## [1] "de suma de var-cov"
```

```
print(mT)
```

```
## [1] 173
```

```
#print(k) #número de variables simuladas
#print(g1pac)
```

Medición del riesgo

```
#Medición de riesgo por instrumento, instrumento-factor de riesgo, instrumento - total
```

```
#Cálculo de matriz de pérdidas y ganancias Acciones y Divisas

#riesgo del acciones y divisas
m=n_if[1]   #PASO CLAVE
X_s_acc_div=matrix(0,Ns,n_if[1]) #Factores de riesgo simulados con base en DeltaX_s x0*(1+Delta_Xs)   #
V_acc_div=matrix(0,Ns,m) #valor simulado a nivel contrato de acciones y tdc
Vfr1_acc_div=matrix(0,Ns,m_fx) #valor simulado a nivel contrato de tdc
Vfr2_acc_div=matrix(0,Ns,m_acc) #valor simulado a nivel contrato de acciones
PG_acc_div=matrix(0,Ns,m) #Pèrdidas y ganancias
PGfr1_acc_div=matrix(0,Ns,m_fx)
PGfr2_acc_div=matrix(0,Ns,m_acc)
PGT_acc_div=matrix(0,Ns,1) #TOTAL ACC Y DIV
PGfr1T_acc_div=matrix(0,Ns,1) #total factor riesgo divisas
PGfr2T_acc_div=matrix(0,Ns,1) #total factor riesgo acciones

DeltaX_s_acc_div=DeltaX_s[,(1:n_if[1])]    #PASO CLAVE
x0_acc_div=stock_prices_EQFX[1,]   #PASO CLAVE

for (i in 1:Ns)
{
  X_s_acc_div[i,]=as.matrix(x0_acc_div*(1+DeltaX_s_acc_div[i,]))
  #PASO CLAVE
  V_acc_div[i,]=cbind(t(pos_fx),t(pos_eq))*X_s_acc_div[i,]
  #PASO CLAVE
  Vfr1_acc_div[i,]=t(pos_fx)*X_s_acc_div[i,1:m_fx]
  #PASO CLAVE
  Vfr2_acc_div[i,]=t(pos_eq)*X_s_acc_div[i,(m_fx+1):(m_fx+m_acc)]
  #PASO CLAVE
  PG_acc_div[i,]=as.matrix(V_acc_div[i,]-V0_acc_div)
  PGfr1_acc_div[i,]=as.matrix(Vfr1_acc_div[i,]-V0_acc_div[,1:m_fx])
  PGfr2_acc_div[i,]=as.matrix(Vfr2_acc_div[i,]-V0_acc_div[,(m_fx+1):(m_fx+m_acc)])
  PGT_acc_div[i,]=sum(PG_acc_div[i,])
  PGfr1T_acc_div[i,]=sum(PGfr1_acc_div[i,])
  PGfr2T_acc_div[i,]=sum(PGfr2_acc_div[i,])
}

print(V_acc_div[1:3,])
```

Acciones y divisas

```
##          [,1]      [,2]     [,3]      [,4]     [,5]     [,6]
## [1,] 16252.10 -16534.64 31314.47 -100891.0 63663.90 90825.42
## [2,] 16121.26 -16588.23 31420.68  -99828.6 62430.42 91048.70
## [3,] 16284.86 -16453.12 31557.77 -100340.7 64694.64 91893.38
```

```
print(Vfr1_acc_div[1:3,])
```

```
##          [,1]      [,2]     [,3]
## [1,] 16252.10 -16534.64 31314.47
## [2,] 16121.26 -16588.23 31420.68
## [3,] 16284.86 -16453.12 31557.77
```

```
print(Vfr2_acc_div[1:3,])
```

```
##           [,1]     [,2]     [,3]
## [1,] -100891.0 63663.90 90825.42
```

```
## [2,]  -99828.6 62430.42 91048.70
## [3,] -100340.7 64694.64 91893.38
```

```
PG_acc_div[1:5,]
```

```
##            [,1]       [,2]        [,3]       [,4]       [,5]       [,6]
## [1,] 129.456870  -93.74685 -53.5355683 -1291.0409   663.8974  -470.5808
## [2,]  -1.391126 -147.32916  52.6791280  -228.5976  -569.5814  -247.3045
## [3,] 162.211062  -12.22560 189.7650725  -740.7319  1694.6431   597.3786
## [4,] -17.824929  106.50296 -22.3749328 -2031.8281   503.3407 -1169.0425
## [5,] -75.892710   31.72488  -0.4884955  1206.6395 -2887.6963  1828.7236
```

```
PGfr1_acc_div[1:5,]
```

```
##            [,1]       [,2]        [,3]
## [1,] 129.456870  -93.74685 -53.5355683
## [2,]  -1.391126 -147.32916  52.6791280
## [3,] 162.211062  -12.22560 189.7650725
## [4,] -17.824929  106.50296 -22.3749328
## [5,] -75.892710   31.72488  -0.4884955
```

```
PGfr2_acc_div[1:5,]
```

```
##            [,1]       [,2]       [,3]
## [1,] -1291.0409   663.8974  -470.5808
## [2,]  -228.5976  -569.5814  -247.3045
## [3,]  -740.7319  1694.6431   597.3786
## [4,] -2031.8281   503.3407 -1169.0425
## [5,]  1206.6395 -2887.6963  1828.7236
```

```
PGT_acc_div[1:5,]
```

```
## [1] -1115.5499 -1141.5247  1891.0403 -2631.2268   103.0105
```

```
#VaR por posición
VaRCont_acc_div=matrix(0,1,m)
VaRfr1_acc_div=matrix(0,1,m_fx)
VaRfr2_acc_div=matrix(0,1,m_acc)
CVaRCont_acc_div=matrix(0,1,m)
CVaRfr1_acc_div=matrix(0,1,m_fx)
CVaRfr2_acc_div=matrix(0,1,m_acc)
for (i in (1:m))
{
  VaRCont_acc_div[i]=quantile(PG_acc_div[,i],1-alpha,Ns)
  CVaRCont_acc_div[i]= mean(merge(which(PG_acc_div[,i]<VaRCont_acc_div[i]),cbind(seq(1,Ns),PG_acc_div[,
  if (i<=m_fx)
  {
  VaRfr1_acc_div[i]=quantile(PGfr1_acc_div[,i],1-alpha,Ns)
  CVaRfr1_acc_div[i]= mean(merge(which(PGfr1_acc_div[,i]<VaRfr1_acc_div[i]),cbind(seq(1,Ns),PGfr1_acc_di
  }
  if (i<=m_acc)
  {
  VaRfr2_acc_div[i]=quantile(PGfr2_acc_div[,i],1-alpha,Ns)
  CVaRfr2_acc_div[i]= mean(merge(which(PGfr2_acc_div[,i]<VaRfr2_acc_div[i]),cbind(seq(1,Ns),PGfr2_acc_di
  }
}
#Met
```

```
#VaRCont_acc_div
#VaRfr1_acc_div
#VaRfr2_acc_div
#CVaRCont_acc_div
#CVaRfr1_acc_div
#CVaRfr2_acc_div


#VaR Total
#Met
VaRTotal_acc_div=quantile(PGT_acc_div,1-alpha,Ns)
CVaRTotal_acc_div= mean(merge(which(PGT_acc_div<VaRTotal_acc_div),cbind(seq(1,Ns),PGT_acc_div), by.x=1,
VaRTotalfr1_acc_div=quantile(PGfr1T_acc_div,1-alpha,Ns)
CVaRTotalfr1_acc_div= mean(PGfr1T_acc_div[which(PGfr1T_acc_div<VaRTotalfr1_acc_div),])
VaRTotalfr2_acc_div=quantile(PGfr2T_acc_div,1-alpha,Ns)
CVaRTotalfr2_acc_div= mean(PGfr2T_acc_div[which(PGfr2T_acc_div<VaRTotalfr2_acc_div),])

#print("Var Empírico")
#print(cbind(VaRTotal_acc_div,sum(V0_acc_div), VaRCont_acc_div, V0_acc_div))
#print("CVar Empírico")
#print(cbind(CVaRTotal_acc_div,sum(V0_acc_div), CVaRCont_acc_div, V0_acc_div))
#print("Var Normal")
#print(cbind(VaRTotal_acc_div,VaRTotalfr1_acc_div,,VaRTotalfr2_acc_div))
#print("CVar Normal")
#print(cbind(CVaRTotal_acc_div,CVaRTotalfr1_acc_div,CVaRTotalfr2_acc_div))
#print(sum(V0_acc_div))
titles<-c('EURUSD.X', 'GBPUSD.X', 'USDMXN.X', 'AMXL.MX', 'GCARSOA1.MX',  'WALMEX.MX')
```

```
print("Var Empírico")
```

```
## [1] "Var Empírico"
```

```
print('Var total')
```

```
## [1] "Var total"
```

```
print(VaRTotal_acc_div)
```

```
##        2%
## -4808.766
```

```
print('Var por posición')
```

```
## [1] "Var por posición"
```

```
print(cbind(titles,t(VaRCont_acc_div)))
```

```
##      titles
## [1,] "EURUSD.X"    "-151.517334423926"
## [2,] "GBPUSD.X"    "-156.31773420196"
## [3,] "USDMXN.X"    "-337.119679565446"
## [4,] "AMXL.MX"     "-2858.60136303177"
## [5,] "GCARSOA1.MX" "-2544.44309490185"
## [6,] "WALMEX.MX"   "-2863.26737946174"
```

```
print('Var del contrato acciones')
```

```
## [1] "Var del contrato acciones"
```

```
print(VaRTotalfr2_acc_div)
```

```
##        2%
## -4828.591
```

```
print('Var del contrato divisas')
```

```
## [1] "Var del contrato divisas"
```

```
print(VaRTotalfr1_acc_div)
```

```
##        2%
## -419.1055
```

```
print("CVar Empírico")
```

```
## [1] "CVar Empírico"
```

```
print('CVar total')
```

```
## [1] "CVar total"
```

```
print(CVaRTotal_acc_div)
```

```
## [1] -5662.817
```

```
print('CVar por posición')
```

```
## [1] "CVar por posición"
```

```
print(cbind(titles,t(CVaRCont_acc_div)))
```

```
##      titles
## [1,] "EURUSD.X"    "-170.7177725434"
## [2,] "GBPUSD.X"    "-181.122247068466"
## [3,] "USDMXN.X"    "-435.595737717704"
## [4,] "AMXL.MX"     "-3786.07439490174"
## [5,] "GCARSOA1.MX" "-2830.25509055016"
## [6,] "WALMEX.MX"   "-3049.18748086851"
```

```
print('CVar del contrato acciones')
```

```
## [1] "CVar del contrato acciones"
```

```
print(CVaRTotalfr2_acc_div)
```

```
## [1] -5662.208
```

```
print('CVar del contrato divisas')
```

```
## [1] "CVar del contrato divisas"
```

```
print(CVaRTotalfr1_acc_div)
```

```
## [1] -503.3902
```

Bonos

```
#Cálculo de matriz de pérdidas y ganancias BONDES
#dimensión
m=count(N_bd)    #PASO CLAVE
X_s_bd=matrix(0,Ns,n_if[4]) #Factores de riesgo simulados con base en DeltaX_s x0*(1+Delta_Xs)   #PASO
```

```
V_bd=matrix(0,Ns,m)
Vfr1_bd=matrix(0,Ns,m)
Vfr2_bd=matrix(0,Ns,m)
Vfr3_bd=matrix(0,Ns,m)
PG_bd=matrix(0,Ns,m) #Pèrdidas y ganancias
PGfr1_bd=matrix(0,Ns,m)
PGfr2_bd=matrix(0,Ns,m)
PGfr3_bd=matrix(0,Ns,m)
PGT_bd=matrix(0,Ns,1)
PGfr1T_bd=matrix(0,Ns,1)
PGfr2T_bd=matrix(0,Ns,1)
PGfr3T_bd=matrix(0,Ns,1)

DeltaX_s_bd=DeltaX_s[,sum(n_if[1:3],1):sum(n_if[1:4])]    #PASO CLAVE
x0_bd=X_bd_ext[1,]    #PASO CLAVE

for (i in 1:Ns)
{
  X_s_bd[i,]=x0_bd*(1+DeltaX_s_bd[i,])
  #PASO CLAVE
  V_bd[i,]=bondeD(contratos_bdmT, nominal_bdm, X_s_bd[i,1], plazocupon_bdmT, VTplazos_bdm, X_s_bd[i,(n_i
  #PASO CLAVE
  Vfr1_bd[i,]=bondeD(contratos_bdmT, nominal_bdm, X_s_bd[i,1], plazocupon_bdmT, VTplazos_bdm, x0_bd[(n_i
  #PASO CLAVE
  Vfr2_bd[i,]=bondeD(contratos_bdmT, nominal_bdm, x0_bd[1], plazocupon_bdmT, VTplazos_bdm, X_s_bd[i,(n_i
  #PASO CLAVE
  Vfr3_bd[i,]=bondeD(contratos_bdmT, nominal_bdm, x0_bd[1], plazocupon_bdmT, VTplazos_bdm, x0_bd[(n_if[4
  #PASO CLAVE
  PG_bd[i,]=V_bd[i,]-V0_bd
  PGfr1_bd[i,]=Vfr1_bd[i,]-V0_bd
  PGfr2_bd[i,]=Vfr2_bd[i,]-V0_bd
  PGfr3_bd[i,]=Vfr3_bd[i,]-V0_bd
  PGT_bd[i,]=sum(PG_bd[i,])
  PGfr1T_bd[i,]=sum(PGfr1_bd[i,])
  PGfr2T_bd[i,]=sum(PGfr2_bd[i,])
  PGfr3T_bd[i,]=sum(PGfr3_bd[i,])
}



PG_bd[1:5,]
```

```
## [1]   688.58663   160.49713   293.79283 -2089.33991    29.38055
```

```
PGfr1_bd[1:5,]
```

```
## [1]  703.50417   55.66262  694.30215 -918.34861 -303.70031
```

```
PGfr2_bd[1:5,]
```

```
## [1]  179.58460  581.79333 -396.72270 -266.11659  -86.06783
```

```
PGT_bd[1:5,]
```

```
## [1]   688.58663   160.49713   293.79283 -2089.33991    29.38055
```

```r
#VaR por posición
VaRCont_bd=matrix(0,1,m)
VaRfr1_bd=matrix(0,1,m)
VaRfr2_bd=matrix(0,1,m)
VaRfr3_bd=matrix(0,1,m)
CVaRCont_bd=matrix(0,1,m)
CVaRfr1_bd=matrix(0,1,m)
CVaRfr2_bd=matrix(0,1,m)
CVaRfr3_bd=matrix(0,1,m)
for (i in (1:m))
{
  VaRCont_bd[i]=quantile(PG_bd[,i],1-alpha,Ns)
  VaRfr1_bd[i]=quantile(PGfr1_bd[,i],1-alpha,Ns)
  VaRfr2_bd[i]=quantile(PGfr2_bd[,i],1-alpha,Ns)
  VaRfr3_bd[i]=quantile(PGfr3_bd[,i],1-alpha,Ns)
  CVaRfr1_bd[i]= mean(merge(which(PGfr1_bd[,i]<VaRfr1_bd[i]),cbind(seq(1,Ns),PGfr1_bd[,i]), by.x=1,by.y=
  CVaRfr2_bd[i]= mean(merge(which(PGfr2_bd[,i]<VaRfr2_bd[i]),cbind(seq(1,Ns),PGfr2_bd[,i]), by.x=1,by.y=
  CVaRfr3_bd[i]= mean(merge(which(PGfr3_bd[,i]<VaRfr3_bd[i]),cbind(seq(1,Ns),PGfr3_bd[,i]), by.x=1,by.y=
  CVaRCont_bd[i]= mean(merge(which(PG_bd[,i]<VaRCont_bd[i]),cbind(seq(1,Ns),PG_bd[,i]), by.x=1,by.y=1)[
}
Met
```

```
## [1] "Cholesky Empírico"
```

```r
VaRCont_bd
```

```
##           [,1]
## [1,] -2289.306
```

```r
VaRfr1_bd
```

```
##           [,1]
## [1,] -1660.188
```

```r
VaRfr2_bd
```

```
##           [,1]
## [1,] -1780.419
```

```r
CVaRCont_bd
```

```
##           [,1]
## [1,] -2768.103
```

```r
CVaRfr1_bd
```

```
##           [,1]
## [1,] -2018.988
```

```r
CVaRfr2_bd
```

```
##           [,1]
## [1,] -2331.949
```

```r
#VaR Total
#Met
VaRTotal_bd=quantile(PGT_bd,1-alpha,Ns)
CVaRTotal_bd= mean(merge(which(PGT_bd<VaRTotal_bd),cbind(seq(1,Ns),PGT_bd), by.x=1,by.y=1)[,2])
VaRTotalfr1_bd=quantile(PGfr1T_bd,1-alpha,Ns)
CVaRTotalfr1_bd= mean(PGfr1T_bd[which(PGfr1T_bd<VaRTotalfr1_bd),])
```

```
VaRTotalfr2_bd=quantile(PGfr2T_bd,1-alpha,Ns)
CVaRTotalfr2_bd= mean(PGfr2T_bd[which(PGfr2T_bd<VaRTotalfr2_bd),])
VaRTotalfr3_bd=quantile(PGfr3T_bd,1-alpha,Ns)
CVaRTotalfr3_bd= mean(PGfr3T_bd[which(PGfr2T_bd<VaRTotalfr2_bd),])


#cbind(VaRTotal_bd,sum(V0_bd), VaRCont_bd, V0_bd)
#cbind(CVaRTotal_bd,sum(V0_bd), CVaRCont_bd, V0_bd)
#cbind(VaRTotal_bd,VaRTotalfr1_bd,VaRTotalfr2_bd,VaRTotalfr3_bd)
#cbind(CVaRTotal_bd,CVaRTotalfr1_bd,CVaRTotalfr2_bd,CVaRTotalfr3_bd)
lista<-c('Tasa descuento','tasa fondeo', 'sobretasa')
```

```
print('Var total')
```

```
## [1] "Var total"
```

```
print(VaRTotal_bd)
```

```
##        2%
## -2289.306
```

```
print('Var por factor de riesgo')
```

```
## [1] "Var por factor de riesgo"
```

```
print(rbind(lista,t(c(VaRTotalfr1_bd,VaRTotalfr2_bd,VaRTotalfr3_bd))))
```

```
##       2%                  2%                  2%
## lista "Tasa descuento"    "tasa fondeo"       "sobretasa"
##       "-1660.18765676098" "-1780.41947724603" "-796.528381030348"
```

```
print('CVar total')
```

```
## [1] "CVar total"
```

```
print(CVaRTotal_bd)
```

```
## [1] -2768.103
```

```
print('CVar por factor de riesgo')
```

```
## [1] "CVar por factor de riesgo"
```

```
print(rbind(lista,t(c(CVaRTotalfr1_bd,CVaRTotalfr2_bd,CVaRTotalfr3_bd))))
```

```
##       [,1]                [,2]                [,3]
## lista "Tasa descuento"    "tasa fondeo"       "sobretasa"
##       "-2018.98811245182" "-2331.94937469329" "-11.1821251731046"
```

Forwards de Divisas

```
#Cálculo de matriz de pérdidas y ganancias FUTUROS TDC
#dimensión
m=ncol(plazos_fwd)  #PASO CLAVE
X_s_fwtdc=matrix(0,Ns,n_if[5]) #Factores de riesgo simulados con base en DeltaX_s x0*(1+Delta_Xs) #PASO
V_fwtdc=matrix(0,Ns,m)
Vfr1_fwtdc=matrix(0,Ns,m)
Vfr2_fwtdc=matrix(0,Ns,m)
Vfr3_fwtdc=matrix(0,Ns,m)
```

```
PG_fwtdc=matrix(0,Ns,m) #Pèrdidas y ganancias
PGfr1_fwtdc=matrix(0,Ns,m)
PGfr2_fwtdc=matrix(0,Ns,m)
PGfr3_fwtdc=matrix(0,Ns,m)
PGT_fwtdc=matrix(0,Ns,1)
PGfr1T_fwtdc=matrix(0,Ns,1)
PGfr2T_fwtdc=matrix(0,Ns,1)
PGfr3T_fwtdc=matrix(0,Ns,1)

DeltaX_s_fwtdc=DeltaX_s[,sum(n_if[1:4],1):sum(n_if[1:5])]  #PASO CLAVE
x0_fwtdc=X_futtdc[1,] #PASO CLAVE

for (i in 1:Ns)
{
  X_s_fwtdc[i,]=x0_fwtdc*(1+DeltaX_s_fwtdc[i,])
  #PASO CLAVE
  V_fwtdc[i,]=futuroTC(plazos_fwd,X_s_fwtdc[i,1:((n_if[5]-1)/2)],X_s_fwtdc[i,((n_if[5]-1)/2+1):(n_if[5]
  #PASO CLAVE
  Vfr1_fwtdc[i,]=futuroTC(plazos_fwd,X_s_fwtdc[i,1:((n_if[5]-1)/2)],x0_fwtdc[((n_if[5]-1)/2+1):(n_if[5]
  #PASO CLAVE
  Vfr2_fwtdc[i,]=futuroTC(plazos_fwd,x0_fwtdc[1:((n_if[5]-1)/2)],X_s_fwtdc[i,((n_if[5]-1)/2+1):(n_if[5]
  #PASO CLAVE
  Vfr3_fwtdc[i,]=futuroTC(plazos_fwd,x0_fwtdc[1:((n_if[5]-1)/2)],x0_fwtdc[((n_if[5]-1)/2+1):(n_if[5]-1)]
  PG_fwtdc[i,]=V_fwtdc[i,]-V0_fwtdc
  PGfr1_fwtdc[i,]=Vfr1_fwtdc[i,]-V0_fwtdc
  PGfr2_fwtdc[i,]=Vfr2_fwtdc[i,]-V0_fwtdc
  PGfr3_fwtdc[i,]=Vfr3_fwtdc[i,]-V0_fwtdc
  PGT_fwtdc[i,]=sum(PG_fwtdc[i,])
  PGfr1T_fwtdc[i,]=sum(PGfr1_fwtdc[i,])
  PGfr2T_fwtdc[i,]=sum(PGfr2_fwtdc[i,])
  PGfr3T_fwtdc[i,]=sum(PGfr3_fwtdc[i,])
}



PG_fwtdc[1:5,]
```

```
## [1]  12.851660   5.659349   9.625306 -16.220243  -2.341755
```

```
PGfr1_fwtdc[1:5,]
```

```
## [1] -0.005593115 -0.012945392  0.000123219 -0.025887305  0.012445317
```

```
PGfr2_fwtdc[1:5,]
```

```
## [1]  0.0094104676  0.0217806884 -0.0002073175  0.0435553371 -0.0209394849
```

```
PGT_fwtdc[1:5,]
```

```
## [1]  12.851660   5.659349   9.625306 -16.220243  -2.341755
```

```
#VaR por posición
VaRCont_fwtdc=matrix(0,1,m)
VaRfr1_fwtdc=matrix(0,1,m)
VaRfr2_fwtdc=matrix(0,1,m)
VaRfr3_fwtdc=matrix(0,1,m)
```

```
CVaRCont_fwtdc=matrix(0,1,m)
CVaRfr1_fwtdc=matrix(0,1,m)
CVaRfr2_fwtdc=matrix(0,1,m)
CVaRfr3_fwtdc=matrix(0,1,m)
for (i in (1:m))
{
  VaRCont_fwtdc[i]=quantile(PG_fwtdc[,i],1-alpha,Ns)
  VaRfr1_fwtdc[i]=quantile(PGfr1_fwtdc[,i],1-alpha,Ns)
  VaRfr2_fwtdc[i]=quantile(PGfr2_fwtdc[,i],1-alpha,Ns)
  VaRfr3_fwtdc[i]=quantile(PGfr3_fwtdc[,i],1-alpha,Ns)
  CVaRfr1_fwtdc[i]= mean(merge(which(PGfr1_fwtdc[,i]<VaRfr1_fwtdc[i]),cbind(seq(1,Ns),PGfr1_fwtdc[,i]),
  CVaRfr2_fwtdc[i]= mean(merge(which(PGfr2_fwtdc[,i]<VaRfr2_fwtdc[i]),cbind(seq(1,Ns),PGfr2_fwtdc[,i]),
  CVaRfr3_fwtdc[i]= mean(merge(which(PGfr3_fwtdc[,i]<VaRfr3_fwtdc[i]),cbind(seq(1,Ns),PGfr3_fwtdc[,i]),
  CVaRCont_fwtdc[i]= mean(merge(which(PG_fwtdc[,i]<VaRCont_fwtdc[i]),cbind(seq(1,Ns),PG_fwtdc[,i]), by.
}
#Met
#VaRCont_fwtdc
#VaRfr1_fwtdc
#VaRfr2_fwtdc
#CVaRCont_fwtdc
#CVaRfr1_fwtdc
#CVaRfr2_fwtdc


#VaR Total
Met
```

```
## [1] "Cholesky Empírico"
```

```
VaRTotal_fwtdc=quantile(PGT_fwtdc,1-alpha,Ns)
CVaRTotal_fwtdc= mean(merge(which(PGT_fwtdc<VaRTotal_fwtdc),cbind(seq(1,Ns),PGT_fwtdc), by.x=1,by.y=1)[
VaRTotalfr1_fwtdc=quantile(PGfr1T_fwtdc,1-alpha,Ns)
CVaRTotalfr1_fwtdc= mean(PGfr1T_fwtdc[which(PGfr1T_fwtdc<VaRTotalfr1_fwtdc),])
VaRTotalfr2_fwtdc=quantile(PGfr2T_fwtdc,1-alpha,Ns)
CVaRTotalfr2_fwtdc= mean(PGfr2T_fwtdc[which(PGfr2T_fwtdc<VaRTotalfr2_fwtdc),])
VaRTotalfr3_fwtdc=quantile(PGfr3T_fwtdc,1-alpha,Ns)
CVaRTotalfr3_fwtdc= mean(PGfr3T_fwtdc[which(PGfr2T_fwtdc<VaRTotalfr2_fwtdc),])


#cbind(VaRTotal_fwtdc,sum(VO_fwtdc), VaRCont_fwtdc, VO_fwtdc)
#cbind(CVaRTotal_fwtdc,sum(VO_fwtdc), CVaRCont_fwtdc, VO_fwtdc)
cbind(VaRTotal_fwtdc,VaRTotalfr1_fwtdc,VaRTotalfr2_fwtdc,VaRTotalfr3_fwtdc)
```

```
##    VaRTotal_fwtdc VaRTotalfr1_fwtdc VaRTotalfr2_fwtdc VaRTotalfr3_fwtdc
## 2%      -34.12911       -0.02190518       -0.03389921         -34.13065
```

```
cbind(CVaRTotal_fwtdc,CVaRTotalfr1_fwtdc,CVaRTotalfr2_fwtdc,CVaRTotalfr3_fwtdc)
```

```
##      CVaRTotal_fwtdc CVaRTotalfr1_fwtdc CVaRTotalfr2_fwtdc CVaRTotalfr3_fwtdc
## [1,]       -41.19546        -0.02718851        -0.04282737           3.233283
```

Forwards IPC

```
#Cálculo de matriz de pérdidas y ganancias FUTUROS IPC
m=ncol(plazos_fwd_ind)  #PASO CLAVE
```

```
X_s_fwind=matrix(0,Ns,n_if[6]) #Factores de riesgo simulados con base en DeltaX_s x0*(1+Delta_Xs) #PASO
V_fwind=matrix(0,Ns,m)
Vfr1_fwind=matrix(0,Ns,m)
Vfr2_fwind=matrix(0,Ns,m)
Vfr3_fwind=matrix(0,Ns,m)
PG_fwind=matrix(0,Ns,m) #Pèrdidas y ganancias
PGfr1_fwind=matrix(0,Ns,m)
PGfr2_fwind=matrix(0,Ns,m)
PGfr3_fwind=matrix(0,Ns,m)
PGT_fwind=matrix(0,Ns,1)
PGfr1T_fwind=matrix(0,Ns,1)
PGfr2T_fwind=matrix(0,Ns,1)
PGfr3T_fwind=matrix(0,Ns,1)

DeltaX_s_fwind=DeltaX_s[,sum(n_if[1:5],1):sum(n_if[1:6])]  #PASO CLAVE
x0_fwind=X_futind[1,] #PASO CLAVE


for (i in 1:Ns)
{
  X_s_fwind[i,]=x0_fwind*(1+DeltaX_s_fwind[i,])
  #PASO CLAVE
  V_fwind[i,]=futuroTC(plazos_fwd_ind,X_s_fwind[i,1:(n_if[6]/3)],X_s_fwind[i,(n_if[6]/3+1):(n_if[6]*2/3)
  Vfr1_fwind[i,]=futuroTC(plazos_fwd_ind,X_s_fwind[i,1:(n_if[6]/3)],x0_fwind[(n_if[6]/3+1):(n_if[6]*2/3)
  #PASO CLAVE
  Vfr2_fwind[i,]=futuroTC(plazos_fwd_ind,x0_fwind[1:(n_if[6]/3)],X_s_fwind[i,(n_if[6]/3+1):(n_if[6]*2/3)
  #PASO CLAVE
  Vfr3_fwind[i,]=futuroTC(plazos_fwd_ind,x0_fwind[1:(n_if[6]/3)],x0_fwind[(n_if[6]/3+1):(n_if[6]*2/3)],X
  PG_fwind[i,]=V_fwind[i,]-V0_fwind
  PGfr1_fwind[i,]=Vfr1_fwind[i,]-V0_fwind
  PGfr2_fwind[i,]=Vfr2_fwind[i,]-V0_fwind
  PGfr3_fwind[i,]=Vfr3_fwind[i,]-V0_fwind
  PGT_fwind[i,]=sum(PG_fwind[i,])
  PGfr1T_fwind[i,]=sum(PGfr1_fwind[i,])
  PGfr2T_fwind[i,]=sum(PGfr2_fwind[i,])
  PGfr3T_fwind[i,]=sum(PGfr3_fwind[i,])
}


PG_fwind[1:5,]
```

```
## [1]  13134.6625  30783.8246   -261.6085  61207.2911 -29132.9527
```

```
#PGfr1_fwind[1:5,]
#PGfr2_fwind[1:5,]
#PGT_fwind[1:5,]



#VaR por posición
VaRCont_fwind=matrix(0,1,m)
VaRfr1_fwind=matrix(0,1,m)
VaRfr2_fwind=matrix(0,1,m)
VaRfr3_fwind=matrix(0,1,m)
```

```
CVaRCont_fwind=matrix(0,1,m)
CVaRfr1_fwind=matrix(0,1,m)
CVaRfr2_fwind=matrix(0,1,m)
CVaRfr3_fwind=matrix(0,1,m)
for (i in (1:m))
{
  VaRCont_fwind[i]=quantile(PG_fwind[,i],1-alpha,Ns)
  VaRfr1_fwind[i]=quantile(PGfr1_fwind[,i],1-alpha,Ns)
  VaRfr2_fwind[i]=quantile(PGfr2_fwind[,i],1-alpha,Ns)
  VaRfr3_fwind[i]=quantile(PGfr3_fwind[,i],1-alpha,Ns)
  CVaRfr1_fwind[i]= mean(merge(which(PGfr1_fwind[,i]<VaRfr1_fwind[i]),cbind(seq(1,Ns),PGfr1_fwind[,i]),
  CVaRfr2_fwind[i]= mean(merge(which(PGfr2_fwind[,i]<VaRfr2_fwind[i]),cbind(seq(1,Ns),PGfr2_fwind[,i]),
  CVaRfr3_fwind[i]= mean(merge(which(PGfr3_fwind[,i]<VaRfr3_fwind[i]),cbind(seq(1,Ns),PGfr3_fwind[,i]),
  CVaRCont_fwind[i]= mean(merge(which(PG_fwind[,i]<VaRCont_fwind[i]),cbind(seq(1,Ns),PG_fwind[,i]), by.
}
#Met
#VaRCont_fwind
#VaRfr1_fwind
#VaRfr2_fwind
#CVaRCont_fwind
#CVaRfr1_fwind
#CVaRfr2_fwind


#VaR Total
Met
```

```
## [1] "Cholesky Empírico"
```

```
VaRTotal_fwind=quantile(PGT_fwind,1-alpha,Ns)
CVaRTotal_fwind= mean(merge(which(PGT_fwind<VaRTotal_fwind),cbind(seq(1,Ns),PGT_fwind), by.x=1,by.y=1)[
VaRTotalfr1_fwind=quantile(PGfr1T_fwind,1-alpha,Ns)
CVaRTotalfr1_fwind= mean(PGfr1T_fwind[which(PGfr1T_fwind<VaRTotalfr1_fwind),])
VaRTotalfr2_fwind=quantile(PGfr2T_fwind,1-alpha,Ns)
CVaRTotalfr2_fwind= mean(PGfr2T_fwind[which(PGfr2T_fwind<VaRTotalfr2_fwind),])
VaRTotalfr3_fwind=quantile(PGfr3T_fwind,1-alpha,Ns)
CVaRTotalfr3_fwind= mean(PGfr3T_fwind[which(PGfr3T_fwind<VaRTotalfr3_fwind),])


#print(cbind(VaRTotal_fwind,sum(VO_fwind), VaRCont_fwind, VO_fwind))
#print(cbind(CVaRTotal_fwind,sum(VO_fwind), CVaRCont_fwind, VO_fwind))
print(cbind(VaRTotal_fwind,VaRTotalfr1_fwind,VaRTotalfr2_fwind,VaRTotalfr3_fwind))
```

```
##    VaRTotal_fwind VaRTotalfr1_fwind VaRTotalfr2_fwind VaRTotalfr3_fwind
## 2%      -47734.62                 0         -530.5078         -47877.53
```

```
cbind(CVaRTotal_fwind,CVaRTotalfr1_fwind,CVaRTotalfr2_fwind,CVaRTotalfr3_fwind)
```

```
##      CVaRTotal_fwind CVaRTotalfr1_fwind CVaRTotalfr2_fwind CVaRTotalfr3_fwind
## [1,]       -60389.49                NaN          -626.5228           -60486.9
```

Swaps

```
#Cálculo de matriz de pérdidas y ganancias SWAP
```

```
#riesgo del swap
m=ncol(N)    #PASO CLAVE
X_s_sw=matrix(0,Ns,n_if[7]) #Factores de riesgo simulados con base en DeltaX_s x0*(1+Delta_Xs)   #PASO
V_sw=matrix(0,Ns,m)
Vfr1_sw=matrix(0,Ns,m)
Vfr2_sw=matrix(0,Ns,m)
PG_sw=matrix(0,Ns,m) #Pèrdidas y ganancias
PGfr1_sw=matrix(0,Ns,m)
PGfr2_sw=matrix(0,Ns,m)
PGT_sw=matrix(0,Ns,1)
PGfr1T_sw=matrix(0,Ns,1)
PGfr2T_sw=matrix(0,Ns,1)

DeltaX_s_sw=DeltaX_s[,sum(n_if[1:6],1):sum(n_if[1:7])]    #PASO CLAVE
x0_sw=as.numeric(c(XtfwdT[1,],Xvp[1,]))   #PASO CLAVE

for (i in 1:Ns)
{
  X_s_sw[i,]=x0_sw*(1+DeltaX_s_sw[i,])
  #PASO CLAVE
  V_sw[i,]=swap(por_swT, contratos_swT, nominal_swT, X_s_sw[i,1:(n_if[7]/2)], tasafija_swT, plazocupon_s
  #PASO CLAVE
  Vfr1_sw[i,]=swap(por_swT, contratos_swT, nominal_swT,X_s_sw[i,1:(n_if[7]/2)], tasafija_swT, plazocupo
  #PASO CLAVE
  Vfr2_sw[i,]=swap(por_swT, contratos_swT, nominal_swT,  XtfwdT[1,], tasafija_swT, plazocupon_swT, VTpla
  #PASO CLAVE
  PG_sw[i,]=V_sw[i,]-V0_sw
  PGfr1_sw[i,]=Vfr1_sw[i,]-V0_sw
  PGfr2_sw[i,]=Vfr2_sw[i,]-V0_sw
  PGT_sw[i,]=sum(PG_sw[i,])
  PGfr1T_sw[i,]=sum(PGfr1_sw[i,])
  PGfr2T_sw[i,]=sum(PGfr2_sw[i,])
}



PG_sw[1:5,]
```

```
##           [,1]      [,2]
## [1,] 15013.677 -1310.353
## [2,]  5293.224 -2359.696
## [3,]  9741.625 -1841.336
## [4,] -5715.157  2357.263
## [5,] 12256.803 -3292.272
```

```
PGfr1_sw[1:5,]
```

```
##           [,1]      [,2]
## [1,] 15014.423 -1318.134
## [2,]  5276.673 -2378.683
## [3,]  9737.190 -1845.488
## [4,] -5713.886  2346.455
## [5,] 12265.323 -3281.308
```

```
PGfr2_sw[1:5,]
```

```
##            [,1]      [,2]
## [1,] -0.6329447  7.809859
## [2,] 17.5085685 18.521064
## [3,]  5.9185804  4.089697
## [4,] -4.3200839 11.010700
## [5,] -9.8926775 -9.936074
```

```
PGT_sw[1:5,]
```

```
## [1] 13703.324  2933.528  7900.289 -3357.894  8964.531
```

```
#VaR por posición
VaRCont_sw=matrix(0,1,m)
VaRfr1_sw=matrix(0,1,m)
VaRfr2_sw=matrix(0,1,m)
CVaRCont_sw=matrix(0,1,m)
CVaRfr1_sw=matrix(0,1,m)
CVaRfr2_sw=matrix(0,1,m)
for (i in (1:m))
{
  VaRCont_sw[i]=quantile(PG_sw[,i],1-alpha,Ns)
  VaRfr1_sw[i]=quantile(PGfr1_sw[,i],1-alpha,Ns)
  VaRfr2_sw[i]=quantile(PGfr2_sw[,i],1-alpha,Ns)
  CVaRfr1_sw[i]= mean(merge(which(PGfr1_sw[,i]<VaRfr1_sw[i]),cbind(seq(1,Ns),PGfr1_sw[,i]), by.x=1,by.y=
  CVaRfr2_sw[i]= mean(merge(which(PGfr2_sw[,i]<VaRfr2_sw[i]),cbind(seq(1,Ns),PGfr2_sw[,i]), by.x=1,by.y=
  CVaRCont_sw[i]= mean(merge(which(PG_sw[,i]<VaRCont_sw[i]),cbind(seq(1,Ns),PG_sw[,i]), by.x=1,by.y=1)[
}
#Met
#VaRCont_sw
#VaRfr1_sw
#VaRfr2_sw
#CVaRCont_sw
#CVaRfr1_sw
#CVaRfr2_sw


#VaR Total
Met
```

```
## [1] "Cholesky Empírico"
```

```
VaRTotal_sw=quantile(PGT_sw,1-alpha,Ns)
CVaRTotal_sw= mean(merge(which(PGT_sw<VaRTotal_sw),cbind(seq(1,Ns),PGT_sw), by.x=1,by.y=1)[,2])
VaRTotalfr1_sw=quantile(PGfr1T_sw,1-alpha,Ns)
CVaRTotalfr1_sw= mean(PGfr1T_sw[which(PGfr1T_sw<VaRTotalfr1_sw),])
VaRTotalfr2_sw=quantile(PGfr2T_sw,1-alpha,Ns)
CVaRTotalfr2_sw= mean(PGfr2T_sw[which(PGfr2T_sw<VaRTotalfr2_sw),])


cbind(VaRTotal_sw, VaRCont_sw)
```

```
##    VaRTotal_sw
## 2%   -11492.31 -15231.82 -8249.816
```

```
cbind(CVaRTotal_sw, CVaRCont_sw)
```

```
##      CVaRTotal_sw
## [1,]    -13699.51 -18255.9 -11105.65
```

```
cbind(VaRTotal_sw,VaRTotalfr1_sw,VaRTotalfr2_sw)
```

```
##    VaRTotal_sw VaRTotalfr1_sw VaRTotalfr2_sw
## 2%   -11492.31      -11470.79      -54.70924
```

```
cbind(CVaRTotal_sw,CVaRTotalfr1_sw,CVaRTotalfr2_sw)
```

```
##      CVaRTotal_sw CVaRTotalfr1_sw CVaRTotalfr2_sw
## [1,]    -13699.51       -13670.89       -65.81485
```

Opciones de tasa de interés

```
#Cálculo de matriz de pérdidas y ganancias Opciones Tasa de interés
#dimensión
m=ncol(plazos_oir)  #PASO CLAVE
X_s_oir=matrix(0,Ns,n_if[8]) #Factores de riesgo simulados con base en DeltaX_s x0*(1+Delta_Xs) #PASO C
V_oir=matrix(0,Ns,m)
Vfr1_oir=matrix(0,Ns,m)
Vfr2_oir=matrix(0,Ns,m)
Vfr3_oir=matrix(0,Ns,m)
PG_oir=matrix(0,Ns,m) #Pèrdidas y ganancias
PGfr1_oir=matrix(0,Ns,m)
PGfr2_oir=matrix(0,Ns,m)
PGfr3_oir=matrix(0,Ns,m)
PGT_oir=matrix(0,Ns,1)
PGfr1T_oir=matrix(0,Ns,1)
PGfr2T_oir=matrix(0,Ns,1)
PGfr3T_oir=matrix(0,Ns,1)

DeltaX_s_oir=DeltaX_s[,sum(n_if[1:7],1):sum(n_if[1:8])]  #PASO CLAVE
x0_oir=X_oir[1,] #PASO CLAVE

for (i in 1:Ns)
{
  X_s_oir[i,]=x0_oir*(1+DeltaX_s_oir[i,])
  #PASO CLAVE
  V_oir[i,]=  opctint(X_s_oir[i,(1:(n_if[8]/3))],X_s_oir[i,((n_if[8]/3+1):(n_if[8]/3*2))],K_oir,X_s_oir
  #PASO CLAVE
  Vfr1_oir[i,]=opctint(X_s_oir[i,(1:(n_if[8]/3))],x0_oir[((n_if[8]/3+1):(n_if[8]/3*2))],K_oir,x0_oir[((
  #PASO CLAVE
  Vfr2_oir[i,]=opctint(x0_oir[(1:(n_if[8]/3))],X_s_oir[i,((n_if[8]/3+1):(n_if[8]/3*2))],K_oir,x0_oir[((
  #PASO CLAVE
  Vfr3_oir[i,]=opctint(x0_oir[(1:(n_if[8]/3))],x0_oir[((n_if[8]/3+1):(n_if[8]/3*2))],K_oir,X_s_oir[i,((
  PG_oir[i,]=V_oir[i,]-V0_oir
  PGfr1_oir[i,]=Vfr1_oir[i,]-V0_oir
  PGfr2_oir[i,]=Vfr2_oir[i,]-V0_oir
  PGfr3_oir[i,]=Vfr3_oir[i,]-V0_oir
  PGT_oir[i,]=sum(PG_oir[i,])
  PGfr1T_oir[i,]=sum(PGfr1_oir[i,])
  PGfr2T_oir[i,]=sum(PGfr2_oir[i,])
```

```
  PGfr3T_oir[i,]=sum(PGfr3_oir[i,])
}



PG_oir[1:5,]
```

```
##              [,1]         [,2]
## [1,] -0.00804739 -0.004640516
## [2,]  0.01333536  0.005013485
## [3,] -0.01550489  0.001918201
## [4,]  0.01574358 -0.006498954
## [5,] -0.02702574  0.013869290
```

```
PGfr1_oir[1:5,]
```

```
##               [,1]          [,2]
## [1,] -0.0011900679 -1.358025e-04
## [2,] -0.0027495717 -3.140652e-04
## [3,]  0.0000262539  2.993667e-06
## [4,] -0.0054813541 -6.271583e-04
## [5,]  0.0026595876  3.027739e-04
```

```
PGfr2_oir[1:5,]
```

```
##              [,1]         [,2]
## [1,] -0.010532450 -0.005874499
## [2,]  0.007734284  0.002142964
## [3,] -0.015449698  0.001945559
## [4,]  0.004536693 -0.012169158
## [5,] -0.021388351  0.016660772
```

```
PGT_oir[1:5,]
```

```
## [1] -0.012687906  0.018348848 -0.013586687  0.009244625 -0.013156455
```

```
#VaR por posición
VaRCont_oir=matrix(0,1,m)
VaRfr1_oir=matrix(0,1,m)
VaRfr2_oir=matrix(0,1,m)
VaRfr3_oir=matrix(0,1,m)
CVaRCont_oir=matrix(0,1,m)
CVaRfr1_oir=matrix(0,1,m)
CVaRfr2_oir=matrix(0,1,m)
CVaRfr3_oir=matrix(0,1,m)
for (i in (1:m))
{
  VaRCont_oir[i]=quantile(PG_oir[,i],1-alpha,Ns)
  VaRfr1_oir[i]=quantile(PGfr1_oir[,i],1-alpha,Ns)
  VaRfr2_oir[i]=quantile(PGfr2_oir[,i],1-alpha,Ns)
  VaRfr3_oir[i]=quantile(PGfr3_oir[,i],1-alpha,Ns)
  CVaRfr1_oir[i]= mean(merge(which(PGfr1_oir[,i]<VaRfr1_oir[i]),cbind(seq(1,Ns),PGfr1_oir[,i]), by.x=1,
  CVaRfr2_oir[i]= mean(merge(which(PGfr2_oir[,i]<VaRfr2_oir[i]),cbind(seq(1,Ns),PGfr2_oir[,i]), by.x=1,
  CVaRfr3_oir[i]= mean(merge(which(PGfr3_oir[,i]<VaRfr3_oir[i]),cbind(seq(1,Ns),PGfr3_oir[,i]), by.x=1,
  CVaRCont_oir[i]= mean(merge(which(PG_oir[,i]<VaRCont_oir[i]),cbind(seq(1,Ns),PG_oir[,i]), by.x=1,by.y=
}
```

```
#Met
#VaRCont_oir
#VaRfr1_oir
#VaRfr2_oir
#CVaRCont_oir
#CVaRfr1_oir
#CVaRfr2_oir


#VaR Total
#Met
VaRTotal_oir=quantile(PGT_oir,1-alpha,Ns)
CVaRTotal_oir= mean(merge(which(PGT_oir<VaRTotal_oir),cbind(seq(1,Ns),PGT_oir), by.x=1,by.y=1)[,2])
VaRTotalfr1_oir=quantile(PGfr1T_oir,1-alpha,Ns)
CVaRTotalfr1_oir= mean(PGfr1T_oir[which(PGfr1T_oir<VaRTotalfr1_oir),])
VaRTotalfr2_oir=quantile(PGfr2T_oir,1-alpha,Ns)
CVaRTotalfr2_oir= mean(PGfr2T_oir[which(PGfr2T_oir<VaRTotalfr2_oir),])
VaRTotalfr3_oir=quantile(PGfr3T_oir,1-alpha,Ns)
CVaRTotalfr3_oir= mean(PGfr3T_oir[which(PGfr2T_oir<VaRTotalfr2_oir),])


cbind(VaRTotal_oir, VaRCont_oir)
```

```
##    VaRTotal_oir
## 2%  -0.03752804 -0.03558353 -0.02050023
```

```
cbind(CVaRTotal_oir, CVaRCont_oir)
```

```
##      CVaRTotal_oir
## [1,]   -0.04912592 -0.04858515 -0.02422213
```

```
cbind(VaRTotal_oir,VaRTotalfr1_oir,VaRTotalfr2_oir,VaRTotalfr3_oir)
```

```
##    VaRTotal_oir VaRTotalfr1_oir VaRTotalfr2_oir VaRTotalfr3_oir
## 2%  -0.03752804     -0.00517353     -0.03706019     -0.01810093
```

```
cbind(CVaRTotal_oir,CVaRTotalfr1_oir,CVaRTotalfr2_oir,CVaRTotalfr3_oir)
```

```
##      CVaRTotal_oir CVaRTotalfr1_oir CVaRTotalfr2_oir CVaRTotalfr3_oir
## [1,]   -0.04912592     -0.006412043      -0.04898526      0.001406324
```

Riesgo total del portafolio

```
#Medición de riesgo por factor de riesgo de todo el portafolios
#Acciones
#1. Acciones
#2. Forward de IPC
PGPort_ACC=PGfr2T_acc_div+ PGfr3T_fwind #Pérdidas y ganancias
VaRPort_ACC=quantile(PGPort_ACC,1-alpha,Ns) #VaR
CVaRPort_ACC= mean(PGPort_ACC[which(PGPort_ACC<VaRPort_ACC)]) #CVaR

print(VaRPort_ACC)
```

```
##       2%
## -48155.1
```

```
print(CVaRPort_ACC)
```

```
## [1] -60067.13
```

```
#Tasa de Interés
#1. Dado que swaps y bondes son de tasa de interés usaremos PGT_bd y PGT_sw
#2. Para futuros usaremos PGfr1T_fwtdc y PGfr2T_fwtdc
PGPort_TI=PGT_bd+PGT_sw+PGfr1T_fwtdc+PGfr2T_fwtdc +PGfr2T_fwind+PGfr1T_fwind +PGfr1T_oir +PGfr2T_oir #P
VaRPort_TI=quantile(PGPort_TI,1-alpha,Ns) #VaR
CVaRPort_TI= mean(PGPort_TI[which(PGPort_TI<VaRPort_TI)]) #CVaR
print(VaRPort_TI)
```

```
##        2%
## -11978.54
```

```
print(CVaRPort_TI)
```

```
## [1] -14197.8
```

```
#Tipo de cambio
#1. Dado que swaps y bondes son de tasa de interés no usamos nada
#2. Para futuros usamos sólo PGfr3T_fwtdc
PGPort_TDC=PGfr1T_acc_div+PGfr3T_fwtdc  #Pérdidas y ganancias
VaRPort_TDC=quantile(PGPort_TDC,1-alpha,Ns) #VaR
CVaRPort_TDC= mean(PGPort_TDC[which(PGPort_TDC<VaRPort_TDC)]) #CVaR
print(VaRPort_TDC)
```

```
##       2%
## -425.811
```

```
print(CVaRPort_TDC)
```

```
## [1] -511.6113
```

```
#Volatilidad
#1. Sólo aplica la volatilidad de Opciones de tasa de interés

PGPort_VOL=PGfr3T_oir  #Pérdidas y ganancias
VaRPort_VOL=quantile(PGPort_VOL,1-alpha,Ns) #VaR
CVaRPort_VOL= mean(PGPort_VOL[which(PGPort_VOL<VaRPort_VOL)]) #CVaR

print(VaRPort_VOL)
```

```
##          2%
## -0.01810093
```

```
print(CVaRPort_VOL)
```

```
## [1] -0.02287157
```

```
#Medición de riesgo de todo el portafolios
#Sumar todos los PGT de todos los instrumentos

PGT_Port=PGPort_ACC+PGPort_TI+PGPort_TDC+PGPort_VOL
VaRTotal_Port=quantile(PGT_Port,1-alpha,Ns) #VaR
CVaRTotal_Port= mean(PGT_Port[which(PGT_Port<VaRTotal_Port)]) #CVaR
print(VaRTotal_Port)
```

```
##        2%
```

```
## -53196.94
```

```
print(CVaRTotal_Port)
```

```
## [1] -66873.52
```

```
print(V0T_port)
```

```
## [1] 1474917
```

Por Cholesky Normal

```
#DIMENSION DE TODOS LOS INSTRUMENTOS
#Son 8 instrumentos financieros (9 si separamos acciones y divisas)
n_if=matrix(0,8,1)
n_if[1]=ncol(stock_prices_EQFX) #acciones y divisas
n_if[2]=ncol(x_bcc) #cetes
n_if[3]=ncol(X_bm) #bonos m
n_if[4]=ncol(X_bd_ext) #bonde
n_if[5]=ncol(X_futtdc) #fut tdc
n_if[6]=ncol(X_futind) #fut ind
n_if[7]=ncol(X_sw) #swaps
n_if[8]=ncol(X_oir) #opciones tasa de interés

#valor del portafolios

V0_port=cbind(V0_acc_div,V0_bcc, V0_bm, V0_bd, V0_fwtdc, V0_fwind, V0_sw, V0_oir) #contrato
V0T_port=sum(V0_port)



#INTEGRACIÓN DE TODOS LOS FACTORES DE RIESGO EN UNA MATRIZ
X_port=cbind(stock_prices_EQFX,x_bcc,X_bm,X_bd_ext,X_futtdc,X_futind,X_sw,X_oir) #Factores de riesgo de

#Cálculo de variaciones Delta_X DEL PORTAFOLIOS
DeltaX_port=as.matrix(X_port[1:(n-1),]/X_port[2:(n),]-1)
DeltaX_port[is.nan(DeltaX_port)] <- 0 #quitamos NaN
DeltaX_port[is.na(DeltaX_port)] <- 0 #quitamos Na
DeltaX_port[is.infinite(DeltaX_port)] <- 0 #quitamos Na


Ns=10000 #Definimos número de simulaciones
alpha=0.98 #Nivel de Confianza para las medidas de riesgo
ChCP=0  #0 Cholesky, 1 Componentes Principales
normempi=0 #0 normal, 1 empírico
eta=.85 #Varianza explicada  (sólo aplica para Componentes Principales)
mT=ncol(DeltaX_port)


VarDeltaX=if(ChCP==0){cor(DeltaX_port)}else{var(DeltaX_port)*(ncol(DeltaX_port)/(ncol(DeltaX_port)-1))}
```

```
## Warning in cor(DeltaX_port): the standard deviation is zero
```

```
if(ChCP==0) {
  CVarDeltaX=as.matrix(chol(VarDeltaX,pivot=TRUE))
  CVarDeltaX[is.nan(CVarDeltaX)] <- 0 #quitamos NaN
```

```
  CVarDeltaX[is.na(CVarDeltaX)] <- 0 #quitamos Na
  print(ncol(CVarDeltaX))
  print(ncol(VarDeltaX))
  M=matrix(0, Ns,mT)
  if(normempi==1)
  {
    for (i in 1:mT)
    {
      M[,i]=quantile(DeltaX_port[,i],runif(Ns))
    }
    Met='Cholesky Empírico'
  } else    {
    for (i in 1:mT)
    {
      M[,i]=rnorm(Ns)*sd(DeltaX_port[,i])*(mT/(mT-1))+mean(DeltaX_port[,i])
    }
    Met='Cholesky Normal'
  }
  DeltaX_s=M%*%CVarDeltaX  #Factores de riesgo simulados
} else { #componentes principales
  G<-eigen(VarDeltaX)      #eigenvectores
  g1<-G$values                  #eigenvalores
  g1p=g1/sum(g1)

  g1pac=cumsum(g1p) #suma acumulada de varianza

  k=which.max(g1pac>eta) #valor mínimo k

  y=as.matrix((DeltaX_port-colMeans(DeltaX_port)))%*%G$vectors[,1:k] #componentes principales
  #/sqrt(diag(var(DeltaX))))
  M=matrix(0, Ns,k)
  if(normempi==0)
  {
    for (i in 1:k)
    {
      M[,i]=rnorm(Ns)*sd(y[,i])+mean(y[,i])
    }
    Met='Componentes Principales Normales'
  } else    {
    for (i in 1:k)
    {
      M[,i]=quantile(y[,i],runif(Ns))
    }
    Met='Componentes Principales Empíricos'
  }
  DeltaX_s=M%*% t(G$vectors[,1:k]) #Factores de riesgo simulados
}
```

```
## Warning in chol.default(VarDeltaX, pivot = TRUE): the matrix is either rank-
## deficient or indefinite
```

```
## [1] 173
## [1] 173
```

```
#Met
#DeltaX_port[1:5,]


#Met


Ms1='Se explica el'
Ms2= 'de varianza'
Ms3= 'de suma de var-cov'

Ms1
```

```
## [1] "Se explica el"
```

```
print(sum(diag(var(DeltaX_s)))/sum(diag(var(DeltaX_port)))) #Comprobación de matriz de varianza-cov vs
```

```
## [1] 0.6715456
```

```
Ms2
```

```
## [1] "de varianza"
```

```
Ms1
```

```
## [1] "Se explica el"
```

```
print(matrix(1,1,ncol(DeltaX_s))%*%var(DeltaX_s)%*%t(matrix(1,1,ncol(DeltaX_s)))/matrix(1,1,ncol(DeltaX
```

```
##          [,1]
## [1,] 0.471561
```

```
Ms3
```

```
## [1] "de suma de var-cov"
```

```
print(mT)
```

```
## [1] 173
```

```
#print(k) #número de variables simuladas
#print(g1pac)
```

Medición del riesgo

```
#Medición de riesgo por instrumento, instrumento-factor de riesgo, instrumento - total


#Cálculo de matriz de pérdidas y ganancias Acciones y Divisas

#riesgo del acciones y divisas
m=n_if[1]   #PASO CLAVE
X_s_acc_div=matrix(0,Ns,n_if[1]) #Factores de riesgo simulados con base en DeltaX_s x0*(1+Delta_Xs)   #
V_acc_div=matrix(0,Ns,m) #valor simulado a nivel contrato de acciones y tdc
Vfr1_acc_div=matrix(0,Ns,m_fx) #valor simulado a nivel contrato de tdc
Vfr2_acc_div=matrix(0,Ns,m_acc) #valor simulado a nivel contrato de acciones
PG_acc_div=matrix(0,Ns,m) #Pèrdidas y ganancias
PGfr1_acc_div=matrix(0,Ns,m_fx)
PGfr2_acc_div=matrix(0,Ns,m_acc)
```

```
PGT_acc_div=matrix(0,Ns,1) #TOTAL ACC Y DIV
PGfr1T_acc_div=matrix(0,Ns,1) #total factor riesgo divisas
PGfr2T_acc_div=matrix(0,Ns,1) #total factor riesgo acciones

DeltaX_s_acc_div=DeltaX_s[,(1:n_if[1])]    #PASO CLAVE
x0_acc_div=stock_prices_EQFX[1,]   #PASO CLAVE

for (i in 1:Ns)
{
  X_s_acc_div[i,]=as.matrix(x0_acc_div*(1+DeltaX_s_acc_div[i,]))
  #PASO CLAVE
  V_acc_div[i,]=cbind(t(pos_fx),t(pos_eq))*X_s_acc_div[i,]
  #PASO CLAVE
  Vfr1_acc_div[i,]=t(pos_fx)*X_s_acc_div[i,1:m_fx]
  #PASO CLAVE
  Vfr2_acc_div[i,]=t(pos_eq)*X_s_acc_div[i,(m_fx+1):(m_fx+m_acc)]
  #PASO CLAVE
  PG_acc_div[i,]=as.matrix(V_acc_div[i,]-V0_acc_div)
  PGfr1_acc_div[i,]=as.matrix(Vfr1_acc_div[i,]-V0_acc_div[,1:m_fx])
  PGfr2_acc_div[i,]=as.matrix(Vfr2_acc_div[i,]-V0_acc_div[,(m_fx+1):(m_fx+m_acc)])
  PGT_acc_div[i,]=sum(PG_acc_div[i,])
  PGfr1T_acc_div[i,]=sum(PGfr1_acc_div[i,])
  PGfr2T_acc_div[i,]=sum(PGfr2_acc_div[i,])
}

print(V_acc_div[1:3,])
```

Acciones y divisas

```
##          [,1]      [,2]     [,3]       [,4]     [,5]     [,6]
## [1,] 16200.95 -16495.72 31382.50  -99700.56 63819.49 91885.13
## [2,] 16186.73 -16486.07 31308.17 -101553.55 63328.17 92875.22
## [3,] 16182.48 -16309.09 31405.47 -101108.32 62856.72 91158.87
```

```
print(Vfr1_acc_div[1:3,])
```

```
##          [,1]      [,2]     [,3]
## [1,] 16200.95 -16495.72 31382.50
## [2,] 16186.73 -16486.07 31308.17
## [3,] 16182.48 -16309.09 31405.47
```

```
print(Vfr2_acc_div[1:3,])
```

```
##            [,1]     [,2]     [,3]
## [1,]  -99700.56 63819.49 91885.13
## [2,] -101553.55 63328.17 92875.22
## [3,] -101108.32 62856.72 91158.87
```

```
PG_acc_div[1:5,]
```

```
##           [,1]       [,2]       [,3]       [,4]      [,5]       [,6]
## [1,]  78.30718  -54.81713   14.49640  -100.5579  819.4874   589.1300
## [2,]  64.08685  -45.17497  -59.83108 -1953.5486  328.1716  1579.2226
## [3,]  59.83046  131.80852   37.46897 -1508.3219 -143.2847  -137.1346
## [4,] 202.63821  -60.44316  -35.97117 -2331.0857  537.1534 -2849.8824
## [5,] -64.35271 -138.30163 -150.91022   545.3072 1144.1494   957.3800
```

```
PGfr1_acc_div[1:5,]
```

```
##           [,1]       [,2]       [,3]
## [1,]  78.30718  -54.81713   14.49640
## [2,]  64.08685  -45.17497  -59.83108
## [3,]  59.83046  131.80852   37.46897
## [4,] 202.63821  -60.44316  -35.97117
## [5,] -64.35271 -138.30163 -150.91022
```

```
PGfr2_acc_div[1:5,]
```

```
##           [,1]      [,2]       [,3]
## [1,]  -100.5579  819.4874   589.1300
## [2,] -1953.5486  328.1716  1579.2226
## [3,] -1508.3219 -143.2847  -137.1346
## [4,] -2331.0857  537.1534 -2849.8824
## [5,]   545.3072 1144.1494   957.3800
```

```
PGT_acc_div[1:5,]
```

```
## [1]  1346.04597   -87.07368 -1559.63320 -4537.59080  2293.27204
```

```
#VaR por posición
VaRCont_acc_div=matrix(0,1,m)
VaRfr1_acc_div=matrix(0,1,m_fx)
VaRfr2_acc_div=matrix(0,1,m_acc)
CVaRCont_acc_div=matrix(0,1,m)
CVaRfr1_acc_div=matrix(0,1,m_fx)
CVaRfr2_acc_div=matrix(0,1,m_acc)
for (i in (1:m))
{
  VaRCont_acc_div[i]=quantile(PG_acc_div[,i],1-alpha,Ns)
  CVaRCont_acc_div[i]= mean(merge(which(PG_acc_div[,i]<VaRCont_acc_div[i]),cbind(seq(1,Ns),PG_acc_div[,
  if (i<=m_fx)
  {
  VaRfr1_acc_div[i]=quantile(PGfr1_acc_div[,i],1-alpha,Ns)
  CVaRfr1_acc_div[i]= mean(merge(which(PGfr1_acc_div[,i]<VaRfr1_acc_div[i]),cbind(seq(1,Ns),PGfr1_acc_d
  }
  if (i<=m_acc)
  {
  VaRfr2_acc_div[i]=quantile(PGfr2_acc_div[,i],1-alpha,Ns)
  CVaRfr2_acc_div[i]= mean(merge(which(PGfr2_acc_div[,i]<VaRfr2_acc_div[i]),cbind(seq(1,Ns),PGfr2_acc_d
  }
}
#Met
#VaRCont_acc_div
#VaRfr1_acc_div
#VaRfr2_acc_div
#CVaRCont_acc_div
#CVaRfr1_acc_div
#CVaRfr2_acc_div


#VaR Total
#Met
VaRTotal_acc_div=quantile(PGT_acc_div,1-alpha,Ns)
```

```
CVaRTotal_acc_div= mean(merge(which(PGT_acc_div<VaRTotal_acc_div),cbind(seq(1,Ns),PGT_acc_div), by.x=1,
VaRTotalfr1_acc_div=quantile(PGfr1T_acc_div,1-alpha,Ns)
CVaRTotalfr1_acc_div= mean(PGfr1T_acc_div[which(PGfr1T_acc_div<VaRTotalfr1_acc_div),])
VaRTotalfr2_acc_div=quantile(PGfr2T_acc_div,1-alpha,Ns)
CVaRTotalfr2_acc_div= mean(PGfr2T_acc_div[which(PGfr2T_acc_div<VaRTotalfr2_acc_div),])

#print("Var Empírico")
#print(cbind(VaRTotal_acc_div,sum(VO_acc_div), VaRCont_acc_div, VO_acc_div))
#print("CVar Empírico")
#print(cbind(CVaRTotal_acc_div,sum(VO_acc_div), CVaRCont_acc_div, VO_acc_div))
#print("Var Normal")
#print(cbind(VaRTotal_acc_div,VaRTotalfr1_acc_div,,VaRTotalfr2_acc_div))
#print("CVar Normal")
#print(cbind(CVaRTotal_acc_div,CVaRTotalfr1_acc_div,CVaRTotalfr2_acc_div))
#print(sum(VO_acc_div))
titles<-c('EURUSD.X', 'GBPUSD.X', 'USDMXN.X', 'AMXL.MX', 'GCARSOA1.MX',  'WALMEX.MX')
```

```
print("Var Normal")
```

```
## [1] "Var Normal"
```

```
print('Var total')
```

```
## [1] "Var total"
```

```
print(VaRTotal_acc_div)
```

```
##        2%
## -5141.854
```

```
print('Var por posición')
```

```
## [1] "Var por posición"
```

```
print(cbind(titles,t(VaRCont_acc_div)))
```

```
##      titles
## [1,] "EURUSD.X"    "-164.371263212634"
## [2,] "GBPUSD.X"    "-167.985612932916"
## [3,] "USDMXN.X"    "-370.370617944912"
## [4,] "AMXL.MX"     "-2926.18640064035"
## [5,] "GCARSOA1.MX" "-2887.59585886592"
## [6,] "WALMEX.MX"   "-2748.87783229332"
```

```
print('Var del contrato acciones')
```

```
## [1] "Var del contrato acciones"
```

```
print(VaRTotalfr2_acc_div)
```

```
##        2%
## -5161.364
```

```
print('Var del contrato divisas')
```

```
## [1] "Var del contrato divisas"
```

```
print(VaRTotalfr1_acc_div)
```

```
##        2%
```

```
## -437.5266
```

```
print("CVar Empírico")
```

```
## [1] "CVar Empírico"
```

```
print('CVar total')
```

```
## [1] "CVar total"
```

```
print(CVaRTotal_acc_div)
```

```
## [1] -6140.43
```

```
print('CVar por posición')
```

```
## [1] "CVar por posición"
```

```
print(cbind(titles,t(CVaRCont_acc_div)))
```

```
##      titles
## [1,] "EURUSD.X"    "-194.826466317777"
## [2,] "GBPUSD.X"    "-196.815212979911"
## [3,] "USDMXN.X"    "-439.721793659569"
## [4,] "AMXL.MX"     "-3418.44590580575"
## [5,] "GCARSOA1.MX" "-3396.45198751714"
## [6,] "WALMEX.MX"   "-3235.35595300484"
```

```
print('CVar del contrato acciones')
```

```
## [1] "CVar del contrato acciones"
```

```
print(CVaRTotalfr2_acc_div)
```

```
## [1] -6123.909
```

```
print('CVar del contrato divisas')
```

```
## [1] "CVar del contrato divisas"
```

```
print(CVaRTotalfr1_acc_div)
```

```
## [1] -519.4661
```

Bonos

```
#Cálculo de matriz de pérdidas y ganancias BONDES
#dimensión
m=count(N_bd)    #PASO CLAVE
X_s_bd=matrix(0,Ns,n_if[4]) #Factores de riesgo simulados con base en DeltaX_s x0*(1+Delta_Xs)   #PASO
V_bd=matrix(0,Ns,m)
Vfr1_bd=matrix(0,Ns,m)
Vfr2_bd=matrix(0,Ns,m)
Vfr3_bd=matrix(0,Ns,m)
PG_bd=matrix(0,Ns,m) #Pèrdidas y ganancias
PGfr1_bd=matrix(0,Ns,m)
PGfr2_bd=matrix(0,Ns,m)
PGfr3_bd=matrix(0,Ns,m)
PGT_bd=matrix(0,Ns,1)
PGfr1T_bd=matrix(0,Ns,1)
```

```
PGfr2T_bd=matrix(0,Ns,1)
PGfr3T_bd=matrix(0,Ns,1)

DeltaX_s_bd=DeltaX_s[,sum(n_if[1:3],1):sum(n_if[1:4])]    #PASO CLAVE
x0_bd=X_bd_ext[1,]    #PASO CLAVE

for (i in 1:Ns)
{
  X_s_bd[i,]=x0_bd*(1+DeltaX_s_bd[i,])
  #PASO CLAVE
  V_bd[i,]=bondeD(contratos_bdmT, nominal_bdm, X_s_bd[i,1], plazocupon_bdmT, VTplazos_bdm, X_s_bd[i,(n_
  #PASO CLAVE
  Vfr1_bd[i,]=bondeD(contratos_bdmT, nominal_bdm, X_s_bd[i,1], plazocupon_bdmT, VTplazos_bdm, x0_bd[(n_
  #PASO CLAVE
  Vfr2_bd[i,]=bondeD(contratos_bdmT, nominal_bdm, x0_bd[1], plazocupon_bdmT, VTplazos_bdm, X_s_bd[i,(n_
  #PASO CLAVE
  Vfr3_bd[i,]=bondeD(contratos_bdmT, nominal_bdm, x0_bd[1], plazocupon_bdmT, VTplazos_bdm, x0_bd[(n_if[
  #PASO CLAVE
  PG_bd[i,]=V_bd[i,]-V0_bd
  PGfr1_bd[i,]=Vfr1_bd[i,]-V0_bd
  PGfr2_bd[i,]=Vfr2_bd[i,]-V0_bd
  PGfr3_bd[i,]=Vfr3_bd[i,]-V0_bd
  PGT_bd[i,]=sum(PG_bd[i,])
  PGfr1T_bd[i,]=sum(PGfr1_bd[i,])
  PGfr2T_bd[i,]=sum(PGfr2_bd[i,])
  PGfr3T_bd[i,]=sum(PGfr3_bd[i,])
}



PG_bd[1:5,]
```

```
## [1] 2185.0592 -864.5564 -419.5199  961.2013 -624.5700
```

```
PGfr1_bd[1:5,]
```

```
## [1] 1559.39084 -533.68224   98.86811 1633.50846 -440.64043
```

```
PGfr2_bd[1:5,]
```

```
## [1]  658.6621 -403.0981 -401.3330 -948.8903 -365.4637
```

```
PGT_bd[1:5,]
```

```
## [1] 2185.0592 -864.5564 -419.5199  961.2013 -624.5700
```

```
#VaR por posición
VaRCont_bd=matrix(0,1,m)
VaRfr1_bd=matrix(0,1,m)
VaRfr2_bd=matrix(0,1,m)
VaRfr3_bd=matrix(0,1,m)
CVaRCont_bd=matrix(0,1,m)
CVaRfr1_bd=matrix(0,1,m)
CVaRfr2_bd=matrix(0,1,m)
CVaRfr3_bd=matrix(0,1,m)
for (i in (1:m))
```

```
{
  VaRCont_bd[i]=quantile(PG_bd[,i],1-alpha,Ns)
  VaRfr1_bd[i]=quantile(PGfr1_bd[,i],1-alpha,Ns)
  VaRfr2_bd[i]=quantile(PGfr2_bd[,i],1-alpha,Ns)
  VaRfr3_bd[i]=quantile(PGfr3_bd[,i],1-alpha,Ns)
  CVaRfr1_bd[i]= mean(merge(which(PGfr1_bd[,i]<VaRfr1_bd[i]),cbind(seq(1,Ns),PGfr1_bd[,i]), by.x=1,by.y=
  CVaRfr2_bd[i]= mean(merge(which(PGfr2_bd[,i]<VaRfr2_bd[i]),cbind(seq(1,Ns),PGfr2_bd[,i]), by.x=1,by.y=
  CVaRfr3_bd[i]= mean(merge(which(PGfr3_bd[,i]<VaRfr3_bd[i]),cbind(seq(1,Ns),PGfr3_bd[,i]), by.x=1,by.y=
  CVaRCont_bd[i]= mean(merge(which(PG_bd[,i]<VaRCont_bd[i]),cbind(seq(1,Ns),PG_bd[,i]), by.x=1,by.y=1)[
}
Met
```

```
## [1] "Cholesky Normal"
```

```
VaRCont_bd
```

```
##           [,1]
## [1,] -2730.175
```

```
VaRfr1_bd
```

```
##           [,1]
## [1,] -2220.105
```

```
VaRfr2_bd
```

```
##           [,1]
## [1,] -1656.862
```

```
CVaRCont_bd
```

```
##          [,1]
## [1,] -3253.93
```

```
CVaRfr1_bd
```

```
##           [,1]
## [1,] -2659.605
```

```
CVaRfr2_bd
```

```
##           [,1]
## [1,] -1943.181
```

```
#VaR Total
#Met
VaRTotal_bd=quantile(PGT_bd,1-alpha,Ns)
CVaRTotal_bd= mean(merge(which(PGT_bd<VaRTotal_bd),cbind(seq(1,Ns),PGT_bd), by.x=1,by.y=1)[,2])
VaRTotalfr1_bd=quantile(PGfr1T_bd,1-alpha,Ns)
CVaRTotalfr1_bd= mean(PGfr1T_bd[which(PGfr1T_bd<VaRTotalfr1_bd),])
VaRTotalfr2_bd=quantile(PGfr2T_bd,1-alpha,Ns)
CVaRTotalfr2_bd= mean(PGfr2T_bd[which(PGfr2T_bd<VaRTotalfr2_bd),])
VaRTotalfr3_bd=quantile(PGfr3T_bd,1-alpha,Ns)
CVaRTotalfr3_bd= mean(PGfr3T_bd[which(PGfr2T_bd<VaRTotalfr2_bd),])


#cbind(VaRTotal_bd,sum(V0_bd), VaRCont_bd, V0_bd)
#cbind(CVaRTotal_bd,sum(V0_bd), CVaRCont_bd, V0_bd)
#cbind(VaRTotal_bd,VaRTotalfr1_bd,VaRTotalfr2_bd,VaRTotalfr3_bd)
```

```
#cbind(CVaRTotal_bd,CVaRTotalfr1_bd,CVaRTotalfr2_bd,CVaRTotalfr3_bd)
lista<-c('Tasa descuento','tasa fondeo', 'sobretasa')
```

```
print('Var total')
```

```
## [1] "Var total"
```

```
print(VaRTotal_bd)
```

```
##        2%
## -2730.175
```

```
print('Var por factor de riesgo')
```

```
## [1] "Var por factor de riesgo"
```

```
print(rbind(lista,t(c(VaRTotalfr1_bd,VaRTotalfr2_bd,VaRTotalfr3_bd))))
```

```
##       2%                   2%                  2%
## lista "Tasa descuento"     "tasa fondeo"       "sobretasa"
##       "-2220.10531078574" "-1656.8616857304" "-689.18759323525"
```

```
print('CVar total')
```

```
## [1] "CVar total"
```

```
print(CVaRTotal_bd)
```

```
## [1] -3253.93
```

```
print('CVar por factor de riesgo')
```

```
## [1] "CVar por factor de riesgo"
```

```
print(rbind(lista,t(c(CVaRTotalfr1_bd,CVaRTotalfr2_bd,CVaRTotalfr3_bd))))
```

```
##       [,1]                 [,2]                 [,3]
## lista "Tasa descuento"     "tasa fondeo"        "sobretasa"
##       "-2659.60463643586" "-1943.18121011898" "-2.04451032649085"
```

Forwards de Divisas

```
#Cálculo de matriz de pérdidas y ganancias FUTUROS TDC
#dimensión
m=ncol(plazos_fwd)  #PASO CLAVE
X_s_fwtdc=matrix(0,Ns,n_if[5]) #Factores de riesgo simulados con base en DeltaX_s x0*(1+Delta_Xs) #PASO
V_fwtdc=matrix(0,Ns,m)
Vfr1_fwtdc=matrix(0,Ns,m)
Vfr2_fwtdc=matrix(0,Ns,m)
Vfr3_fwtdc=matrix(0,Ns,m)
PG_fwtdc=matrix(0,Ns,m) #Pèrdidas y ganancias
PGfr1_fwtdc=matrix(0,Ns,m)
PGfr2_fwtdc=matrix(0,Ns,m)
PGfr3_fwtdc=matrix(0,Ns,m)
PGT_fwtdc=matrix(0,Ns,1)
PGfr1T_fwtdc=matrix(0,Ns,1)
PGfr2T_fwtdc=matrix(0,Ns,1)
PGfr3T_fwtdc=matrix(0,Ns,1)
```

```
DeltaX_s_fwtdc=DeltaX_s[,sum(n_if[1:4],1):sum(n_if[1:5])]  #PASO CLAVE
x0_fwtdc=X_futtdc[1,] #PASO CLAVE

for (i in 1:Ns)
{
  X_s_fwtdc[i,]=x0_fwtdc*(1+DeltaX_s_fwtdc[i,])
  #PASO CLAVE
  V_fwtdc[i,]=futuroTC(plazos_fwd,X_s_fwtdc[i,1:((n_if[5]-1)/2)],X_s_fwtdc[i,((n_if[5]-1)/2+1):(n_if[5]
  #PASO CLAVE
  Vfr1_fwtdc[i,]=futuroTC(plazos_fwd,X_s_fwtdc[i,1:((n_if[5]-1)/2)],x0_fwtdc[((n_if[5]-1)/2+1):(n_if[5]
  #PASO CLAVE
  Vfr2_fwtdc[i,]=futuroTC(plazos_fwd,x0_fwtdc[1:((n_if[5]-1)/2)],X_s_fwtdc[i,((n_if[5]-1)/2+1):(n_if[5]
  #PASO CLAVE
  Vfr3_fwtdc[i,]=futuroTC(plazos_fwd,x0_fwtdc[1:((n_if[5]-1)/2)],x0_fwtdc[((n_if[5]-1)/2+1):(n_if[5]-1)]
  PG_fwtdc[i,]=V_fwtdc[i,]-V0_fwtdc
  PGfr1_fwtdc[i,]=Vfr1_fwtdc[i,]-V0_fwtdc
  PGfr2_fwtdc[i,]=Vfr2_fwtdc[i,]-V0_fwtdc
  PGfr3_fwtdc[i,]=Vfr3_fwtdc[i,]-V0_fwtdc
  PGT_fwtdc[i,]=sum(PG_fwtdc[i,])
  PGfr1T_fwtdc[i,]=sum(PGfr1_fwtdc[i,])
  PGfr2T_fwtdc[i,]=sum(PGfr2_fwtdc[i,])
  PGfr3T_fwtdc[i,]=sum(PGfr3_fwtdc[i,])
}



PG_fwtdc[1:5,]
```

```
## [1]  -5.135641  -5.179339 -11.192749  15.535999   6.969130
```

```
PGfr1_fwtdc[1:5,]
```

```
## [1] -0.001222898  0.002801145 -0.003939159  0.007678432  0.005745857
```

```
PGfr2_fwtdc[1:5,]
```

```
## [1]  0.002057541 -0.004712965  0.006627675 -0.012919089 -0.009667494
```

```
PGT_fwtdc[1:5,]
```

```
## [1]  -5.135641  -5.179339 -11.192749  15.535999   6.969130
```

```
#VaR por posición
VaRCont_fwtdc=matrix(0,1,m)
VaRfr1_fwtdc=matrix(0,1,m)
VaRfr2_fwtdc=matrix(0,1,m)
VaRfr3_fwtdc=matrix(0,1,m)
CVaRCont_fwtdc=matrix(0,1,m)
CVaRfr1_fwtdc=matrix(0,1,m)
CVaRfr2_fwtdc=matrix(0,1,m)
CVaRfr3_fwtdc=matrix(0,1,m)
for (i in (1:m))
{
  VaRCont_fwtdc[i]=quantile(PG_fwtdc[,i],1-alpha,Ns)
  VaRfr1_fwtdc[i]=quantile(PGfr1_fwtdc[,i],1-alpha,Ns)
  VaRfr2_fwtdc[i]=quantile(PGfr2_fwtdc[,i],1-alpha,Ns)
```

```
  VaRfr3_fwtdc[i]=quantile(PGfr3_fwtdc[,i],1-alpha,Ns)
  CVaRfr1_fwtdc[i]= mean(merge(which(PGfr1_fwtdc[,i]<VaRfr1_fwtdc[i]),cbind(seq(1,Ns),PGfr1_fwtdc[,i]),
  CVaRfr2_fwtdc[i]= mean(merge(which(PGfr2_fwtdc[,i]<VaRfr2_fwtdc[i]),cbind(seq(1,Ns),PGfr2_fwtdc[,i]),
  CVaRfr3_fwtdc[i]= mean(merge(which(PGfr3_fwtdc[,i]<VaRfr3_fwtdc[i]),cbind(seq(1,Ns),PGfr3_fwtdc[,i]),
  CVaRCont_fwtdc[i]= mean(merge(which(PG_fwtdc[,i]<VaRCont_fwtdc[i]),cbind(seq(1,Ns),PG_fwtdc[,i]), by.
}
#Met
#VaRCont_fwtdc
#VaRfr1_fwtdc
#VaRfr2_fwtdc
#CVaRCont_fwtdc
#CVaRfr1_fwtdc
#CVaRfr2_fwtdc


#VaR Total
Met
```

```
## [1] "Cholesky Normal"
```

```
VaRTotal_fwtdc=quantile(PGT_fwtdc,1-alpha,Ns)
CVaRTotal_fwtdc= mean(merge(which(PGT_fwtdc<VaRTotal_fwtdc),cbind(seq(1,Ns),PGT_fwtdc), by.x=1,by.y=1)[
VaRTotalfr1_fwtdc=quantile(PGfr1T_fwtdc,1-alpha,Ns)
CVaRTotalfr1_fwtdc= mean(PGfr1T_fwtdc[which(PGfr1T_fwtdc<VaRTotalfr1_fwtdc),])
VaRTotalfr2_fwtdc=quantile(PGfr2T_fwtdc,1-alpha,Ns)
CVaRTotalfr2_fwtdc= mean(PGfr2T_fwtdc[which(PGfr2T_fwtdc<VaRTotalfr2_fwtdc),])
VaRTotalfr3_fwtdc=quantile(PGfr3T_fwtdc,1-alpha,Ns)
CVaRTotalfr3_fwtdc= mean(PGfr3T_fwtdc[which(PGfr2T_fwtdc<VaRTotalfr2_fwtdc),])


#cbind(VaRTotal_fwtdc,sum(VO_fwtdc), VaRCont_fwtdc, VO_fwtdc)
#cbind(CVaRTotal_fwtdc,sum(VO_fwtdc), CVaRCont_fwtdc, VO_fwtdc)
cbind(VaRTotal_fwtdc,VaRTotalfr1_fwtdc,VaRTotalfr2_fwtdc,VaRTotalfr3_fwtdc)
```

```
##    VaRTotal_fwtdc VaRTotalfr1_fwtdc VaRTotalfr2_fwtdc VaRTotalfr3_fwtdc
## 2%       -43.1885       -0.01923432       -0.03078657         -43.18818
```

```
cbind(CVaRTotal_fwtdc,CVaRTotalfr1_fwtdc,CVaRTotalfr2_fwtdc,CVaRTotalfr3_fwtdc)
```

```
##      CVaRTotal_fwtdc CVaRTotalfr1_fwtdc CVaRTotalfr2_fwtdc CVaRTotalfr3_fwtdc
## [1,]       -51.56703        -0.02277249        -0.03533381           2.716522
```

Forwards IPC

```
#Cálculo de matriz de pérdidas y ganancias FUTUROS IPC
m=ncol(plazos_fwd_ind)  #PASO CLAVE
X_s_fwind=matrix(0,Ns,n_if[6]) #Factores de riesgo simulados con base en DeltaX_s x0*(1+Delta_Xs) #PASO
V_fwind=matrix(0,Ns,m)
Vfr1_fwind=matrix(0,Ns,m)
Vfr2_fwind=matrix(0,Ns,m)
Vfr3_fwind=matrix(0,Ns,m)
PG_fwind=matrix(0,Ns,m) #Pèrdidas y ganancias
PGfr1_fwind=matrix(0,Ns,m)
PGfr2_fwind=matrix(0,Ns,m)
PGfr3_fwind=matrix(0,Ns,m)
```

```
PGT_fwind=matrix(0,Ns,1)
PGfr1T_fwind=matrix(0,Ns,1)
PGfr2T_fwind=matrix(0,Ns,1)
PGfr3T_fwind=matrix(0,Ns,1)

DeltaX_s_fwind=DeltaX_s[,sum(n_if[1:5],1):sum(n_if[1:6])]  #PASO CLAVE
x0_fwind=X_futind[1,] #PASO CLAVE


for (i in 1:Ns)
{
  X_s_fwind[i,]=x0_fwind*(1+DeltaX_s_fwind[i,])
  #PASO CLAVE
  V_fwind[i,]=futuroTC(plazos_fwd_ind,X_s_fwind[i,1:(n_if[6]/3)],X_s_fwind[i,(n_if[6]/3+1):(n_if[6]*2/3)
  Vfr1_fwind[i,]=futuroTC(plazos_fwd_ind,X_s_fwind[i,1:(n_if[6]/3)],x0_fwind[(n_if[6]/3+1):(n_if[6]*2/3)
  #PASO CLAVE
  Vfr2_fwind[i,]=futuroTC(plazos_fwd_ind,x0_fwind[1:(n_if[6]/3)],X_s_fwind[i,(n_if[6]/3+1):(n_if[6]*2/3)
  #PASO CLAVE
  Vfr3_fwind[i,]=futuroTC(plazos_fwd_ind,x0_fwind[1:(n_if[6]/3)],x0_fwind[(n_if[6]/3+1):(n_if[6]*2/3)],
  PG_fwind[i,]=V_fwind[i,]-V0_fwind
  PGfr1_fwind[i,]=Vfr1_fwind[i,]-V0_fwind
  PGfr2_fwind[i,]=Vfr2_fwind[i,]-V0_fwind
  PGfr3_fwind[i,]=Vfr3_fwind[i,]-V0_fwind
  PGT_fwind[i,]=sum(PG_fwind[i,])
  PGfr1T_fwind[i,]=sum(PGfr1_fwind[i,])
  PGfr2T_fwind[i,]=sum(PGfr2_fwind[i,])
  PGfr3T_fwind[i,]=sum(PGfr3_fwind[i,])
}


PG_fwind[1:5,]
```

```
## [1]   2729.227  -6388.544   9102.596 -18366.598 -13341.855
```

```
#PGfr1_fwind[1:5,]
#PGfr2_fwind[1:5,]
#PGT_fwind[1:5,]



#VaR por posición
VaRCont_fwind=matrix(0,1,m)
VaRfr1_fwind=matrix(0,1,m)
VaRfr2_fwind=matrix(0,1,m)
VaRfr3_fwind=matrix(0,1,m)
CVaRCont_fwind=matrix(0,1,m)
CVaRfr1_fwind=matrix(0,1,m)
CVaRfr2_fwind=matrix(0,1,m)
CVaRfr3_fwind=matrix(0,1,m)
for (i in (1:m))
{
  VaRCont_fwind[i]=quantile(PG_fwind[,i],1-alpha,Ns)
  VaRfr1_fwind[i]=quantile(PGfr1_fwind[,i],1-alpha,Ns)
  VaRfr2_fwind[i]=quantile(PGfr2_fwind[,i],1-alpha,Ns)
```

```
  VaRfr3_fwind[i]=quantile(PGfr3_fwind[,i],1-alpha,Ns)
  CVaRfr1_fwind[i]= mean(merge(which(PGfr1_fwind[,i]<VaRfr1_fwind[i]),cbind(seq(1,Ns),PGfr1_fwind[,i]),
  CVaRfr2_fwind[i]= mean(merge(which(PGfr2_fwind[,i]<VaRfr2_fwind[i]),cbind(seq(1,Ns),PGfr2_fwind[,i]),
  CVaRfr3_fwind[i]= mean(merge(which(PGfr3_fwind[,i]<VaRfr3_fwind[i]),cbind(seq(1,Ns),PGfr3_fwind[,i]),
  CVaRCont_fwind[i]= mean(merge(which(PG_fwind[,i]<VaRCont_fwind[i]),cbind(seq(1,Ns),PG_fwind[,i]), by.
}
#Met
#VaRCont_fwind
#VaRfr1_fwind
#VaRfr2_fwind
#CVaRCont_fwind
#CVaRfr1_fwind
#CVaRfr2_fwind


#VaR Total
Met
```

```
## [1] "Cholesky Normal"
```

```
VaRTotal_fwind=quantile(PGT_fwind,1-alpha,Ns)
CVaRTotal_fwind= mean(merge(which(PGT_fwind<VaRTotal_fwind),cbind(seq(1,Ns),PGT_fwind), by.x=1,by.y=1)[
VaRTotalfr1_fwind=quantile(PGfr1T_fwind,1-alpha,Ns)
CVaRTotalfr1_fwind= mean(PGfr1T_fwind[which(PGfr1T_fwind<VaRTotalfr1_fwind),])
VaRTotalfr2_fwind=quantile(PGfr2T_fwind,1-alpha,Ns)
CVaRTotalfr2_fwind= mean(PGfr2T_fwind[which(PGfr2T_fwind<VaRTotalfr2_fwind),])
VaRTotalfr3_fwind=quantile(PGfr3T_fwind,1-alpha,Ns)
CVaRTotalfr3_fwind= mean(PGfr3T_fwind[which(PGfr3T_fwind<VaRTotalfr3_fwind),])


#print(cbind(VaRTotal_fwind,sum(VO_fwind), VaRCont_fwind, VO_fwind))
#print(cbind(CVaRTotal_fwind,sum(VO_fwind), CVaRCont_fwind, VO_fwind))
print(cbind(VaRTotal_fwind,VaRTotalfr1_fwind,VaRTotalfr2_fwind,VaRTotalfr3_fwind))
```

```
##    VaRTotal_fwind VaRTotalfr1_fwind VaRTotalfr2_fwind VaRTotalfr3_fwind
## 2%      -43335.03                 0         -551.7975         -43481.45
```

```
cbind(CVaRTotal_fwind,CVaRTotalfr1_fwind,CVaRTotalfr2_fwind,CVaRTotalfr3_fwind)
```

```
##      CVaRTotal_fwind CVaRTotalfr1_fwind CVaRTotalfr2_fwind CVaRTotalfr3_fwind
## [1,]       -49836.53                NaN           -653.503          -49903.64
```

Swaps

```
#Cálculo de matriz de pérdidas y ganancias SWAP

#riesgo del swap
m=ncol(N)    #PASO CLAVE
X_s_sw=matrix(0,Ns,n_if[7]) #Factores de riesgo simulados con base en DeltaX_s x0*(1+Delta_Xs)   #PASO
V_sw=matrix(0,Ns,m)
Vfr1_sw=matrix(0,Ns,m)
Vfr2_sw=matrix(0,Ns,m)
PG_sw=matrix(0,Ns,m) #Pèrdidas y ganancias
PGfr1_sw=matrix(0,Ns,m)
PGfr2_sw=matrix(0,Ns,m)
```

```
PGT_sw=matrix(0,Ns,1)
PGfr1T_sw=matrix(0,Ns,1)
PGfr2T_sw=matrix(0,Ns,1)

DeltaX_s_sw=DeltaX_s[,sum(n_if[1:6],1):sum(n_if[1:7])]    #PASO CLAVE
x0_sw=as.numeric(c(XtfwdT[1,],Xvp[1,]))   #PASO CLAVE

for (i in 1:Ns)
{
  X_s_sw[i,]=x0_sw*(1+DeltaX_s_sw[i,])
  #PASO CLAVE
  V_sw[i,]=swap(por_swT, contratos_swT, nominal_swT, X_s_sw[i,1:(n_if[7]/2)], tasafija_swT, plazocupon_s
  #PASO CLAVE
  Vfr1_sw[i,]=swap(por_swT, contratos_swT, nominal_swT,X_s_sw[i,1:(n_if[7]/2)], tasafija_swT, plazocupo
  #PASO CLAVE
  Vfr2_sw[i,]=swap(por_swT, contratos_swT, nominal_swT,  XtfwdT[1,], tasafija_swT, plazocupon_swT, VTpla
  #PASO CLAVE
  PG_sw[i,]=V_sw[i,]-V0_sw
  PGfr1_sw[i,]=Vfr1_sw[i,]-V0_sw
  PGfr2_sw[i,]=Vfr2_sw[i,]-V0_sw
  PGT_sw[i,]=sum(PG_sw[i,])
  PGfr1T_sw[i,]=sum(PGfr1_sw[i,])
  PGfr2T_sw[i,]=sum(PGfr2_sw[i,])
}



PG_sw[1:5,]
```

```
##           [,1]      [,2]
## [1,]  8103.428 -4031.114
## [2,] -1354.789 -2347.541
## [3,]  8446.935 -1838.542
## [4,]  1184.575  2697.869
## [5,]  9321.273 -6588.824
```

```
PGfr1_sw[1:5,]
```

```
##           [,1]      [,2]
## [1,]  8117.194 -4033.406
## [2,] -1395.201 -2354.609
## [3,]  8490.237 -1828.909
## [4,]  1164.942  2694.019
## [5,]  9292.279 -6597.023
```

```
PGfr2_sw[1:5,]
```

```
##           [,1]      [,2]
## [1,] -14.26269  2.389685
## [2,]  37.19861  6.938272
## [3,] -48.78475 -9.580698
## [4,]  22.04588  4.215708
## [5,]  28.96517  7.164373
```

```
PGT_sw[1:5,]
```

```
## [1]  4072.314 -3702.330  6608.393  3882.443  2732.449
```

```
#VaR por posición
VaRCont_sw=matrix(0,1,m)
VaRfr1_sw=matrix(0,1,m)
VaRfr2_sw=matrix(0,1,m)
CVaRCont_sw=matrix(0,1,m)
CVaRfr1_sw=matrix(0,1,m)
CVaRfr2_sw=matrix(0,1,m)
for (i in (1:m))
{
  VaRCont_sw[i]=quantile(PG_sw[,i],1-alpha,Ns)
  VaRfr1_sw[i]=quantile(PGfr1_sw[,i],1-alpha,Ns)
  VaRfr2_sw[i]=quantile(PGfr2_sw[,i],1-alpha,Ns)
  CVaRfr1_sw[i]= mean(merge(which(PGfr1_sw[,i]<VaRfr1_sw[i]),cbind(seq(1,Ns),PGfr1_sw[,i]), by.x=1,by.y
  CVaRfr2_sw[i]= mean(merge(which(PGfr2_sw[,i]<VaRfr2_sw[i]),cbind(seq(1,Ns),PGfr2_sw[,i]), by.x=1,by.y
  CVaRCont_sw[i]= mean(merge(which(PG_sw[,i]<VaRCont_sw[i]),cbind(seq(1,Ns),PG_sw[,i]), by.x=1,by.y=1)[
}
#Met
#VaRCont_sw
#VaRfr1_sw
#VaRfr2_sw
#CVaRCont_sw
#CVaRfr1_sw
#CVaRfr2_sw


#VaR Total
Met
```

```
## [1] "Cholesky Normal"
```

```
VaRTotal_sw=quantile(PGT_sw,1-alpha,Ns)
CVaRTotal_sw= mean(merge(which(PGT_sw<VaRTotal_sw),cbind(seq(1,Ns),PGT_sw), by.x=1,by.y=1)[,2])
VaRTotalfr1_sw=quantile(PGfr1T_sw,1-alpha,Ns)
CVaRTotalfr1_sw= mean(PGfr1T_sw[which(PGfr1T_sw<VaRTotalfr1_sw),])
VaRTotalfr2_sw=quantile(PGfr2T_sw,1-alpha,Ns)
CVaRTotalfr2_sw= mean(PGfr2T_sw[which(PGfr2T_sw<VaRTotalfr2_sw),])


cbind(VaRTotal_sw, VaRCont_sw)
```

```
##    VaRTotal_sw
## 2%   -12074.04 -16338.21 -7344.958
```

```
cbind(CVaRTotal_sw, CVaRCont_sw)
```

```
##      CVaRTotal_sw
## [1,]    -14500.45 -19709.75 -8570
```

```
cbind(VaRTotal_sw,VaRTotalfr1_sw,VaRTotalfr2_sw)
```

```
##    VaRTotal_sw VaRTotalfr1_sw VaRTotalfr2_sw
## 2%   -12074.04      -12069.98      -59.65231
```

```
cbind(CVaRTotal_sw,CVaRTotalfr1_sw,CVaRTotalfr2_sw)
```

```
##      CVaRTotal_sw CVaRTotalfr1_sw CVaRTotalfr2_sw
```

```
## [1,]    -14500.45       -14478.58       -71.44649
```

Opciones de tasa de interés

```
#Cálculo de matriz de pérdidas y ganancias Opciones Tasa de interés
#dimensión
m=ncol(plazos_oir)  #PASO CLAVE
X_s_oir=matrix(0,Ns,n_if[8]) #Factores de riesgo simulados con base en DeltaX_s x0*(1+Delta_Xs) #PASO C
V_oir=matrix(0,Ns,m)
Vfr1_oir=matrix(0,Ns,m)
Vfr2_oir=matrix(0,Ns,m)
Vfr3_oir=matrix(0,Ns,m)
PG_oir=matrix(0,Ns,m) #Pèrdidas y ganancias
PGfr1_oir=matrix(0,Ns,m)
PGfr2_oir=matrix(0,Ns,m)
PGfr3_oir=matrix(0,Ns,m)
PGT_oir=matrix(0,Ns,1)
PGfr1T_oir=matrix(0,Ns,1)
PGfr2T_oir=matrix(0,Ns,1)
PGfr3T_oir=matrix(0,Ns,1)

DeltaX_s_oir=DeltaX_s[,sum(n_if[1:7],1):sum(n_if[1:8])]  #PASO CLAVE
x0_oir=X_oir[1,] #PASO CLAVE

for (i in 1:Ns)
{
  X_s_oir[i,]=x0_oir*(1+DeltaX_s_oir[i,])
  #PASO CLAVE
  V_oir[i,]=  opctint(X_s_oir[i,(1:(n_if[8]/3))],X_s_oir[i,((n_if[8]/3+1):(n_if[8]/3*2))],K_oir,X_s_oir
  #PASO CLAVE
  Vfr1_oir[i,]=opctint(X_s_oir[i,(1:(n_if[8]/3))],x0_oir[((n_if[8]/3+1):(n_if[8]/3*2))],K_oir,x0_oir[((
  #PASO CLAVE
  Vfr2_oir[i,]=opctint(x0_oir[(1:(n_if[8]/3))],X_s_oir[i,((n_if[8]/3+1):(n_if[8]/3*2))],K_oir,x0_oir[((
  #PASO CLAVE
  Vfr3_oir[i,]=opctint(x0_oir[(1:(n_if[8]/3))],x0_oir[((n_if[8]/3+1):(n_if[8]/3*2))],K_oir,X_s_oir[i,((
  PG_oir[i,]=V_oir[i,]-V0_oir
  PGfr1_oir[i,]=Vfr1_oir[i,]-V0_oir
  PGfr2_oir[i,]=Vfr2_oir[i,]-V0_oir
  PGfr3_oir[i,]=Vfr3_oir[i,]-V0_oir
  PGT_oir[i,]=sum(PG_oir[i,])
  PGfr1T_oir[i,]=sum(PGfr1_oir[i,])
  PGfr2T_oir[i,]=sum(PGfr2_oir[i,])
  PGfr3T_oir[i,]=sum(PGfr3_oir[i,])
}



PG_oir[1:5,]
```

```
##              [,1]         [,2]
## [1,]  0.005551995 -0.006403271
## [2,] -0.006610336  0.011644513
## [3,] -0.003953663 -0.009864083
## [4,] -0.011648767 -0.005300373
```

```
## [5,]  0.003624579  0.010678981
```

```
PGfr1_oir[1:5,]
```

```
##                [,1]          [,2]
## [1,] -0.0002604747 -2.970655e-05
## [2,]  0.0005972171  6.807520e-05
## [3,] -0.0008384834 -9.566128e-05
## [4,]  0.0016390059  1.867059e-04
## [5,]  0.0012259144  1.396845e-04
```

```
PGfr2_oir[1:5,]
```

```
##               [,1]         [,2]
## [1,]  0.005019099 -0.006673027
## [2,] -0.005371114  0.012270634
## [3,] -0.005693626 -0.010729877
## [4,] -0.008234087 -0.003600981
## [5,]  0.006129748  0.011963521
```

```
PGT_oir[1:5,]
```

```
## [1] -0.0008512762  0.0050341767 -0.0138177460 -0.0169491401  0.0143035602
```

```
#VaR por posición
VaRCont_oir=matrix(0,1,m)
VaRfr1_oir=matrix(0,1,m)
VaRfr2_oir=matrix(0,1,m)
VaRfr3_oir=matrix(0,1,m)
CVaRCont_oir=matrix(0,1,m)
CVaRfr1_oir=matrix(0,1,m)
CVaRfr2_oir=matrix(0,1,m)
CVaRfr3_oir=matrix(0,1,m)
for (i in (1:m))
{
  VaRCont_oir[i]=quantile(PG_oir[,i],1-alpha,Ns)
  VaRfr1_oir[i]=quantile(PGfr1_oir[,i],1-alpha,Ns)
  VaRfr2_oir[i]=quantile(PGfr2_oir[,i],1-alpha,Ns)
  VaRfr3_oir[i]=quantile(PGfr3_oir[,i],1-alpha,Ns)
  CVaRfr1_oir[i]= mean(merge(which(PGfr1_oir[,i]<VaRfr1_oir[i]),cbind(seq(1,Ns),PGfr1_oir[,i]), by.x=1,
  CVaRfr2_oir[i]= mean(merge(which(PGfr2_oir[,i]<VaRfr2_oir[i]),cbind(seq(1,Ns),PGfr2_oir[,i]), by.x=1,
  CVaRfr3_oir[i]= mean(merge(which(PGfr3_oir[,i]<VaRfr3_oir[i]),cbind(seq(1,Ns),PGfr3_oir[,i]), by.x=1,
  CVaRCont_oir[i]= mean(merge(which(PG_oir[,i]<VaRCont_oir[i]),cbind(seq(1,Ns),PG_oir[,i]), by.x=1,by.y
}
#Met
#VaRCont_oir
#VaRfr1_oir
#VaRfr2_oir
#CVaRCont_oir
#CVaRfr1_oir
#CVaRfr2_oir


#VaR Total
#Met
VaRTotal_oir=quantile(PGT_oir,1-alpha,Ns)
CVaRTotal_oir= mean(merge(which(PGT_oir<VaRTotal_oir),cbind(seq(1,Ns),PGT_oir), by.x=1,by.y=1)[,2])
```

```
VaRTotalfr1_oir=quantile(PGfr1T_oir,1-alpha,Ns)
CVaRTotalfr1_oir= mean(PGfr1T_oir[which(PGfr1T_oir<VaRTotalfr1_oir),])
VaRTotalfr2_oir=quantile(PGfr2T_oir,1-alpha,Ns)
CVaRTotalfr2_oir= mean(PGfr2T_oir[which(PGfr2T_oir<VaRTotalfr2_oir),])
VaRTotalfr3_oir=quantile(PGfr3T_oir,1-alpha,Ns)
CVaRTotalfr3_oir= mean(PGfr3T_oir[which(PGfr2T_oir<VaRTotalfr2_oir),])


cbind(VaRTotal_oir, VaRCont_oir)
```

```
##    VaRTotal_oir
## 2%  -0.03552751 -0.02997257 -0.02117473
```

```
cbind(CVaRTotal_oir, CVaRCont_oir)
```

```
##      CVaRTotal_oir
## [1,]   -0.04101466 -0.03486608 -0.02523128
```

```
cbind(VaRTotal_oir,VaRTotalfr1_oir,VaRTotalfr2_oir,VaRTotalfr3_oir)
```

```
##    VaRTotal_oir VaRTotalfr1_oir VaRTotalfr2_oir VaRTotalfr3_oir
## 2%  -0.03552751    -0.004545478     -0.03450307     -0.01643817
```

```
cbind(CVaRTotal_oir,CVaRTotalfr1_oir,CVaRTotalfr2_oir,CVaRTotalfr3_oir)
```

```
##      CVaRTotal_oir CVaRTotalfr1_oir CVaRTotalfr2_oir CVaRTotalfr3_oir
## [1,]   -0.04101466     -0.005376724      -0.03970937        0.0024841
```

Riesgo total del portafolio

```
#Medición de riesgo por factor de riesgo de todo el portafolios
#Acciones
#1. Acciones
#2. Forward de IPC
PGPort_ACC=PGfr2T_acc_div+ PGfr3T_fwind #Pérdidas y ganancias
VaRPort_ACC=quantile(PGPort_ACC,1-alpha,Ns) #VaR
CVaRPort_ACC= mean(PGPort_ACC[which(PGPort_ACC<VaRPort_ACC)]) #CVaR

print(VaRPort_ACC)
```

```
##        2%
## -42441.05
```

```
print(CVaRPort_ACC)
```

```
## [1] -48997.04
```

```
#Tasa de Interés
#1. Dado que swaps y bondes son de tasa de interés usaremos PGT_bd y PGT_sw
#2. Para futuros usaremos PGfr1T_fwtdc y PGfr2T_fwtdc
PGPort_TI=PGT_bd+PGT_sw+PGfr1T_fwtdc+PGfr2T_fwtdc +PGfr2T_fwind+PGfr1T_fwind +PGfr1T_oir +PGfr2T_oir #P
VaRPort_TI=quantile(PGPort_TI,1-alpha,Ns) #VaR
CVaRPort_TI= mean(PGPort_TI[which(PGPort_TI<VaRPort_TI)]) #CVaR
print(VaRPort_TI)
```

```
##        2%
## -12816.81
```

```
print(CVaRPort_TI)
```

```
## [1] -15620.81
```

```
#Tipo de cambio
#1. Dado que swaps y bondes son de tasa de interés no usamos nada
#2. Para futuros usamos sólo PGfr3T_fwtdc
PGPort_TDC=PGfr1T_acc_div+PGfr3T_fwtdc  #Pérdidas y ganancias
VaRPort_TDC=quantile(PGPort_TDC,1-alpha,Ns) #VaR
CVaRPort_TDC= mean(PGPort_TDC[which(PGPort_TDC<VaRPort_TDC)]) #CVaR
print(VaRPort_TDC)
```

```
##        2%
## -445.2806
```

```
print(CVaRPort_TDC)
```

```
## [1] -528.455
```

```
#Volatilidad
#1. Sólo aplica la volatilidad de Opciones de tasa de interés

PGPort_VOL=PGfr3T_oir  #Pérdidas y ganancias
VaRPort_VOL=quantile(PGPort_VOL,1-alpha,Ns) #VaR
CVaRPort_VOL= mean(PGPort_VOL[which(PGPort_VOL<VaRPort_VOL)]) #CVaR

print(VaRPort_VOL)
```

```
##          2%
## -0.01643817
```

```
print(CVaRPort_VOL)
```

```
## [1] -0.01886745
```

```
#Medición de riesgo de todo el portafolios
#Sumar todos los PGT de todos los instrumentos

PGT_Port=PGPort_ACC+PGPort_TI+PGPort_TDC+PGPort_VOL
VaRTotal_Port=quantile(PGT_Port,1-alpha,Ns) #VaR
CVaRTotal_Port= mean(PGT_Port[which(PGT_Port<VaRTotal_Port)]) #CVaR
print(VaRTotal_Port)
```

```
##        2%
## -48423.71
```

```
print(CVaRTotal_Port)
```

```
## [1] -55912.14
```

```
print(V0T_port)
```

```
## [1] 1474917
```

Por Compenetes principales empírico

Por Cholesky Normal

```
#DIMENSION DE TODOS LOS INSTRUMENTOS
#Son 8 instrumentos financieros (9 si separamos acciones y divisas)
n_if=matrix(0,8,1)
n_if[1]=ncol(stock_prices_EQFX) #acciones y divisas
n_if[2]=ncol(x_bcc) #cetes
n_if[3]=ncol(X_bm) #bonos m
n_if[4]=ncol(X_bd_ext) #bonde
n_if[5]=ncol(X_futtdc) #fut tdc
n_if[6]=ncol(X_futind) #fut ind
n_if[7]=ncol(X_sw) #swaps
n_if[8]=ncol(X_oir) #opciones tasa de interés

#valor del portafolios

V0_port=cbind(V0_acc_div,V0_bcc, V0_bm, V0_bd, V0_fwtdc, V0_fwind, V0_sw, V0_oir) #contrato
V0T_port=sum(V0_port)



#INTEGRACIÓN DE TODOS LOS FACTORES DE RIESGO EN UNA MATRIZ
X_port=cbind(stock_prices_EQFX,x_bcc,X_bm,X_bd_ext,X_futtdc,X_futind,X_sw,X_oir) #Factores de riesgo de

#Cálculo de variaciones Delta_X DEL PORTAFOLIOS
DeltaX_port=as.matrix(X_port[1:(n-1),]/X_port[2:(n),]-1)
DeltaX_port[is.nan(DeltaX_port)] <- 0 #quitamos NaN
DeltaX_port[is.na(DeltaX_port)] <- 0 #quitamos Na
DeltaX_port[is.infinite(DeltaX_port)] <- 0 #quitamos Na


Ns=10000 #Definimos número de simulaciones
alpha=0.98 #Nivel de Confianza para las medidas de riesgo
ChCP=1  #0 Cholesky, 1 Componentes Principales
normempi=1 #0 normal, 1 empírico
eta=.85 #Varianza explicada  (sólo aplica para Componentes Principales)
mT=ncol(DeltaX_port)


VarDeltaX=if(ChCP==0){cor(DeltaX_port)}else{var(DeltaX_port)*(ncol(DeltaX_port)/(ncol(DeltaX_port)-1))}

if(ChCP==0) {
  CVarDeltaX=as.matrix(chol(VarDeltaX,pivot=TRUE))
  CVarDeltaX[is.nan(CVarDeltaX)] <- 0 #quitamos NaN
  CVarDeltaX[is.na(CVarDeltaX)] <- 0 #quitamos Na
  print(ncol(CVarDeltaX))
  print(ncol(VarDeltaX))
  M=matrix(0, Ns,mT)
  if(normempi==1)
  {
    for (i in 1:mT)
    {
      M[,i]=quantile(DeltaX_port[,i],runif(Ns))
    }
    Met='Cholesky Empírico'
```

```
  } else    {
    for (i in 1:mT)
    {
      M[,i]=rnorm(Ns)*sd(DeltaX_port[,i])*(mT/(mT-1))+mean(DeltaX_port[,i])
    }
    Met='Cholesky Normal'
  }
  DeltaX_s=M%*%CVarDeltaX  #Factores de riesgo simulados
} else { #componentes principales
  G<-eigen(VarDeltaX)      #eigenvectores
  g1<-G$values                  #eigenvalores
  g1p=g1/sum(g1)

  g1pac=cumsum(g1p) #suma acumulada de varianza

  k=which.max(g1pac>eta) #valor mínimo k

  y=as.matrix((DeltaX_port-colMeans(DeltaX_port)))%*%G$vectors[,1:k] #componentes principales
  #/sqrt(diag(var(DeltaX))))
  M=matrix(0, Ns,k)
  if(normempi==0)
  {
    for (i in 1:k)
    {
      M[,i]=rnorm(Ns)*sd(y[,i])+mean(y[,i])
    }
    Met='Componentes Principales Normales'
  } else    {
    for (i in 1:k)
    {
      M[,i]=quantile(y[,i],runif(Ns))
    }
    Met='Componentes Principales Empíricos'
  }
  DeltaX_s=M%*% t(G$vectors[,1:k]) #Factores de riesgo simulados
}
#Met
#DeltaX_port[1:5,]


#Met


Ms1='Se explica el'
Ms2= 'de varianza'
Ms3= 'de suma de var-cov'

Ms1
```

```
## [1] "Se explica el"
```

```
print(sum(diag(var(DeltaX_s)))/sum(diag(var(DeltaX_port)))) #Comprobación de matriz de varianza-cov vs
```

```
## [1] 0.7538594
```

```
Ms2
```

```
## [1] "de varianza"
```

```
Ms1
```

```
## [1] "Se explica el"
```

```
print(matrix(1,1,ncol(DeltaX_s))%*%var(DeltaX_s)%*%t(matrix(1,1,ncol(DeltaX_s)))/matrix(1,1,ncol(DeltaX
```

```
##           [,1]
## [1,] 0.8422098
```

```
Ms3
```

```
## [1] "de suma de var-cov"
```

```
print(mT)
```

```
## [1] 173
```

```
#print(k) #número de variables simuladas
#print(g1pac)
```

Medición del riesgo

```
#Medición de riesgo por instrumento, instrumento-factor de riesgo, instrumento - total


#Cálculo de matriz de pérdidas y ganancias Acciones y Divisas

#riesgo del acciones y divisas
m=n_if[1]   #PASO CLAVE
X_s_acc_div=matrix(0,Ns,n_if[1]) #Factores de riesgo simulados con base en DeltaX_s x0*(1+Delta_Xs)   #
V_acc_div=matrix(0,Ns,m) #valor simulado a nivel contrato de acciones y tdc
Vfr1_acc_div=matrix(0,Ns,m_fx) #valor simulado a nivel contrato de tdc
Vfr2_acc_div=matrix(0,Ns,m_acc) #valor simulado a nivel contrato de acciones
PG_acc_div=matrix(0,Ns,m) #Pèrdidas y ganancias
PGfr1_acc_div=matrix(0,Ns,m_fx)
PGfr2_acc_div=matrix(0,Ns,m_acc)
PGT_acc_div=matrix(0,Ns,1) #TOTAL ACC Y DIV
PGfr1T_acc_div=matrix(0,Ns,1) #total factor riesgo divisas
PGfr2T_acc_div=matrix(0,Ns,1) #total factor riesgo acciones

DeltaX_s_acc_div=DeltaX_s[,(1:n_if[1])]    #PASO CLAVE
x0_acc_div=stock_prices_EQFX[1,]   #PASO CLAVE

for (i in 1:Ns)
{
  X_s_acc_div[i,]=as.matrix(x0_acc_div*(1+DeltaX_s_acc_div[i,]))
  #PASO CLAVE
  V_acc_div[i,]=cbind(t(pos_fx),t(pos_eq))*X_s_acc_div[i,]
  #PASO CLAVE
  Vfr1_acc_div[i,]=t(pos_fx)*X_s_acc_div[i,1:m_fx]
  #PASO CLAVE
  Vfr2_acc_div[i,]=t(pos_eq)*X_s_acc_div[i,(m_fx+1):(m_fx+m_acc)]
  #PASO CLAVE
```

```
  PG_acc_div[i,]=as.matrix(V_acc_div[i,]-V0_acc_div)
  PGfr1_acc_div[i,]=as.matrix(Vfr1_acc_div[i,]-V0_acc_div[,1:m_fx])
  PGfr2_acc_div[i,]=as.matrix(Vfr2_acc_div[i,]-V0_acc_div[,(m_fx+1):(m_fx+m_acc)])
  PGT_acc_div[i,]=sum(PG_acc_div[i,])
  PGfr1T_acc_div[i,]=sum(PGfr1_acc_div[i,])
  PGfr2T_acc_div[i,]=sum(PGfr2_acc_div[i,])
}

print(V_acc_div[1:3,])
```

Acciones y divisas

```
##          [,1]     [,2]     [,3]      [,4]     [,5]     [,6]
## [1,] 16137.81 -16459.8 31423.27 -99441.95 62758.42 90907.54
## [2,] 16110.72 -16436.5 31307.60 -99440.31 62593.17 91431.94
## [3,] 16124.82 -16448.1 31370.57 -99441.35 62912.26 91044.63
```

```
print(Vfr1_acc_div[1:3,])
```

```
##          [,1]     [,2]     [,3]
## [1,] 16137.81 -16459.8 31423.27
## [2,] 16110.72 -16436.5 31307.60
## [3,] 16124.82 -16448.1 31370.57
```

```
print(Vfr2_acc_div[1:3,])
```

```
##           [,1]     [,2]     [,3]
## [1,] -99441.95 62758.42 90907.54
## [2,] -99440.31 62593.17 91431.94
## [3,] -99441.35 62912.26 91044.63
```

```
PG_acc_div[1:5,]
```

```
##            [,1]       [,2]       [,3]      [,4]       [,5]      [,6]
## [1,]  15.160255 -18.899347  55.271309  158.0511 -241.58457 -388.4636
## [2,] -11.928574   4.402825 -60.398950  159.6934 -406.82930  135.9338
## [3,]   2.176711  -7.197813   2.566183  158.6499  -87.73849 -251.3749
## [4,]   5.529595  -5.097024  45.797339  173.1300  122.36149 -378.0774
## [5,] -10.028343   6.587420  -8.915327 -387.9049 -198.24865 -236.2847
```

```
PGfr1_acc_div[1:5,]
```

```
##            [,1]       [,2]       [,3]
## [1,]  15.160255 -18.899347  55.271309
## [2,] -11.928574   4.402825 -60.398950
## [3,]   2.176711  -7.197813   2.566183
## [4,]   5.529595  -5.097024  45.797339
## [5,] -10.028343   6.587420  -8.915327
```

```
PGfr2_acc_div[1:5,]
```

```
##           [,1]       [,2]      [,3]
## [1,]  158.0511 -241.58457 -388.4636
## [2,]  159.6934 -406.82930  135.9338
## [3,]  158.6499  -87.73849 -251.3749
## [4,]  173.1300  122.36149 -378.0774
## [5,] -387.9049 -198.24865 -236.2847
```

```
PGT_acc_div[1:5,]
```

```
## [1] -420.46483 -179.12682 -182.91836  -36.35604 -834.79449
```

```
#VaR por posición
VaRCont_acc_div=matrix(0,1,m)
VaRfr1_acc_div=matrix(0,1,m_fx)
VaRfr2_acc_div=matrix(0,1,m_acc)
CVaRCont_acc_div=matrix(0,1,m)
CVaRfr1_acc_div=matrix(0,1,m_fx)
CVaRfr2_acc_div=matrix(0,1,m_acc)
for (i in (1:m))
{
  VaRCont_acc_div[i]=quantile(PG_acc_div[,i],1-alpha,Ns)
  CVaRCont_acc_div[i]= mean(merge(which(PG_acc_div[,i]<VaRCont_acc_div[i]),cbind(seq(1,Ns),PG_acc_div[,
  if (i<=m_fx)
  {
  VaRfr1_acc_div[i]=quantile(PGfr1_acc_div[,i],1-alpha,Ns)
  CVaRfr1_acc_div[i]= mean(merge(which(PGfr1_acc_div[,i]<VaRfr1_acc_div[i]),cbind(seq(1,Ns),PGfr1_acc_di
  }
  if (i<=m_acc)
  {
  VaRfr2_acc_div[i]=quantile(PGfr2_acc_div[,i],1-alpha,Ns)
  CVaRfr2_acc_div[i]= mean(merge(which(PGfr2_acc_div[,i]<VaRfr2_acc_div[i]),cbind(seq(1,Ns),PGfr2_acc_di
  }
}
#Met
#VaRCont_acc_div
#VaRfr1_acc_div
#VaRfr2_acc_div
#CVaRCont_acc_div
#CVaRfr1_acc_div
#CVaRfr2_acc_div


#VaR Total
#Met
VaRTotal_acc_div=quantile(PGT_acc_div,1-alpha,Ns)
CVaRTotal_acc_div= mean(merge(which(PGT_acc_div<VaRTotal_acc_div),cbind(seq(1,Ns),PGT_acc_div), by.x=1,
VaRTotalfr1_acc_div=quantile(PGfr1T_acc_div,1-alpha,Ns)
CVaRTotalfr1_acc_div= mean(PGfr1T_acc_div[which(PGfr1T_acc_div<VaRTotalfr1_acc_div),])
VaRTotalfr2_acc_div=quantile(PGfr2T_acc_div,1-alpha,Ns)
CVaRTotalfr2_acc_div= mean(PGfr2T_acc_div[which(PGfr2T_acc_div<VaRTotalfr2_acc_div),])

#print("Var Empírico")
#print(cbind(VaRTotal_acc_div,sum(V0_acc_div), VaRCont_acc_div, V0_acc_div))
#print("CVar Empírico")
#print(cbind(CVaRTotal_acc_div,sum(V0_acc_div), CVaRCont_acc_div, V0_acc_div))
#print("Var Normal")
#print(cbind(VaRTotal_acc_div,VaRTotalfr1_acc_div,,VaRTotalfr2_acc_div))
#print("CVar Normal")
#print(cbind(CVaRTotal_acc_div,CVaRTotalfr1_acc_div,CVaRTotalfr2_acc_div))
#print(sum(V0_acc_div))
titles<-c('EURUSD.X', 'GBPUSD.X', 'USDMXN.X', 'AMXL.MX', 'GCARSOA1.MX',  'WALMEX.MX')
```

```
print("Var Componentes principales empírico")
```

```
## [1] "Var Componentes principales empírico"
```

```
print('Var total')
```

```
## [1] "Var total"
```

```
print(VaRTotal_acc_div)
```

```
##        2%
## -772.4949
```

```
print('Var por posición')
```

```
## [1] "Var por posición"
```

```
print(cbind(titles,t(VaRCont_acc_div)))
```

```
##      titles
## [1,] "EURUSD.X"    "-28.2096895258772"
## [2,] "GBPUSD.X"    "-20.3133617620625"
## [3,] "USDMXN.X"    "-77.1265434967284"
## [4,] "AMXL.MX"     "-630.152412132138"
## [5,] "GCARSOA1.MX" "-824.174319410285"
## [6,] "WALMEX.MX"   "-557.159239225901"
```

```
print('Var del contrato acciones')
```

```
## [1] "Var del contrato acciones"
```

```
print(VaRTotalfr2_acc_div)
```

```
##        2%
## -735.0845
```

```
print('Var del contrato divisas')
```

```
## [1] "Var del contrato divisas"
```

```
print(VaRTotalfr1_acc_div)
```

```
##        2%
## -85.02496
```

```
print("CVar Empírico")
```

```
## [1] "CVar Empírico"
```

```
print('CVar total')
```

```
## [1] "CVar total"
```

```
print(CVaRTotal_acc_div)
```

```
## [1] -1085.954
```

```
print('CVar por posición')
```

```
## [1] "CVar por posición"
```

```
print(cbind(titles,t(CVaRCont_acc_div)))
```

```
##      titles
## [1,] "EURUSD.X"    "-33.0600468996965"
## [2,] "GBPUSD.X"    "-24.6486771238014"
## [3,] "USDMXN.X"    "-89.049541156378"
## [4,] "AMXL.MX"     "-822.563055105019"
## [5,] "GCARSOA1.MX" "-1003.65042655105"
## [6,] "WALMEX.MX"   "-676.516386362638"
```

```
print('CVar del contrato acciones')
```

```
## [1] "CVar del contrato acciones"
```

```
print(CVaRTotalfr2_acc_div)
```

```
## [1] -1061.009
```

```
print('CVar del contrato divisas')
```

```
## [1] "CVar del contrato divisas"
```

```
print(CVaRTotalfr1_acc_div)
```

```
## [1] -98.18355
```

Bonos

```
#Cálculo de matriz de pérdidas y ganancias BONDES
#dimensión
m=count(N_bd)    #PASO CLAVE
X_s_bd=matrix(0,Ns,n_if[4]) #Factores de riesgo simulados con base en DeltaX_s x0*(1+Delta_Xs)   #PASO
V_bd=matrix(0,Ns,m)
Vfr1_bd=matrix(0,Ns,m)
Vfr2_bd=matrix(0,Ns,m)
Vfr3_bd=matrix(0,Ns,m)
PG_bd=matrix(0,Ns,m) #Pèrdidas y ganancias
PGfr1_bd=matrix(0,Ns,m)
PGfr2_bd=matrix(0,Ns,m)
PGfr3_bd=matrix(0,Ns,m)
PGT_bd=matrix(0,Ns,1)
PGfr1T_bd=matrix(0,Ns,1)
PGfr2T_bd=matrix(0,Ns,1)
PGfr3T_bd=matrix(0,Ns,1)

DeltaX_s_bd=DeltaX_s[,sum(n_if[1:3],1):sum(n_if[1:4])]    #PASO CLAVE
x0_bd=X_bd_ext[1,]    #PASO CLAVE

for (i in 1:Ns)
{
  X_s_bd[i,]=x0_bd*(1+DeltaX_s_bd[i,])
  #PASO CLAVE
  V_bd[i,]=bondeD(contratos_bdmT, nominal_bdm, X_s_bd[i,1], plazocupon_bdmT, VTplazos_bdm, X_s_bd[i,(n_
  #PASO CLAVE
  Vfr1_bd[i,]=bondeD(contratos_bdmT, nominal_bdm, X_s_bd[i,1], plazocupon_bdmT, VTplazos_bdm, x0_bd[(n_
  #PASO CLAVE
  Vfr2_bd[i,]=bondeD(contratos_bdmT, nominal_bdm, x0_bd[1], plazocupon_bdmT, VTplazos_bdm, X_s_bd[i,(n_
  #PASO CLAVE
  Vfr3_bd[i,]=bondeD(contratos_bdmT, nominal_bdm, x0_bd[1], plazocupon_bdmT, VTplazos_bdm, x0_bd[(n_if[
```

```
  #PASO CLAVE
  PG_bd[i,]=V_bd[i,]-V0_bd
  PGfr1_bd[i,]=Vfr1_bd[i,]-V0_bd
  PGfr2_bd[i,]=Vfr2_bd[i,]-V0_bd
  PGfr3_bd[i,]=Vfr3_bd[i,]-V0_bd
  PGT_bd[i,]=sum(PG_bd[i,])
  PGfr1T_bd[i,]=sum(PGfr1_bd[i,])
  PGfr2T_bd[i,]=sum(PGfr2_bd[i,])
  PGfr3T_bd[i,]=sum(PGfr3_bd[i,])
}



PG_bd[1:5,]
```

```
## [1]  -467.0128  1610.2225  1442.8175 -1933.4229  -799.7940
```

```
PGfr1_bd[1:5,]
```

```
## [1] -690.74412  549.93825 -739.85749   35.94434  638.95048
```

```
PGfr2_bd[1:5,]
```

```
## [1]   166.6760   779.2571  1603.1026 -1446.7052 -1056.9950
```

```
PGT_bd[1:5,]
```

```
## [1]  -467.0128  1610.2225  1442.8175 -1933.4229  -799.7940
```

```
#VaR por posición
VaRCont_bd=matrix(0,1,m)
VaRfr1_bd=matrix(0,1,m)
VaRfr2_bd=matrix(0,1,m)
VaRfr3_bd=matrix(0,1,m)
CVaRCont_bd=matrix(0,1,m)
CVaRfr1_bd=matrix(0,1,m)
CVaRfr2_bd=matrix(0,1,m)
CVaRfr3_bd=matrix(0,1,m)
for (i in (1:m))
{
  VaRCont_bd[i]=quantile(PG_bd[,i],1-alpha,Ns)
  VaRfr1_bd[i]=quantile(PGfr1_bd[,i],1-alpha,Ns)
  VaRfr2_bd[i]=quantile(PGfr2_bd[,i],1-alpha,Ns)
  VaRfr3_bd[i]=quantile(PGfr3_bd[,i],1-alpha,Ns)
  CVaRfr1_bd[i]= mean(merge(which(PGfr1_bd[,i]<VaRfr1_bd[i]),cbind(seq(1,Ns),PGfr1_bd[,i]), by.x=1,by.y=
  CVaRfr2_bd[i]= mean(merge(which(PGfr2_bd[,i]<VaRfr2_bd[i]),cbind(seq(1,Ns),PGfr2_bd[,i]), by.x=1,by.y=
  CVaRfr3_bd[i]= mean(merge(which(PGfr3_bd[,i]<VaRfr3_bd[i]),cbind(seq(1,Ns),PGfr3_bd[,i]), by.x=1,by.y=
  CVaRCont_bd[i]= mean(merge(which(PG_bd[,i]<VaRCont_bd[i]),cbind(seq(1,Ns),PG_bd[,i]), by.x=1,by.y=1)[
}
Met
```

```
## [1] "Componentes Principales Empíricos"
```

```
VaRCont_bd
```

```
##           [,1]
## [1,] -3274.974
```

```
VaRfr1_bd
```

```
##           [,1]
## [1,] -2301.434
```

```
VaRfr2_bd
```

```
##          [,1]
## [1,] -2034.99
```

```
CVaRCont_bd
```

```
##           [,1]
## [1,] -3989.658
```

```
CVaRfr1_bd
```

```
##           [,1]
## [1,] -2671.711
```

```
CVaRfr2_bd
```

```
##           [,1]
## [1,] -2501.974
```

```
#VaR Total
#Met
VaRTotal_bd=quantile(PGT_bd,1-alpha,Ns)
CVaRTotal_bd= mean(merge(which(PGT_bd<VaRTotal_bd),cbind(seq(1,Ns),PGT_bd), by.x=1,by.y=1)[,2])
VaRTotalfr1_bd=quantile(PGfr1T_bd,1-alpha,Ns)
CVaRTotalfr1_bd= mean(PGfr1T_bd[which(PGfr1T_bd<VaRTotalfr1_bd),])
VaRTotalfr2_bd=quantile(PGfr2T_bd,1-alpha,Ns)
CVaRTotalfr2_bd= mean(PGfr2T_bd[which(PGfr2T_bd<VaRTotalfr2_bd),])
VaRTotalfr3_bd=quantile(PGfr3T_bd,1-alpha,Ns)
CVaRTotalfr3_bd= mean(PGfr3T_bd[which(PGfr2T_bd<VaRTotalfr2_bd),])


#cbind(VaRTotal_bd,sum(VO_bd), VaRCont_bd, VO_bd)
#cbind(CVaRTotal_bd,sum(VO_bd), CVaRCont_bd, VO_bd)
#cbind(VaRTotal_bd,VaRTotalfr1_bd,VaRTotalfr2_bd,VaRTotalfr3_bd)
#cbind(CVaRTotal_bd,CVaRTotalfr1_bd,CVaRTotalfr2_bd,CVaRTotalfr3_bd)
lista<-c('Tasa descuento','tasa fondeo', 'sobretasa')
```

```
print('Var total')
```

```
## [1] "Var total"
```

```
print(VaRTotal_bd)
```

```
##        2%
## -3274.974
```

```
print('Var por factor de riesgo')
```

```
## [1] "Var por factor de riesgo"
```

```
print(rbind(lista,t(c(VaRTotalfr1_bd,VaRTotalfr2_bd,VaRTotalfr3_bd))))
```

```
##       2%                  2%                  2%
## lista "Tasa descuento"    "tasa fondeo"       "sobretasa"
##       "-2301.43385025399" "-2034.98988954418" "-736.033040589429"
```

```
print('CVar total')
```

```
## [1] "CVar total"
```

```
print(CVaRTotal_bd)
```

```
## [1] -3989.658
```

```
print('CVar por factor de riesgo')
```

```
## [1] "CVar por factor de riesgo"
```

```
print(rbind(lista,t(c(CVaRTotalfr1_bd,CVaRTotalfr2_bd,CVaRTotalfr3_bd))))
```

```
##       [,1]                [,2]                [,3]
## lista "Tasa descuento"    "tasa fondeo"       "sobretasa"
##       "-2671.71100165683" "-2501.97409923006" "-905.245790334314"
```

Forwards de Divisas

```
#Cálculo de matriz de pérdidas y ganancias FUTUROS TDC
#dimensión
m=ncol(plazos_fwd)  #PASO CLAVE
X_s_fwtdc=matrix(0,Ns,n_if[5]) #Factores de riesgo simulados con base en DeltaX_s x0*(1+Delta_Xs) #PASO
V_fwtdc=matrix(0,Ns,m)
Vfr1_fwtdc=matrix(0,Ns,m)
Vfr2_fwtdc=matrix(0,Ns,m)
Vfr3_fwtdc=matrix(0,Ns,m)
PG_fwtdc=matrix(0,Ns,m) #Pèrdidas y ganancias
PGfr1_fwtdc=matrix(0,Ns,m)
PGfr2_fwtdc=matrix(0,Ns,m)
PGfr3_fwtdc=matrix(0,Ns,m)
PGT_fwtdc=matrix(0,Ns,1)
PGfr1T_fwtdc=matrix(0,Ns,1)
PGfr2T_fwtdc=matrix(0,Ns,1)
PGfr3T_fwtdc=matrix(0,Ns,1)

DeltaX_s_fwtdc=DeltaX_s[,sum(n_if[1:4],1):sum(n_if[1:5])]  #PASO CLAVE
x0_fwtdc=X_futtdc[1,] #PASO CLAVE

for (i in 1:Ns)
{
  X_s_fwtdc[i,]=x0_fwtdc*(1+DeltaX_s_fwtdc[i,])
  #PASO CLAVE
  V_fwtdc[i,]=futuroTC(plazos_fwd,X_s_fwtdc[i,1:((n_if[5]-1)/2)],X_s_fwtdc[i,((n_if[5]-1)/2+1):(n_if[5]
  #PASO CLAVE
  Vfr1_fwtdc[i,]=futuroTC(plazos_fwd,X_s_fwtdc[i,1:((n_if[5]-1)/2)],x0_fwtdc[((n_if[5]-1)/2+1):(n_if[5]
  #PASO CLAVE
  Vfr2_fwtdc[i,]=futuroTC(plazos_fwd,x0_fwtdc[1:((n_if[5]-1)/2)],X_s_fwtdc[i,((n_if[5]-1)/2+1):(n_if[5]
  #PASO CLAVE
  Vfr3_fwtdc[i,]=futuroTC(plazos_fwd,x0_fwtdc[1:((n_if[5]-1)/2)],x0_fwtdc[((n_if[5]-1)/2+1):(n_if[5]-1)]
  PG_fwtdc[i,]=V_fwtdc[i,]-V0_fwtdc
  PGfr1_fwtdc[i,]=Vfr1_fwtdc[i,]-V0_fwtdc
  PGfr2_fwtdc[i,]=Vfr2_fwtdc[i,]-V0_fwtdc
  PGfr3_fwtdc[i,]=Vfr3_fwtdc[i,]-V0_fwtdc
  PGT_fwtdc[i,]=sum(PG_fwtdc[i,])
```

```
  PGfr1T_fwtdc[i,]=sum(PGfr1_fwtdc[i,])
  PGfr2T_fwtdc[i,]=sum(PGfr2_fwtdc[i,])
  PGfr3T_fwtdc[i,]=sum(PGfr3_fwtdc[i,])
}



PG_fwtdc[1:5,]
```

```
## [1]  3.7038227 -4.0260594  0.1760134  3.0495742 -0.5241484
```

```
PGfr1_fwtdc[1:5,]
```

```
## [1]  0.0006558052 -0.0003607733 -0.0003262710 -0.0006497014  0.0057951130
```

```
PGfr2_fwtdc[1:5,]
```

```
## [1]  0.020388963 -0.001263253  0.005352639 -0.001292843  0.064094336
```

```
PGT_fwtdc[1:5,]
```

```
## [1]  3.7038227 -4.0260594  0.1760134  3.0495742 -0.5241484
```

```
#VaR por posición
VaRCont_fwtdc=matrix(0,1,m)
VaRfr1_fwtdc=matrix(0,1,m)
VaRfr2_fwtdc=matrix(0,1,m)
VaRfr3_fwtdc=matrix(0,1,m)
CVaRCont_fwtdc=matrix(0,1,m)
CVaRfr1_fwtdc=matrix(0,1,m)
CVaRfr2_fwtdc=matrix(0,1,m)
CVaRfr3_fwtdc=matrix(0,1,m)
for (i in (1:m))
{
  VaRCont_fwtdc[i]=quantile(PG_fwtdc[,i],1-alpha,Ns)
  VaRfr1_fwtdc[i]=quantile(PGfr1_fwtdc[,i],1-alpha,Ns)
  VaRfr2_fwtdc[i]=quantile(PGfr2_fwtdc[,i],1-alpha,Ns)
  VaRfr3_fwtdc[i]=quantile(PGfr3_fwtdc[,i],1-alpha,Ns)
  CVaRfr1_fwtdc[i]= mean(merge(which(PGfr1_fwtdc[,i]<VaRfr1_fwtdc[i]),cbind(seq(1,Ns),PGfr1_fwtdc[,i]),
  CVaRfr2_fwtdc[i]= mean(merge(which(PGfr2_fwtdc[,i]<VaRfr2_fwtdc[i]),cbind(seq(1,Ns),PGfr2_fwtdc[,i]),
  CVaRfr3_fwtdc[i]= mean(merge(which(PGfr3_fwtdc[,i]<VaRfr3_fwtdc[i]),cbind(seq(1,Ns),PGfr3_fwtdc[,i]),
  CVaRCont_fwtdc[i]= mean(merge(which(PG_fwtdc[,i]<VaRCont_fwtdc[i]),cbind(seq(1,Ns),PG_fwtdc[,i]), by.
}
#Met
#VaRCont_fwtdc
#VaRfr1_fwtdc
#VaRfr2_fwtdc
#CVaRCont_fwtdc
#CVaRfr1_fwtdc
#CVaRfr2_fwtdc


#VaR Total
Met
```

```
## [1] "Componentes Principales Empíricos"
```

```
VaRTotal_fwtdc=quantile(PGT_fwtdc,1-alpha,Ns)
CVaRTotal_fwtdc= mean(merge(which(PGT_fwtdc<VaRTotal_fwtdc),cbind(seq(1,Ns),PGT_fwtdc), by.x=1,by.y=1)[
VaRTotalfr1_fwtdc=quantile(PGfr1T_fwtdc,1-alpha,Ns)
CVaRTotalfr1_fwtdc= mean(PGfr1T_fwtdc[which(PGfr1T_fwtdc<VaRTotalfr1_fwtdc),])
VaRTotalfr2_fwtdc=quantile(PGfr2T_fwtdc,1-alpha,Ns)
CVaRTotalfr2_fwtdc= mean(PGfr2T_fwtdc[which(PGfr2T_fwtdc<VaRTotalfr2_fwtdc),])
VaRTotalfr3_fwtdc=quantile(PGfr3T_fwtdc,1-alpha,Ns)
CVaRTotalfr3_fwtdc= mean(PGfr3T_fwtdc[which(PGfr2T_fwtdc<VaRTotalfr2_fwtdc),])


#cbind(VaRTotal_fwtdc,sum(VO_fwtdc), VaRCont_fwtdc, VO_fwtdc)
#cbind(CVaRTotal_fwtdc,sum(VO_fwtdc), CVaRCont_fwtdc, VO_fwtdc)
cbind(VaRTotal_fwtdc,VaRTotalfr1_fwtdc,VaRTotalfr2_fwtdc,VaRTotalfr3_fwtdc)
```

```
##    VaRTotal_fwtdc VaRTotalfr1_fwtdc VaRTotalfr2_fwtdc VaRTotalfr3_fwtdc
## 2%      -5.133042      -0.002406626       -0.02290429         -5.139011
```

```
cbind(CVaRTotal_fwtdc,CVaRTotalfr1_fwtdc,CVaRTotalfr2_fwtdc,CVaRTotalfr3_fwtdc)
```

```
##      CVaRTotal_fwtdc CVaRTotalfr1_fwtdc CVaRTotalfr2_fwtdc CVaRTotalfr3_fwtdc
## [1,]         -5.93205       -0.002901309        -0.02792417         -0.2898978
```

Forwards IPC

```
#Cálculo de matriz de pérdidas y ganancias FUTUROS IPC
m=ncol(plazos_fwd_ind)  #PASO CLAVE
X_s_fwind=matrix(0,Ns,n_if[6]) #Factores de riesgo simulados con base en DeltaX_s x0*(1+Delta_Xs) #PASO
V_fwind=matrix(0,Ns,m)
Vfr1_fwind=matrix(0,Ns,m)
Vfr2_fwind=matrix(0,Ns,m)
Vfr3_fwind=matrix(0,Ns,m)
PG_fwind=matrix(0,Ns,m) #Pèrdidas y ganancias
PGfr1_fwind=matrix(0,Ns,m)
PGfr2_fwind=matrix(0,Ns,m)
PGfr3_fwind=matrix(0,Ns,m)
PGT_fwind=matrix(0,Ns,1)
PGfr1T_fwind=matrix(0,Ns,1)
PGfr2T_fwind=matrix(0,Ns,1)
PGfr3T_fwind=matrix(0,Ns,1)

DeltaX_s_fwind=DeltaX_s[,sum(n_if[1:5],1):sum(n_if[1:6])]  #PASO CLAVE
x0_fwind=X_futind[1,] #PASO CLAVE


for (i in 1:Ns)
{
  X_s_fwind[i,]=x0_fwind*(1+DeltaX_s_fwind[i,])
  #PASO CLAVE
  V_fwind[i,]=futuroTC(plazos_fwd_ind,X_s_fwind[i,1:(n_if[6]/3)],X_s_fwind[i,(n_if[6]/3+1):(n_if[6]*2/3)
  Vfr1_fwind[i,]=futuroTC(plazos_fwd_ind,X_s_fwind[i,1:(n_if[6]/3)],x0_fwind[(n_if[6]/3+1):(n_if[6]*2/3)
  #PASO CLAVE
  Vfr2_fwind[i,]=futuroTC(plazos_fwd_ind,x0_fwind[1:(n_if[6]/3)],X_s_fwind[i,(n_if[6]/3+1):(n_if[6]*2/3)
  #PASO CLAVE
  Vfr3_fwind[i,]=futuroTC(plazos_fwd_ind,x0_fwind[1:(n_if[6]/3)],x0_fwind[(n_if[6]/3+1):(n_if[6]*2/3)],x
```

```
  PG_fwind[i,]=V_fwind[i,]-V0_fwind
  PGfr1_fwind[i,]=Vfr1_fwind[i,]-V0_fwind
  PGfr2_fwind[i,]=Vfr2_fwind[i,]-V0_fwind
  PGfr3_fwind[i,]=Vfr3_fwind[i,]-V0_fwind
  PGT_fwind[i,]=sum(PG_fwind[i,])
  PGfr1T_fwind[i,]=sum(PGfr1_fwind[i,])
  PGfr2T_fwind[i,]=sum(PGfr2_fwind[i,])
  PGfr3T_fwind[i,]=sum(PGfr3_fwind[i,])
}


PG_fwind[1:5,]
```

```
## [1] -3030.8307 -3937.1041 -1936.7674 -2414.5466  -790.1794
```

```
#PGfr1_fwind[1:5,]
#PGfr2_fwind[1:5,]
#PGT_fwind[1:5,]



#VaR por posición
VaRCont_fwind=matrix(0,1,m)
VaRfr1_fwind=matrix(0,1,m)
VaRfr2_fwind=matrix(0,1,m)
VaRfr3_fwind=matrix(0,1,m)
CVaRCont_fwind=matrix(0,1,m)
CVaRfr1_fwind=matrix(0,1,m)
CVaRfr2_fwind=matrix(0,1,m)
CVaRfr3_fwind=matrix(0,1,m)
for (i in (1:m))
{
  VaRCont_fwind[i]=quantile(PG_fwind[,i],1-alpha,Ns)
  VaRfr1_fwind[i]=quantile(PGfr1_fwind[,i],1-alpha,Ns)
  VaRfr2_fwind[i]=quantile(PGfr2_fwind[,i],1-alpha,Ns)
  VaRfr3_fwind[i]=quantile(PGfr3_fwind[,i],1-alpha,Ns)
  CVaRfr1_fwind[i]= mean(merge(which(PGfr1_fwind[,i]<VaRfr1_fwind[i]),cbind(seq(1,Ns),PGfr1_fwind[,i]),
  CVaRfr2_fwind[i]= mean(merge(which(PGfr2_fwind[,i]<VaRfr2_fwind[i]),cbind(seq(1,Ns),PGfr2_fwind[,i]),
  CVaRfr3_fwind[i]= mean(merge(which(PGfr3_fwind[,i]<VaRfr3_fwind[i]),cbind(seq(1,Ns),PGfr3_fwind[,i]),
  CVaRCont_fwind[i]= mean(merge(which(PG_fwind[,i]<VaRCont_fwind[i]),cbind(seq(1,Ns),PG_fwind[,i]), by.
}
#Met
#VaRCont_fwind
#VaRfr1_fwind
#VaRfr2_fwind
#CVaRCont_fwind
#CVaRfr1_fwind
#CVaRfr2_fwind


#VaR Total
Met
```

```
## [1] "Componentes Principales Empíricos"
```

```
VaRTotal_fwind=quantile(PGT_fwind,1-alpha,Ns)
CVaRTotal_fwind= mean(merge(which(PGT_fwind<VaRTotal_fwind),cbind(seq(1,Ns),PGT_fwind), by.x=1,by.y=1)[
VaRTotalfr1_fwind=quantile(PGfr1T_fwind,1-alpha,Ns)
CVaRTotalfr1_fwind= mean(PGfr1T_fwind[which(PGfr1T_fwind<VaRTotalfr1_fwind),])
VaRTotalfr2_fwind=quantile(PGfr2T_fwind,1-alpha,Ns)
CVaRTotalfr2_fwind= mean(PGfr2T_fwind[which(PGfr2T_fwind<VaRTotalfr2_fwind),])
VaRTotalfr3_fwind=quantile(PGfr3T_fwind,1-alpha,Ns)
CVaRTotalfr3_fwind= mean(PGfr3T_fwind[which(PGfr3T_fwind<VaRTotalfr3_fwind),])


#print(cbind(VaRTotal_fwind,sum(VO_fwind), VaRCont_fwind, VO_fwind))
#print(cbind(CVaRTotal_fwind,sum(VO_fwind), CVaRCont_fwind, VO_fwind))
print(cbind(VaRTotal_fwind,VaRTotalfr1_fwind,VaRTotalfr2_fwind,VaRTotalfr3_fwind))
```

```
##    VaRTotal_fwind VaRTotalfr1_fwind VaRTotalfr2_fwind VaRTotalfr3_fwind
## 2%      -7944.148                 0          -477.952         -7851.706
```

```
cbind(CVaRTotal_fwind,CVaRTotalfr1_fwind,CVaRTotalfr2_fwind,CVaRTotalfr3_fwind)
```

```
##      CVaRTotal_fwind CVaRTotalfr1_fwind CVaRTotalfr2_fwind CVaRTotalfr3_fwind
## [1,]       -9415.303                NaN          -565.0077          -9376.661
```

Swaps

```
#Cálculo de matriz de pérdidas y ganancias SWAP

#riesgo del swap
m=ncol(N)    #PASO CLAVE
X_s_sw=matrix(0,Ns,n_if[7]) #Factores de riesgo simulados con base en DeltaX_s x0*(1+Delta_Xs)   #PASO
V_sw=matrix(0,Ns,m)
Vfr1_sw=matrix(0,Ns,m)
Vfr2_sw=matrix(0,Ns,m)
PG_sw=matrix(0,Ns,m) #Pèrdidas y ganancias
PGfr1_sw=matrix(0,Ns,m)
PGfr2_sw=matrix(0,Ns,m)
PGT_sw=matrix(0,Ns,1)
PGfr1T_sw=matrix(0,Ns,1)
PGfr2T_sw=matrix(0,Ns,1)

DeltaX_s_sw=DeltaX_s[,sum(n_if[1:6],1):sum(n_if[1:7])]    #PASO CLAVE
x0_sw=as.numeric(c(XtfwdT[1,],Xvp[1,]))   #PASO CLAVE

for (i in 1:Ns)
{
  X_s_sw[i,]=x0_sw*(1+DeltaX_s_sw[i,])
  #PASO CLAVE
  V_sw[i,]=swap(por_swT, contratos_swT, nominal_swT, X_s_sw[i,1:(n_if[7]/2)], tasafija_swT, plazocupon_s
  #PASO CLAVE
  Vfr1_sw[i,]=swap(por_swT, contratos_swT, nominal_swT,X_s_sw[i,1:(n_if[7]/2)], tasafija_swT, plazocupo
  #PASO CLAVE
  Vfr2_sw[i,]=swap(por_swT, contratos_swT, nominal_swT,  XtfwdT[1,], tasafija_swT, plazocupon_swT, VTpla
  #PASO CLAVE
  PG_sw[i,]=V_sw[i,]-V0_sw
  PGfr1_sw[i,]=Vfr1_sw[i,]-V0_sw
```

```
  PGfr2_sw[i,]=Vfr2_sw[i,]-V0_sw
  PGT_sw[i,]=sum(PG_sw[i,])
  PGfr1T_sw[i,]=sum(PGfr1_sw[i,])
  PGfr2T_sw[i,]=sum(PGfr2_sw[i,])
}



PG_sw[1:5,]
```

```
##            [,1]        [,2]
## [1,]   5083.713  -7273.8958
## [2,]  -1548.918   -614.9932
## [3,]   3300.219   -635.2056
## [4,] -21595.053   3766.1405
## [5,]  60977.503 -21800.3118
```

```
PGfr1_sw[1:5,]
```

```
##            [,1]        [,2]
## [1,]   5035.715  -7299.6672
## [2,]  -1537.000   -611.1445
## [3,]   3291.252   -640.4820
## [4,] -21629.078   3760.3514
## [5,]  60914.444 -21916.8945
```

```
PGfr2_sw[1:5,]
```

```
##            [,1]      [,2]
## [1,]  44.783624 23.209490
## [2,] -11.369350 -3.814326
## [3,]   9.494095  5.245754
## [4,]  25.684823  6.105394
## [5,] 214.851611 90.306555
```

```
PGT_sw[1:5,]
```

```
## [1]  -2190.183  -2163.911   2665.014 -17828.913  39177.191
```

```
#VaR por posición
VaRCont_sw=matrix(0,1,m)
VaRfr1_sw=matrix(0,1,m)
VaRfr2_sw=matrix(0,1,m)
CVaRCont_sw=matrix(0,1,m)
CVaRfr1_sw=matrix(0,1,m)
CVaRfr2_sw=matrix(0,1,m)
for (i in (1:m))
{
  VaRCont_sw[i]=quantile(PG_sw[,i],1-alpha,Ns)
  VaRfr1_sw[i]=quantile(PGfr1_sw[,i],1-alpha,Ns)
  VaRfr2_sw[i]=quantile(PGfr2_sw[,i],1-alpha,Ns)
  CVaRfr1_sw[i]= mean(merge(which(PGfr1_sw[,i]<VaRfr1_sw[i]),cbind(seq(1,Ns),PGfr1_sw[,i]), by.x=1,by.y=
  CVaRfr2_sw[i]= mean(merge(which(PGfr2_sw[,i]<VaRfr2_sw[i]),cbind(seq(1,Ns),PGfr2_sw[,i]), by.x=1,by.y=
  CVaRCont_sw[i]= mean(merge(which(PG_sw[,i]<VaRCont_sw[i]),cbind(seq(1,Ns),PG_sw[,i]), by.x=1,by.y=1)[
}
#Met
```

```
#VaRCont_sw
#VaRfr1_sw
#VaRfr2_sw
#CVaRCont_sw
#CVaRfr1_sw
#CVaRfr2_sw


#VaR Total
Met
```

```
## [1] "Componentes Principales Empíricos"
```

```
VaRTotal_sw=quantile(PGT_sw,1-alpha,Ns)
CVaRTotal_sw= mean(merge(which(PGT_sw<VaRTotal_sw),cbind(seq(1,Ns),PGT_sw), by.x=1,by.y=1)[,2])
VaRTotalfr1_sw=quantile(PGfr1T_sw,1-alpha,Ns)
CVaRTotalfr1_sw= mean(PGfr1T_sw[which(PGfr1T_sw<VaRTotalfr1_sw),])
VaRTotalfr2_sw=quantile(PGfr2T_sw,1-alpha,Ns)
CVaRTotalfr2_sw= mean(PGfr2T_sw[which(PGfr2T_sw<VaRTotalfr2_sw),])


cbind(VaRTotal_sw, VaRCont_sw)
```

```
##    VaRTotal_sw
## 2%    -18721.9 -27080.27 -26071.14
```

```
cbind(CVaRTotal_sw, CVaRCont_sw)
```

```
##      CVaRTotal_sw
## [1,]     -22473.3 -32688.43 -35661.19
```

```
cbind(VaRTotal_sw,VaRTotalfr1_sw,VaRTotalfr2_sw)
```

```
##    VaRTotal_sw VaRTotalfr1_sw VaRTotalfr2_sw
## 2%    -18721.9      -18610.19      -107.2127
```

```
cbind(CVaRTotal_sw,CVaRTotalfr1_sw,CVaRTotalfr2_sw)
```

```
##      CVaRTotal_sw CVaRTotalfr1_sw CVaRTotalfr2_sw
## [1,]     -22473.3       -22382.57       -133.9819
```

Opciones de tasa de interés

```
#Cálculo de matriz de pérdidas y ganancias Opciones Tasa de interés
#dimensión
m=ncol(plazos_oir)  #PASO CLAVE
X_s_oir=matrix(0,Ns,n_if[8]) #Factores de riesgo simulados con base en DeltaX_s x0*(1+Delta_Xs) #PASO C
V_oir=matrix(0,Ns,m)
Vfr1_oir=matrix(0,Ns,m)
Vfr2_oir=matrix(0,Ns,m)
Vfr3_oir=matrix(0,Ns,m)
PG_oir=matrix(0,Ns,m) #Pèrdidas y ganancias
PGfr1_oir=matrix(0,Ns,m)
PGfr2_oir=matrix(0,Ns,m)
PGfr3_oir=matrix(0,Ns,m)
PGT_oir=matrix(0,Ns,1)
PGfr1T_oir=matrix(0,Ns,1)
```

```
PGfr2T_oir=matrix(0,Ns,1)
PGfr3T_oir=matrix(0,Ns,1)

DeltaX_s_oir=DeltaX_s[,sum(n_if[1:7],1):sum(n_if[1:8])]  #PASO CLAVE
x0_oir=X_oir[1,] #PASO CLAVE

for (i in 1:Ns)
{
  X_s_oir[i,]=x0_oir*(1+DeltaX_s_oir[i,])
  #PASO CLAVE
  V_oir[i,]=  opctint(X_s_oir[i,(1:(n_if[8]/3))],X_s_oir[i,((n_if[8]/3+1):(n_if[8]/3*2))],K_oir,X_s_oir
  #PASO CLAVE
  Vfr1_oir[i,]=opctint(X_s_oir[i,(1:(n_if[8]/3))],x0_oir[((n_if[8]/3+1):(n_if[8]/3*2))],K_oir,x0_oir[((
  #PASO CLAVE
  Vfr2_oir[i,]=opctint(x0_oir[(1:(n_if[8]/3))],X_s_oir[i,((n_if[8]/3+1):(n_if[8]/3*2))],K_oir,x0_oir[((
  #PASO CLAVE
  Vfr3_oir[i,]=opctint(x0_oir[(1:(n_if[8]/3))],x0_oir[((n_if[8]/3+1):(n_if[8]/3*2))],K_oir,X_s_oir[i,((
  PG_oir[i,]=V_oir[i,]-V0_oir
  PGfr1_oir[i,]=Vfr1_oir[i,]-V0_oir
  PGfr2_oir[i,]=Vfr2_oir[i,]-V0_oir
  PGfr3_oir[i,]=Vfr3_oir[i,]-V0_oir
  PGT_oir[i,]=sum(PG_oir[i,])
  PGfr1T_oir[i,]=sum(PGfr1_oir[i,])
  PGfr2T_oir[i,]=sum(PGfr2_oir[i,])
  PGfr3T_oir[i,]=sum(PGfr3_oir[i,])
}



PG_oir[1:5,]
```

```
##              [,1]         [,2]
## [1,] -0.001323403 -0.036118406
## [2,]  0.024660440  0.017035134
## [3,]  0.009638583  0.003852365
## [4,]  0.028063491 -0.032725888
## [5,] -0.037137023  0.034572858
```

```
PGfr1_oir[1:5,]
```

```
##               [,1]          [,2]
## [1,] -0.0021853422 -2.730064e-04
## [2,]  0.0009006386  7.848389e-05
## [3,] -0.0002695865 -5.086286e-05
## [4,] -0.0020219122 -1.864920e-04
## [5,] -0.0121723255 -1.369369e-03
```

```
PGfr2_oir[1:5,]
```

```
##              [,1]         [,2]
## [1,]  0.016543727 -0.030358176
## [2,] -0.001334409 -0.006330685
## [3,]  0.005371467  0.005700979
## [4,]  0.038754824 -0.031580455
## [5,] -0.021651443  0.036150252
```

```r
PGT_oir[1:5,]
```

```
## [1] -0.037441810  0.041695574  0.013490948 -0.004662397 -0.002564165
```

```r
#VaR por posición
VaRCont_oir=matrix(0,1,m)
VaRfr1_oir=matrix(0,1,m)
VaRfr2_oir=matrix(0,1,m)
VaRfr3_oir=matrix(0,1,m)
CVaRCont_oir=matrix(0,1,m)
CVaRfr1_oir=matrix(0,1,m)
CVaRfr2_oir=matrix(0,1,m)
CVaRfr3_oir=matrix(0,1,m)
for (i in (1:m))
{
  VaRCont_oir[i]=quantile(PG_oir[,i],1-alpha,Ns)
  VaRfr1_oir[i]=quantile(PGfr1_oir[,i],1-alpha,Ns)
  VaRfr2_oir[i]=quantile(PGfr2_oir[,i],1-alpha,Ns)
  VaRfr3_oir[i]=quantile(PGfr3_oir[,i],1-alpha,Ns)
  CVaRfr1_oir[i]= mean(merge(which(PGfr1_oir[,i]<VaRfr1_oir[i]),cbind(seq(1,Ns),PGfr1_oir[,i]), by.x=1,
  CVaRfr2_oir[i]= mean(merge(which(PGfr2_oir[,i]<VaRfr2_oir[i]),cbind(seq(1,Ns),PGfr2_oir[,i]), by.x=1,
  CVaRfr3_oir[i]= mean(merge(which(PGfr3_oir[,i]<VaRfr3_oir[i]),cbind(seq(1,Ns),PGfr3_oir[,i]), by.x=1,
  CVaRCont_oir[i]= mean(merge(which(PG_oir[,i]<VaRCont_oir[i]),cbind(seq(1,Ns),PG_oir[,i]), by.x=1,by.y
}
#Met
#VaRCont_oir
#VaRfr1_oir
#VaRfr2_oir
#CVaRCont_oir
#CVaRfr1_oir
#CVaRfr2_oir


#VaR Total
#Met
VaRTotal_oir=quantile(PGT_oir,1-alpha,Ns)
CVaRTotal_oir= mean(merge(which(PGT_oir<VaRTotal_oir),cbind(seq(1,Ns),PGT_oir), by.x=1,by.y=1)[,2])
VaRTotalfr1_oir=quantile(PGfr1T_oir,1-alpha,Ns)
CVaRTotalfr1_oir= mean(PGfr1T_oir[which(PGfr1T_oir<VaRTotalfr1_oir),])
VaRTotalfr2_oir=quantile(PGfr2T_oir,1-alpha,Ns)
CVaRTotalfr2_oir= mean(PGfr2T_oir[which(PGfr2T_oir<VaRTotalfr2_oir),])
VaRTotalfr3_oir=quantile(PGfr3T_oir,1-alpha,Ns)
CVaRTotalfr3_oir= mean(PGfr3T_oir[which(PGfr2T_oir<VaRTotalfr2_oir),])


cbind(VaRTotal_oir, VaRCont_oir)
```

```
##    VaRTotal_oir
## 2%  -0.06360256 -0.06611972 -0.03522779
```

```r
cbind(CVaRTotal_oir, CVaRCont_oir)
```

```
##      CVaRTotal_oir
## [1,]    -0.1097517 -0.09394049 -0.05280802
```

```
cbind(VaRTotal_oir,VaRTotalfr1_oir,VaRTotalfr2_oir,VaRTotalfr3_oir)
```

```
##    VaRTotal_oir VaRTotalfr1_oir VaRTotalfr2_oir VaRTotalfr3_oir
## 2%  -0.06360256      -0.0110442     -0.02641088     -0.06273491
```

```
cbind(CVaRTotal_oir,CVaRTotalfr1_oir,CVaRTotalfr2_oir,CVaRTotalfr3_oir)
```

```
##      CVaRTotal_oir CVaRTotalfr1_oir CVaRTotalfr2_oir CVaRTotalfr3_oir
## [1,]    -0.1097517      -0.01605984      -0.03126089       0.01736628
```

Riesgo total del portafolio

```
#Medición de riesgo por factor de riesgo de todo el portafolios
#Acciones
#1. Acciones
#2. Forward de IPC
PGPort_ACC=PGfr2T_acc_div+ PGfr3T_fwind #Pérdidas y ganancias
VaRPort_ACC=quantile(PGPort_ACC,1-alpha,Ns) #VaR
CVaRPort_ACC= mean(PGPort_ACC[which(PGPort_ACC<VaRPort_ACC)]) #CVaR

print(VaRPort_ACC)
```

```
##        2%
## -8354.182
```

```
print(CVaRPort_ACC)
```

```
## [1] -9878.891
```

```
#Tasa de Interés
#1. Dado que swaps y bondes son de tasa de interés usaremos PGT_bd y PGT_sw
#2. Para futuros usaremos PGfr1T_fwtdc y PGfr2T_fwtdc
PGPort_TI=PGT_bd+PGT_sw+PGfr1T_fwtdc+PGfr2T_fwtdc +PGfr2T_fwind+PGfr1T_fwind +PGfr1T_oir +PGfr2T_oir #P
VaRPort_TI=quantile(PGPort_TI,1-alpha,Ns) #VaR
CVaRPort_TI= mean(PGPort_TI[which(PGPort_TI<VaRPort_TI)]) #CVaR
print(VaRPort_TI)
```

```
##        2%
## -18824.61
```

```
print(CVaRPort_TI)
```

```
## [1] -23002.8
```

```
#Tipo de cambio
#1. Dado que swaps y bondes son de tasa de interés no usamos nada
#2. Para futuros usamos sólo PGfr3T_fwtdc
PGPort_TDC=PGfr1T_acc_div+PGfr3T_fwtdc  #Pérdidas y ganancias
VaRPort_TDC=quantile(PGPort_TDC,1-alpha,Ns) #VaR
CVaRPort_TDC= mean(PGPort_TDC[which(PGPort_TDC<VaRPort_TDC)]) #CVaR
print(VaRPort_TDC)
```

```
##        2%
## -90.12542
```

```
print(CVaRPort_TDC)
```

```
## [1] -104.0559
```

```
#Volatilidad
#1. Sólo aplica la volatilidad de Opciones de tasa de interés

PGPort_VOL=PGfr3T_oir  #Pérdidas y ganancias
VaRPort_VOL=quantile(PGPort_VOL,1-alpha,Ns) #VaR
CVaRPort_VOL= mean(PGPort_VOL[which(PGPort_VOL<VaRPort_VOL)]) #CVaR

print(VaRPort_VOL)
```

```
##          2%
## -0.06273491
```

```
print(CVaRPort_VOL)
```

```
## [1] -0.1169776
```

```
#Medición de riesgo de todo el portafolios
#Sumar todos los PGT de todos los instrumentos

PGT_Port=PGPort_ACC+PGPort_TI+PGPort_TDC+PGPort_VOL
VaRTotal_Port=quantile(PGT_Port,1-alpha,Ns) #VaR
CVaRTotal_Port= mean(PGT_Port[which(PGT_Port<VaRTotal_Port)]) #CVaR
print(VaRTotal_Port)
```

```
##        2%
## -22437.22
```

```
print(CVaRTotal_Port)
```

```
## [1] -27180.21
```

```
print(V0T_port)
```

```
## [1] 1474917
```

Por componentes principales normal

```
#DIMENSION DE TODOS LOS INSTRUMENTOS
#Son 8 instrumentos financieros (9 si separamos acciones y divisas)
n_if=matrix(0,8,1)
n_if[1]=ncol(stock_prices_EQFX) #acciones y divisas
n_if[2]=ncol(x_bcc) #cetes
n_if[3]=ncol(X_bm) #bonos m
n_if[4]=ncol(X_bd_ext) #bonde
n_if[5]=ncol(X_futtdc) #fut tdc
n_if[6]=ncol(X_futind) #fut ind
n_if[7]=ncol(X_sw) #swaps
n_if[8]=ncol(X_oir) #opciones tasa de interés

#valor del portafolios

V0_port=cbind(V0_acc_div,V0_bcc, V0_bm, V0_bd, V0_fwtdc, V0_fwind, V0_sw, V0_oir) #contrato
V0T_port=sum(V0_port)



#INTEGRACIÓN DE TODOS LOS FACTORES DE RIESGO EN UNA MATRIZ
```

```
X_port=cbind(stock_prices_EQFX,x_bcc,X_bm,X_bd_ext,X_futtdc,X_futind,X_sw,X_oir) #Factores de riesgo de

#Cálculo de variaciones Delta_X DEL PORTAFOLIOS
DeltaX_port=as.matrix(X_port[1:(n-1),]/X_port[2:(n),]-1)
DeltaX_port[is.nan(DeltaX_port)] <- 0 #quitamos NaN
DeltaX_port[is.na(DeltaX_port)] <- 0 #quitamos Na
DeltaX_port[is.infinite(DeltaX_port)] <- 0 #quitamos Na


Ns=10000 #Definimos número de simulaciones
alpha=0.98 #Nivel de Confianza para las medidas de riesgo
ChCP=1  #0 Cholesky, 1 Componentes Principales
normempi=0 #0 normal, 1 empírico
eta=.85 #Varianza explicada  (sólo aplica para Componentes Principales)
mT=ncol(DeltaX_port)


VarDeltaX=if(ChCP==0){cor(DeltaX_port)}else{var(DeltaX_port)*(ncol(DeltaX_port)/(ncol(DeltaX_port)-1))}

if(ChCP==0) {
  CVarDeltaX=as.matrix(chol(VarDeltaX,pivot=TRUE))
  CVarDeltaX[is.nan(CVarDeltaX)] <- 0 #quitamos NaN
  CVarDeltaX[is.na(CVarDeltaX)] <- 0 #quitamos Na
  print(ncol(CVarDeltaX))
  print(ncol(VarDeltaX))
  M=matrix(0, Ns,mT)
  if(normempi==1)
  {
    for (i in 1:mT)
    {
      M[,i]=quantile(DeltaX_port[,i],runif(Ns))
    }
    Met='Cholesky Empírico'
  } else    {
    for (i in 1:mT)
    {
      M[,i]=rnorm(Ns)*sd(DeltaX_port[,i])*(mT/(mT-1))+mean(DeltaX_port[,i])
    }
    Met='Cholesky Normal'
  }
  DeltaX_s=M%*%CVarDeltaX  #Factores de riesgo simulados
} else { #componentes principales
  G<-eigen(VarDeltaX)      #eigenvectores
  g1<-G$values                  #eigenvalores
  g1p=g1/sum(g1)

  g1pac=cumsum(g1p) #suma acumulada de varianza

  k=which.max(g1pac>eta) #valor mínimo k

  y=as.matrix((DeltaX_port-colMeans(DeltaX_port)))%*%G$vectors[,1:k] #componentes principales
  #/sqrt(diag(var(DeltaX))))
  M=matrix(0, Ns,k)
```

```
  if(normempi==0)
  {
    for (i in 1:k)
    {
      M[,i]=rnorm(Ns)*sd(y[,i])+mean(y[,i])
    }
    Met='Componentes Principales Normales'
  } else    {
    for (i in 1:k)
    {
      M[,i]=quantile(y[,i],runif(Ns))
    }
    Met='Componentes Principales Empíricos'
  }
  DeltaX_s=M%*% t(G$vectors[,1:k]) #Factores de riesgo simulados
}
#Met
#DeltaX_port[1:5,]


#Met


Ms1='Se explica el'
Ms2= 'de varianza'
Ms3= 'de suma de var-cov'

Ms1
```

```
## [1] "Se explica el"
```

```
print(sum(diag(var(DeltaX_s)))/sum(diag(var(DeltaX_port)))) #Comprobación de matriz de varianza-cov vs
```

```
## [1] 0.8549171
```

```
Ms2
```

```
## [1] "de varianza"
```

```
Ms1
```

```
## [1] "Se explica el"
```

```
print(matrix(1,1,ncol(DeltaX_s))%*%var(DeltaX_s)%*%t(matrix(1,1,ncol(DeltaX_s)))/matrix(1,1,ncol(DeltaX
```

```
##           [,1]
## [1,] 0.9797433
```

```
Ms3
```

```
## [1] "de suma de var-cov"
```

```
print(mT)
```

```
## [1] 173
```

```
#print(k) #número de variables simuladas
#print(g1pac)
```

Medición del riesgo

```
#Medición de riesgo por instrumento, instrumento-factor de riesgo, instrumento - total


#Cálculo de matriz de pérdidas y ganancias Acciones y Divisas

#riesgo del acciones y divisas
m=n_if[1]   #PASO CLAVE
X_s_acc_div=matrix(0,Ns,n_if[1]) #Factores de riesgo simulados con base en DeltaX_s x0*(1+Delta_Xs)   #
V_acc_div=matrix(0,Ns,m) #valor simulado a nivel contrato de acciones y tdc
Vfr1_acc_div=matrix(0,Ns,m_fx) #valor simulado a nivel contrato de tdc
Vfr2_acc_div=matrix(0,Ns,m_acc) #valor simulado a nivel contrato de acciones
PG_acc_div=matrix(0,Ns,m) #Pèrdidas y ganancias
PGfr1_acc_div=matrix(0,Ns,m_fx)
PGfr2_acc_div=matrix(0,Ns,m_acc)
PGT_acc_div=matrix(0,Ns,1) #TOTAL ACC Y DIV
PGfr1T_acc_div=matrix(0,Ns,1) #total factor riesgo divisas
PGfr2T_acc_div=matrix(0,Ns,1) #total factor riesgo acciones

DeltaX_s_acc_div=DeltaX_s[,(1:n_if[1])]    #PASO CLAVE
x0_acc_div=stock_prices_EQFX[1,]   #PASO CLAVE

for (i in 1:Ns)
{
  X_s_acc_div[i,]=as.matrix(x0_acc_div*(1+DeltaX_s_acc_div[i,]))
  #PASO CLAVE
  V_acc_div[i,]=cbind(t(pos_fx),t(pos_eq))*X_s_acc_div[i,]
  #PASO CLAVE
  Vfr1_acc_div[i,]=t(pos_fx)*X_s_acc_div[i,1:m_fx]
  #PASO CLAVE
  Vfr2_acc_div[i,]=t(pos_eq)*X_s_acc_div[i,(m_fx+1):(m_fx+m_acc)]
  #PASO CLAVE
  PG_acc_div[i,]=as.matrix(V_acc_div[i,]-V0_acc_div)
  PGfr1_acc_div[i,]=as.matrix(Vfr1_acc_div[i,]-V0_acc_div[,1:m_fx])
  PGfr2_acc_div[i,]=as.matrix(Vfr2_acc_div[i,]-V0_acc_div[,(m_fx+1):(m_fx+m_acc)])
  PGT_acc_div[i,]=sum(PG_acc_div[i,])
  PGfr1T_acc_div[i,]=sum(PGfr1_acc_div[i,])
  PGfr2T_acc_div[i,]=sum(PGfr2_acc_div[i,])
}

print(V_acc_div[1:3,])
```

Acciones y divisas

```
##          [,1]      [,2]     [,3]       [,4]     [,5]     [,6]
## [1,] 16113.10 -16425.25 31332.33 -100070.53 63793.18 91217.46
## [2,] 16122.73 -16440.91 31389.47  -99451.38 63021.63 91118.70
## [3,] 16115.68 -16427.78 31353.16  -99690.57 63946.34 90992.03
```

```
print(Vfr1_acc_div[1:3,])
```

```
##          [,1]      [,2]     [,3]
## [1,] 16113.10 -16425.25 31332.33
## [2,] 16122.73 -16440.91 31389.47
```

```
## [3,] 16115.68 -16427.78 31353.16
```

```
print(Vfr2_acc_div[1:3,])
```

```
##            [,1]     [,2]     [,3]
## [1,] -100070.53 63793.18 91217.46
## [2,]  -99451.38 63021.63 91118.70
## [3,]  -99690.57 63946.34 90992.03
```

```
PG_acc_div[1:5,]
```

```
##              [,1]        [,2]       [,3]       [,4]       [,5]       [,6]
## [1,]  -9.54915685 15.64333039 -35.673208 -470.53074 793.179571  -78.54415
## [2,]   0.07921706 -0.01192294  21.470294  148.62182  21.631080 -177.30628
## [3,]  -6.96303716 13.11443779 -14.838565  -90.56605 946.340856 -303.97186
## [4,]   1.16684427 -1.49447815  -9.605674   71.49211 -78.292705  -74.29751
## [5,] -19.22431715 14.89647079 -30.666589 -109.17448   5.776331  192.97952
```

```
PGfr1_acc_div[1:5,]
```

```
##              [,1]        [,2]       [,3]
## [1,]  -9.54915685 15.64333039 -35.673208
## [2,]   0.07921706 -0.01192294  21.470294
## [3,]  -6.96303716 13.11443779 -14.838565
## [4,]   1.16684427 -1.49447815  -9.605674
## [5,] -19.22431715 14.89647079 -30.666589
```

```
PGfr2_acc_div[1:5,]
```

```
##            [,1]       [,2]       [,3]
## [1,] -470.53074 793.179571  -78.54415
## [2,]  148.62182  21.631080 -177.30628
## [3,]  -90.56605 946.340856 -303.97186
## [4,]   71.49211 -78.292705  -74.29751
## [5,] -109.17448   5.776331  192.97952
```

```
PGT_acc_div[1:5,]
```

```
## [1] 214.52565  14.48422 543.11578 -91.03141  54.58694
```

```
#VaR por posición
VaRCont_acc_div=matrix(0,1,m)
VaRfr1_acc_div=matrix(0,1,m_fx)
VaRfr2_acc_div=matrix(0,1,m_acc)
CVaRCont_acc_div=matrix(0,1,m)
CVaRfr1_acc_div=matrix(0,1,m_fx)
CVaRfr2_acc_div=matrix(0,1,m_acc)
for (i in (1:m))
{
  VaRCont_acc_div[i]=quantile(PG_acc_div[,i],1-alpha,Ns)
  CVaRCont_acc_div[i]= mean(merge(which(PG_acc_div[,i]<VaRCont_acc_div[i]),cbind(seq(1,Ns),PG_acc_div[,
  if (i<=m_fx)
  {
  VaRfr1_acc_div[i]=quantile(PGfr1_acc_div[,i],1-alpha,Ns)
  CVaRfr1_acc_div[i]= mean(merge(which(PGfr1_acc_div[,i]<VaRfr1_acc_div[i]),cbind(seq(1,Ns),PGfr1_acc_di
  }
  if (i<=m_acc)
  {
```

```
  VaRfr2_acc_div[i]=quantile(PGfr2_acc_div[,i],1-alpha,Ns)
  CVaRfr2_acc_div[i]= mean(merge(which(PGfr2_acc_div[,i]<VaRfr2_acc_div[i]),cbind(seq(1,Ns),PGfr2_acc_di
  }
}
#Met
#VaRCont_acc_div
#VaRfr1_acc_div
#VaRfr2_acc_div
#CVaRCont_acc_div
#CVaRfr1_acc_div
#CVaRfr2_acc_div


#VaR Total
#Met
VaRTotal_acc_div=quantile(PGT_acc_div,1-alpha,Ns)
CVaRTotal_acc_div= mean(merge(which(PGT_acc_div<VaRTotal_acc_div),cbind(seq(1,Ns),PGT_acc_div), by.x=1,b
VaRTotalfr1_acc_div=quantile(PGfr1T_acc_div,1-alpha,Ns)
CVaRTotalfr1_acc_div= mean(PGfr1T_acc_div[which(PGfr1T_acc_div<VaRTotalfr1_acc_div),])
VaRTotalfr2_acc_div=quantile(PGfr2T_acc_div,1-alpha,Ns)
CVaRTotalfr2_acc_div= mean(PGfr2T_acc_div[which(PGfr2T_acc_div<VaRTotalfr2_acc_div),])

#print("Var Empírico")
#print(cbind(VaRTotal_acc_div,sum(VO_acc_div), VaRCont_acc_div, VO_acc_div))
#print("CVar Empírico")
#print(cbind(CVaRTotal_acc_div,sum(VO_acc_div), CVaRCont_acc_div, VO_acc_div))
#print("Var Normal")
#print(cbind(VaRTotal_acc_div,VaRTotalfr1_acc_div,,VaRTotalfr2_acc_div))
#print("CVar Normal")
#print(cbind(CVaRTotal_acc_div,CVaRTotalfr1_acc_div,CVaRTotalfr2_acc_div))
#print(sum(VO_acc_div))
titles<-c('EURUSD.X', 'GBPUSD.X', 'USDMXN.X', 'AMXL.MX', 'GCARSOA1.MX',  'WALMEX.MX')
```

```
print("Var Normal")
```

```
## [1] "Var Normal"
```

```
print('Var total')
```

```
## [1] "Var total"
```

```
print(VaRTotal_acc_div)
```

```
##       2%
## -812.925
```

```
print('Var por posición')
```

```
## [1] "Var por posición"
```

```
print(cbind(titles,t(VaRCont_acc_div)))
```

```
##      titles
## [1,] "EURUSD.X"    "-28.8567919335074"
## [2,] "GBPUSD.X"    "-20.6392192368237"
## [3,] "USDMXN.X"    "-74.9466105988102"
## [4,] "AMXL.MX"     "-546.868367014409"
```

```
## [5,] "GCARSOA1.MX" "-914.112804217568"
## [6,] "WALMEX.MX"   "-609.251248894801"
```

```
print('Var del contrato acciones')
```

```
## [1] "Var del contrato acciones"
```

```
print(VaRTotalfr2_acc_div)
```

```
##        2%
## -779.5127
```

```
print('Var del contrato divisas')
```

```
## [1] "Var del contrato divisas"
```

```
print(VaRTotalfr1_acc_div)
```

```
##        2%
## -83.31467
```

```
print("CVar Empírico")
```

```
## [1] "CVar Empírico"
```

```
print('CVar total')
```

```
## [1] "CVar total"
```

```
print(CVaRTotal_acc_div)
```

```
## [1] -952.5225
```

```
print('CVar por posición')
```

```
## [1] "CVar por posición"
```

```
print(cbind(titles,t(CVaRCont_acc_div)))
```

```
##      titles
## [1,] "EURUSD.X"    "-33.6331952664941"
## [2,] "GBPUSD.X"    "-24.1752064658914"
## [3,] "USDMXN.X"    "-88.5019316443619"
## [4,] "AMXL.MX"     "-645.429358313214"
## [5,] "GCARSOA1.MX" "-1083.02172026711"
## [6,] "WALMEX.MX"   "-706.757791244254"
```

```
print('CVar del contrato acciones')
```

```
## [1] "CVar del contrato acciones"
```

```
print(CVaRTotalfr2_acc_div)
```

```
## [1] -913.4088
```

```
print('CVar del contrato divisas')
```

```
## [1] "CVar del contrato divisas"
```

```
print(CVaRTotalfr1_acc_div)
```

```
## [1] -98.30021
```

Bonos

```
#Cálculo de matriz de pérdidas y ganancias BONDES
#dimensión
m=count(N_bd)    #PASO CLAVE
X_s_bd=matrix(0,Ns,n_if[4]) #Factores de riesgo simulados con base en DeltaX_s x0*(1+Delta_Xs)   #PASO
V_bd=matrix(0,Ns,m)
Vfr1_bd=matrix(0,Ns,m)
Vfr2_bd=matrix(0,Ns,m)
Vfr3_bd=matrix(0,Ns,m)
PG_bd=matrix(0,Ns,m) #Pèrdidas y ganancias
PGfr1_bd=matrix(0,Ns,m)
PGfr2_bd=matrix(0,Ns,m)
PGfr3_bd=matrix(0,Ns,m)
PGT_bd=matrix(0,Ns,1)
PGfr1T_bd=matrix(0,Ns,1)
PGfr2T_bd=matrix(0,Ns,1)
PGfr3T_bd=matrix(0,Ns,1)

DeltaX_s_bd=DeltaX_s[,sum(n_if[1:3],1):sum(n_if[1:4])]    #PASO CLAVE
x0_bd=X_bd_ext[1,]    #PASO CLAVE

for (i in 1:Ns)
{
  X_s_bd[i,]=x0_bd*(1+DeltaX_s_bd[i,])
  #PASO CLAVE
  V_bd[i,]=bondeD(contratos_bdmT, nominal_bdm, X_s_bd[i,1], plazocupon_bdmT, VTplazos_bdm, X_s_bd[i,(n_
  #PASO CLAVE
  Vfr1_bd[i,]=bondeD(contratos_bdmT, nominal_bdm, X_s_bd[i,1], plazocupon_bdmT, VTplazos_bdm, x0_bd[(n_
  #PASO CLAVE
  Vfr2_bd[i,]=bondeD(contratos_bdmT, nominal_bdm, x0_bd[1], plazocupon_bdmT, VTplazos_bdm, X_s_bd[i,(n_
  #PASO CLAVE
  Vfr3_bd[i,]=bondeD(contratos_bdmT, nominal_bdm, x0_bd[1], plazocupon_bdmT, VTplazos_bdm, x0_bd[(n_if[
  #PASO CLAVE
  PG_bd[i,]=V_bd[i,]-V0_bd
  PGfr1_bd[i,]=Vfr1_bd[i,]-V0_bd
  PGfr2_bd[i,]=Vfr2_bd[i,]-V0_bd
  PGfr3_bd[i,]=Vfr3_bd[i,]-V0_bd
  PGT_bd[i,]=sum(PG_bd[i,])
  PGfr1T_bd[i,]=sum(PGfr1_bd[i,])
  PGfr2T_bd[i,]=sum(PGfr2_bd[i,])
  PGfr3T_bd[i,]=sum(PGfr3_bd[i,])
}



PG_bd[1:5,]
```

```
## [1] -1080.85001 -2787.46430 -1568.57320  -520.39648    13.93439
```

```
PGfr1_bd[1:5,]
```

```
## [1]   776.6228 -1406.1831   376.1003  -490.3228   124.1604
```

```
PGfr2_bd[1:5,]
```

```
## [1] -1364.93145 -1015.73163 -1429.57836   -22.30366   -80.77971
```

```
PGT_bd[1:5,]
```

```
## [1] -1080.85001 -2787.46430 -1568.57320  -520.39648    13.93439
```

```
#VaR por posición
VaRCont_bd=matrix(0,1,m)
VaRfr1_bd=matrix(0,1,m)
VaRfr2_bd=matrix(0,1,m)
VaRfr3_bd=matrix(0,1,m)
CVaRCont_bd=matrix(0,1,m)
CVaRfr1_bd=matrix(0,1,m)
CVaRfr2_bd=matrix(0,1,m)
CVaRfr3_bd=matrix(0,1,m)
for (i in (1:m))
{
  VaRCont_bd[i]=quantile(PG_bd[,i],1-alpha,Ns)
  VaRfr1_bd[i]=quantile(PGfr1_bd[,i],1-alpha,Ns)
  VaRfr2_bd[i]=quantile(PGfr2_bd[,i],1-alpha,Ns)
  VaRfr3_bd[i]=quantile(PGfr3_bd[,i],1-alpha,Ns)
  CVaRfr1_bd[i]= mean(merge(which(PGfr1_bd[,i]<VaRfr1_bd[i]),cbind(seq(1,Ns),PGfr1_bd[,i]), by.x=1,by.y=
  CVaRfr2_bd[i]= mean(merge(which(PGfr2_bd[,i]<VaRfr2_bd[i]),cbind(seq(1,Ns),PGfr2_bd[,i]), by.x=1,by.y=
  CVaRfr3_bd[i]= mean(merge(which(PGfr3_bd[,i]<VaRfr3_bd[i]),cbind(seq(1,Ns),PGfr3_bd[,i]), by.x=1,by.y=
  CVaRCont_bd[i]= mean(merge(which(PG_bd[,i]<VaRCont_bd[i]),cbind(seq(1,Ns),PG_bd[,i]), by.x=1,by.y=1)[
}
Met
```

```
## [1] "Componentes Principales Normales"
```

```
VaRCont_bd
```

```
##           [,1]
## [1,] -3987.294
```

```
VaRfr1_bd
```

```
##           [,1]
## [1,] -2999.556
```

```
VaRfr2_bd
```

```
##           [,1]
## [1,] -1872.088
```

```
CVaRCont_bd
```

```
##           [,1]
## [1,] -4700.824
```

```
CVaRfr1_bd
```

```
##           [,1]
## [1,] -3494.747
```

```
CVaRfr2_bd
```

```
##           [,1]
## [1,] -2205.464
```

```
#VaR Total
#Met
VaRTotal_bd=quantile(PGT_bd,1-alpha,Ns)
CVaRTotal_bd= mean(merge(which(PGT_bd<VaRTotal_bd),cbind(seq(1,Ns),PGT_bd), by.x=1,by.y=1)[,2])
VaRTotalfr1_bd=quantile(PGfr1T_bd,1-alpha,Ns)
CVaRTotalfr1_bd= mean(PGfr1T_bd[which(PGfr1T_bd<VaRTotalfr1_bd),])
VaRTotalfr2_bd=quantile(PGfr2T_bd,1-alpha,Ns)
CVaRTotalfr2_bd= mean(PGfr2T_bd[which(PGfr2T_bd<VaRTotalfr2_bd),])
VaRTotalfr3_bd=quantile(PGfr3T_bd,1-alpha,Ns)
CVaRTotalfr3_bd= mean(PGfr3T_bd[which(PGfr2T_bd<VaRTotalfr2_bd),])


#cbind(VaRTotal_bd,sum(V0_bd), VaRCont_bd, V0_bd)
#cbind(CVaRTotal_bd,sum(V0_bd), CVaRCont_bd, V0_bd)
#cbind(VaRTotal_bd,VaRTotalfr1_bd,VaRTotalfr2_bd,VaRTotalfr3_bd)
#cbind(CVaRTotal_bd,CVaRTotalfr1_bd,CVaRTotalfr2_bd,CVaRTotalfr3_bd)
lista<-c('Tasa descuento','tasa fondeo', 'sobretasa')
```

```
print('Var total')
```

```
## [1] "Var total"
```

```
print(VaRTotal_bd)
```

```
##        2%
## -3987.294
```

```
print('Var por factor de riesgo')
```

```
## [1] "Var por factor de riesgo"
```

```
print(rbind(lista,t(c(VaRTotalfr1_bd,VaRTotalfr2_bd,VaRTotalfr3_bd))))
```

```
##       2%                   2%                   2%
## lista "Tasa descuento"     "tasa fondeo"        "sobretasa"
##       "-2999.55571618987" "-1872.08765504774" "-676.632793671188"
```

```
print('CVar total')
```

```
## [1] "CVar total"
```

```
print(CVaRTotal_bd)
```

```
## [1] -4700.824
```

```
print('CVar por factor de riesgo')
```

```
## [1] "CVar por factor de riesgo"
```

```
print(rbind(lista,t(c(CVaRTotalfr1_bd,CVaRTotalfr2_bd,CVaRTotalfr3_bd))))
```

```
##       [,1]                 [,2]                 [,3]
## lista "Tasa descuento"     "tasa fondeo"        "sobretasa"
##       "-3494.74652536273" "-2205.46439386204" "-797.849124672862"
```

Forwards de Divisas

```
#Cálculo de matriz de pérdidas y ganancias FUTUROS TDC
#dimensión
```

```
m=ncol(plazos_fwd)  #PASO CLAVE
X_s_fwtdc=matrix(0,Ns,n_if[5]) #Factores de riesgo simulados con base en DeltaX_s x0*(1+Delta_Xs) #PASO
V_fwtdc=matrix(0,Ns,m)
Vfr1_fwtdc=matrix(0,Ns,m)
Vfr2_fwtdc=matrix(0,Ns,m)
Vfr3_fwtdc=matrix(0,Ns,m)
PG_fwtdc=matrix(0,Ns,m) #Pèrdidas y ganancias
PGfr1_fwtdc=matrix(0,Ns,m)
PGfr2_fwtdc=matrix(0,Ns,m)
PGfr3_fwtdc=matrix(0,Ns,m)
PGT_fwtdc=matrix(0,Ns,1)
PGfr1T_fwtdc=matrix(0,Ns,1)
PGfr2T_fwtdc=matrix(0,Ns,1)
PGfr3T_fwtdc=matrix(0,Ns,1)

DeltaX_s_fwtdc=DeltaX_s[,sum(n_if[1:4],1):sum(n_if[1:5])]  #PASO CLAVE
x0_fwtdc=X_futtdc[1,] #PASO CLAVE

for (i in 1:Ns)
{
  X_s_fwtdc[i,]=x0_fwtdc*(1+DeltaX_s_fwtdc[i,])
  #PASO CLAVE
  V_fwtdc[i,]=futuroTC(plazos_fwd,X_s_fwtdc[i,1:((n_if[5]-1)/2)],X_s_fwtdc[i,((n_if[5]-1)/2+1):(n_if[5]
  #PASO CLAVE
  Vfr1_fwtdc[i,]=futuroTC(plazos_fwd,X_s_fwtdc[i,1:((n_if[5]-1)/2)],x0_fwtdc[((n_if[5]-1)/2+1):(n_if[5]
  #PASO CLAVE
  Vfr2_fwtdc[i,]=futuroTC(plazos_fwd,x0_fwtdc[1:((n_if[5]-1)/2)],X_s_fwtdc[i,((n_if[5]-1)/2+1):(n_if[5]
  #PASO CLAVE
  Vfr3_fwtdc[i,]=futuroTC(plazos_fwd,x0_fwtdc[1:((n_if[5]-1)/2)],x0_fwtdc[((n_if[5]-1)/2+1):(n_if[5]-1)]
  PG_fwtdc[i,]=V_fwtdc[i,]-V0_fwtdc
  PGfr1_fwtdc[i,]=Vfr1_fwtdc[i,]-V0_fwtdc
  PGfr2_fwtdc[i,]=Vfr2_fwtdc[i,]-V0_fwtdc
  PGfr3_fwtdc[i,]=Vfr3_fwtdc[i,]-V0_fwtdc
  PGT_fwtdc[i,]=sum(PG_fwtdc[i,])
  PGfr1T_fwtdc[i,]=sum(PGfr1_fwtdc[i,])
  PGfr2T_fwtdc[i,]=sum(PGfr2_fwtdc[i,])
  PGfr3T_fwtdc[i,]=sum(PGfr3_fwtdc[i,])
}



PG_fwtdc[1:5,]
```

```
## [1] -2.3767902  1.4111458 -1.0206776 -0.6350975 -2.0676759
```

```
PGfr1_fwtdc[1:5,]
```

```
## [1] -0.0001894976 -0.0016095287 -0.0030303012  0.0004935736 -0.0021022975
```

```
PGfr2_fwtdc[1:5,]
```

```
## [1]  0.0003368803 -0.0178284667 -0.0289418443  0.0044437818 -0.0222333725
```

```
PGT_fwtdc[1:5,]
```

```
## [1] -2.3767902  1.4111458 -1.0206776 -0.6350975 -2.0676759
```

```
#VaR por posición
VaRCont_fwtdc=matrix(0,1,m)
VaRfr1_fwtdc=matrix(0,1,m)
VaRfr2_fwtdc=matrix(0,1,m)
VaRfr3_fwtdc=matrix(0,1,m)
CVaRCont_fwtdc=matrix(0,1,m)
CVaRfr1_fwtdc=matrix(0,1,m)
CVaRfr2_fwtdc=matrix(0,1,m)
CVaRfr3_fwtdc=matrix(0,1,m)
for (i in (1:m))
{
  VaRCont_fwtdc[i]=quantile(PG_fwtdc[,i],1-alpha,Ns)
  VaRfr1_fwtdc[i]=quantile(PGfr1_fwtdc[,i],1-alpha,Ns)
  VaRfr2_fwtdc[i]=quantile(PGfr2_fwtdc[,i],1-alpha,Ns)
  VaRfr3_fwtdc[i]=quantile(PGfr3_fwtdc[,i],1-alpha,Ns)
  CVaRfr1_fwtdc[i]= mean(merge(which(PGfr1_fwtdc[,i]<VaRfr1_fwtdc[i]),cbind(seq(1,Ns),PGfr1_fwtdc[,i]),
  CVaRfr2_fwtdc[i]= mean(merge(which(PGfr2_fwtdc[,i]<VaRfr2_fwtdc[i]),cbind(seq(1,Ns),PGfr2_fwtdc[,i]),
  CVaRfr3_fwtdc[i]= mean(merge(which(PGfr3_fwtdc[,i]<VaRfr3_fwtdc[i]),cbind(seq(1,Ns),PGfr3_fwtdc[,i]),
  CVaRCont_fwtdc[i]= mean(merge(which(PG_fwtdc[,i]<VaRCont_fwtdc[i]),cbind(seq(1,Ns),PG_fwtdc[,i]), by.
}
#Met
#VaRCont_fwtdc
#VaRfr1_fwtdc
#VaRfr2_fwtdc
#CVaRCont_fwtdc
#CVaRfr1_fwtdc
#CVaRfr2_fwtdc


#VaR Total
Met
```

```
## [1] "Componentes Principales Normales"
```

```
VaRTotal_fwtdc=quantile(PGT_fwtdc,1-alpha,Ns)
CVaRTotal_fwtdc= mean(merge(which(PGT_fwtdc<VaRTotal_fwtdc),cbind(seq(1,Ns),PGT_fwtdc), by.x=1,by.y=1)[
VaRTotalfr1_fwtdc=quantile(PGfr1T_fwtdc,1-alpha,Ns)
CVaRTotalfr1_fwtdc= mean(PGfr1T_fwtdc[which(PGfr1T_fwtdc<VaRTotalfr1_fwtdc),])
VaRTotalfr2_fwtdc=quantile(PGfr2T_fwtdc,1-alpha,Ns)
CVaRTotalfr2_fwtdc= mean(PGfr2T_fwtdc[which(PGfr2T_fwtdc<VaRTotalfr2_fwtdc),])
VaRTotalfr3_fwtdc=quantile(PGfr3T_fwtdc,1-alpha,Ns)
CVaRTotalfr3_fwtdc= mean(PGfr3T_fwtdc[which(PGfr2T_fwtdc<VaRTotalfr2_fwtdc),])


#cbind(VaRTotal_fwtdc,sum(VO_fwtdc), VaRCont_fwtdc, VO_fwtdc)
#cbind(CVaRTotal_fwtdc,sum(VO_fwtdc), CVaRCont_fwtdc, VO_fwtdc)
cbind(VaRTotal_fwtdc,VaRTotalfr1_fwtdc,VaRTotalfr2_fwtdc,VaRTotalfr3_fwtdc)
```

```
##    VaRTotal_fwtdc VaRTotalfr1_fwtdc VaRTotalfr2_fwtdc VaRTotalfr3_fwtdc
## 2%       -5.00105       -0.00366638       -0.03491962          -4.99376
```

```
cbind(CVaRTotal_fwtdc,CVaRTotalfr1_fwtdc,CVaRTotalfr2_fwtdc,CVaRTotalfr3_fwtdc)
```

```
##      CVaRTotal_fwtdc CVaRTotalfr1_fwtdc CVaRTotalfr2_fwtdc CVaRTotalfr3_fwtdc
## [1,]       -5.896804       -0.004341415        -0.04249788         -0.4662096
```

Forwards IPC

```
#Cálculo de matriz de pérdidas y ganancias FUTUROS IPC
m=ncol(plazos_fwd_ind)  #PASO CLAVE
X_s_fwind=matrix(0,Ns,n_if[6]) #Factores de riesgo simulados con base en DeltaX_s x0*(1+Delta_Xs) #PASO
V_fwind=matrix(0,Ns,m)
Vfr1_fwind=matrix(0,Ns,m)
Vfr2_fwind=matrix(0,Ns,m)
Vfr3_fwind=matrix(0,Ns,m)
PG_fwind=matrix(0,Ns,m) #Pèrdidas y ganancias
PGfr1_fwind=matrix(0,Ns,m)
PGfr2_fwind=matrix(0,Ns,m)
PGfr3_fwind=matrix(0,Ns,m)
PGT_fwind=matrix(0,Ns,1)
PGfr1T_fwind=matrix(0,Ns,1)
PGfr2T_fwind=matrix(0,Ns,1)
PGfr3T_fwind=matrix(0,Ns,1)

DeltaX_s_fwind=DeltaX_s[,sum(n_if[1:5],1):sum(n_if[1:6])]  #PASO CLAVE
x0_fwind=X_futind[1,] #PASO CLAVE


for (i in 1:Ns)
{
  X_s_fwind[i,]=x0_fwind*(1+DeltaX_s_fwind[i,])
  #PASO CLAVE
  V_fwind[i,]=futuroTC(plazos_fwd_ind,X_s_fwind[i,1:(n_if[6]/3)],X_s_fwind[i,(n_if[6]/3+1):(n_if[6]*2/3)
  Vfr1_fwind[i,]=futuroTC(plazos_fwd_ind,X_s_fwind[i,1:(n_if[6]/3)],x0_fwind[(n_if[6]/3+1):(n_if[6]*2/3)
  #PASO CLAVE
  Vfr2_fwind[i,]=futuroTC(plazos_fwd_ind,x0_fwind[1:(n_if[6]/3)],X_s_fwind[i,(n_if[6]/3+1):(n_if[6]*2/3)
  #PASO CLAVE
  Vfr3_fwind[i,]=futuroTC(plazos_fwd_ind,x0_fwind[1:(n_if[6]/3)],x0_fwind[(n_if[6]/3+1):(n_if[6]*2/3)],X
  PG_fwind[i,]=V_fwind[i,]-V0_fwind
  PGfr1_fwind[i,]=Vfr1_fwind[i,]-V0_fwind
  PGfr2_fwind[i,]=Vfr2_fwind[i,]-V0_fwind
  PGfr3_fwind[i,]=Vfr3_fwind[i,]-V0_fwind
  PGT_fwind[i,]=sum(PG_fwind[i,])
  PGfr1T_fwind[i,]=sum(PGfr1_fwind[i,])
  PGfr2T_fwind[i,]=sum(PGfr2_fwind[i,])
  PGfr3T_fwind[i,]=sum(PGfr3_fwind[i,])
}


PG_fwind[1:5,]
```

```
## [1]  6085.75329 -2100.36855  3485.83188 -1279.18479    15.08106
```

```
#PGfr1_fwind[1:5,]
#PGfr2_fwind[1:5,]
#PGT_fwind[1:5,]



#VaR por posición
VaRCont_fwind=matrix(0,1,m)
```

```
VaRfr1_fwind=matrix(0,1,m)
VaRfr2_fwind=matrix(0,1,m)
VaRfr3_fwind=matrix(0,1,m)
CVaRCont_fwind=matrix(0,1,m)
CVaRfr1_fwind=matrix(0,1,m)
CVaRfr2_fwind=matrix(0,1,m)
CVaRfr3_fwind=matrix(0,1,m)
for (i in (1:m))
{
  VaRCont_fwind[i]=quantile(PG_fwind[,i],1-alpha,Ns)
  VaRfr1_fwind[i]=quantile(PGfr1_fwind[,i],1-alpha,Ns)
  VaRfr2_fwind[i]=quantile(PGfr2_fwind[,i],1-alpha,Ns)
  VaRfr3_fwind[i]=quantile(PGfr3_fwind[,i],1-alpha,Ns)
  CVaRfr1_fwind[i]= mean(merge(which(PGfr1_fwind[,i]<VaRfr1_fwind[i]),cbind(seq(1,Ns),PGfr1_fwind[,i]),
  CVaRfr2_fwind[i]= mean(merge(which(PGfr2_fwind[,i]<VaRfr2_fwind[i]),cbind(seq(1,Ns),PGfr2_fwind[,i]),
  CVaRfr3_fwind[i]= mean(merge(which(PGfr3_fwind[,i]<VaRfr3_fwind[i]),cbind(seq(1,Ns),PGfr3_fwind[,i]),
  CVaRCont_fwind[i]= mean(merge(which(PG_fwind[,i]<VaRCont_fwind[i]),cbind(seq(1,Ns),PG_fwind[,i]), by.
}
#Met
#VaRCont_fwind
#VaRfr1_fwind
#VaRfr2_fwind
#CVaRCont_fwind
#CVaRfr1_fwind
#CVaRfr2_fwind


#VaR Total
Met
```

```
## [1] "Componentes Principales Normales"
```

```
VaRTotal_fwind=quantile(PGT_fwind,1-alpha,Ns)
CVaRTotal_fwind= mean(merge(which(PGT_fwind<VaRTotal_fwind),cbind(seq(1,Ns),PGT_fwind), by.x=1,by.y=1)[
VaRTotalfr1_fwind=quantile(PGfr1T_fwind,1-alpha,Ns)
CVaRTotalfr1_fwind= mean(PGfr1T_fwind[which(PGfr1T_fwind<VaRTotalfr1_fwind),])
VaRTotalfr2_fwind=quantile(PGfr2T_fwind,1-alpha,Ns)
CVaRTotalfr2_fwind= mean(PGfr2T_fwind[which(PGfr2T_fwind<VaRTotalfr2_fwind),])
VaRTotalfr3_fwind=quantile(PGfr3T_fwind,1-alpha,Ns)
CVaRTotalfr3_fwind= mean(PGfr3T_fwind[which(PGfr3T_fwind<VaRTotalfr3_fwind),])


#print(cbind(VaRTotal_fwind,sum(V0_fwind), VaRCont_fwind, V0_fwind))
#print(cbind(CVaRTotal_fwind,sum(V0_fwind), CVaRCont_fwind, V0_fwind))
print(cbind(VaRTotal_fwind,VaRTotalfr1_fwind,VaRTotalfr2_fwind,VaRTotalfr3_fwind))
```

```
##    VaRTotal_fwind VaRTotalfr1_fwind VaRTotalfr2_fwind VaRTotalfr3_fwind
## 2%      -9238.828                 0         -460.9364         -9276.065
```

```
cbind(CVaRTotal_fwind,CVaRTotalfr1_fwind,CVaRTotalfr2_fwind,CVaRTotalfr3_fwind)
```

```
##      CVaRTotal_fwind CVaRTotalfr1_fwind CVaRTotalfr2_fwind CVaRTotalfr3_fwind
## [1,]       -11130.51                NaN          -531.9058          -11142.47
```

Swaps

```
#Cálculo de matriz de pérdidas y ganancias SWAP

#riesgo del swap
m=ncol(N)    #PASO CLAVE
X_s_sw=matrix(0,Ns,n_if[7]) #Factores de riesgo simulados con base en DeltaX_s x0*(1+Delta_Xs)   #PASO
V_sw=matrix(0,Ns,m)
Vfr1_sw=matrix(0,Ns,m)
Vfr2_sw=matrix(0,Ns,m)
PG_sw=matrix(0,Ns,m) #Pèrdidas y ganancias
PGfr1_sw=matrix(0,Ns,m)
PGfr2_sw=matrix(0,Ns,m)
PGT_sw=matrix(0,Ns,1)
PGfr1T_sw=matrix(0,Ns,1)
PGfr2T_sw=matrix(0,Ns,1)

DeltaX_s_sw=DeltaX_s[,sum(n_if[1:6],1):sum(n_if[1:7])]    #PASO CLAVE
x0_sw=as.numeric(c(XtfwdT[1,],Xvp[1,]))   #PASO CLAVE

for (i in 1:Ns)
{
  X_s_sw[i,]=x0_sw*(1+DeltaX_s_sw[i,])
  #PASO CLAVE
  V_sw[i,]=swap(por_swT, contratos_swT, nominal_swT, X_s_sw[i,1:(n_if[7]/2)], tasafija_swT, plazocupon_s
  #PASO CLAVE
  Vfr1_sw[i,]=swap(por_swT, contratos_swT, nominal_swT,X_s_sw[i,1:(n_if[7]/2)], tasafija_swT, plazocupo
  #PASO CLAVE
  Vfr2_sw[i,]=swap(por_swT, contratos_swT, nominal_swT,  XtfwdT[1,], tasafija_swT, plazocupon_swT, VTpla
  #PASO CLAVE
  PG_sw[i,]=V_sw[i,]-V0_sw
  PGfr1_sw[i,]=Vfr1_sw[i,]-V0_sw
  PGfr2_sw[i,]=Vfr2_sw[i,]-V0_sw
  PGT_sw[i,]=sum(PG_sw[i,])
  PGfr1T_sw[i,]=sum(PGfr1_sw[i,])
  PGfr2T_sw[i,]=sum(PGfr2_sw[i,])
}



PG_sw[1:5,]
```

```
##            [,1]       [,2]
## [1,]  11196.881  -960.0128
## [2,] -21904.706  6842.9718
## [3,] -30499.742 13536.0469
## [4,]   6827.133 -1788.5546
## [5,] -20236.970  7300.7748
```

```
PGfr1_sw[1:5,]
```

```
##            [,1]       [,2]
## [1,]  11217.923  -953.5755
## [2,] -21848.453  6862.2103
## [3,] -30397.709 13565.3364
```

```
## [4,]   6814.947 -1794.7963
## [5,] -20150.902  7328.7454
```

```
PGfr2_sw[1:5,]
```

```
##           [,1]       [,2]
## [1,] -26.08632  -6.320952
## [2,] -43.51591 -21.149602
## [3,] -78.89375 -36.057480
## [4,]  13.50266   6.093011
## [5,] -69.63303 -31.114156
```

```
PGT_sw[1:5,]
```

```
## [1]  10236.868 -15061.734 -16963.695   5038.578 -12936.196
```

```
#VaR por posición
VaRCont_sw=matrix(0,1,m)
VaRfr1_sw=matrix(0,1,m)
VaRfr2_sw=matrix(0,1,m)
CVaRCont_sw=matrix(0,1,m)
CVaRfr1_sw=matrix(0,1,m)
CVaRfr2_sw=matrix(0,1,m)
for (i in (1:m))
{
  VaRCont_sw[i]=quantile(PG_sw[,i],1-alpha,Ns)
  VaRfr1_sw[i]=quantile(PGfr1_sw[,i],1-alpha,Ns)
  VaRfr2_sw[i]=quantile(PGfr2_sw[,i],1-alpha,Ns)
  CVaRfr1_sw[i]= mean(merge(which(PGfr1_sw[,i]<VaRfr1_sw[i]),cbind(seq(1,Ns),PGfr1_sw[,i]), by.x=1,by.y
  CVaRfr2_sw[i]= mean(merge(which(PGfr2_sw[,i]<VaRfr2_sw[i]),cbind(seq(1,Ns),PGfr2_sw[,i]), by.x=1,by.y
  CVaRCont_sw[i]= mean(merge(which(PG_sw[,i]<VaRCont_sw[i]),cbind(seq(1,Ns),PG_sw[,i]), by.x=1,by.y=1)[
}
#Met
#VaRCont_sw
#VaRfr1_sw
#VaRfr2_sw
#CVaRCont_sw
#CVaRfr1_sw
#CVaRfr2_sw


#VaR Total
Met
```

```
## [1] "Componentes Principales Normales"
```

```
VaRTotal_sw=quantile(PGT_sw,1-alpha,Ns)
CVaRTotal_sw= mean(merge(which(PGT_sw<VaRTotal_sw),cbind(seq(1,Ns),PGT_sw), by.x=1,by.y=1)[,2])
VaRTotalfr1_sw=quantile(PGfr1T_sw,1-alpha,Ns)
CVaRTotalfr1_sw= mean(PGfr1T_sw[which(PGfr1T_sw<VaRTotalfr1_sw),])
VaRTotalfr2_sw=quantile(PGfr2T_sw,1-alpha,Ns)
CVaRTotalfr2_sw= mean(PGfr2T_sw[which(PGfr2T_sw<VaRTotalfr2_sw),])


cbind(VaRTotal_sw, VaRCont_sw)
```

```
##    VaRTotal_sw
```

```
## 2%   -27515.13 -41366.33 -15345.07
```

```
cbind(CVaRTotal_sw, CVaRCont_sw)
```

```
##      CVaRTotal_sw
## [1,]     -32975.1 -49367.07 -17932.85
```

```
cbind(VaRTotal_sw,VaRTotalfr1_sw,VaRTotalfr2_sw)
```

```
##    VaRTotal_sw VaRTotalfr1_sw VaRTotalfr2_sw
## 2%   -27515.13      -27307.61      -161.0642
```

```
cbind(CVaRTotal_sw,CVaRTotalfr1_sw,CVaRTotalfr2_sw)
```

```
##      CVaRTotal_sw CVaRTotalfr1_sw CVaRTotalfr2_sw
## [1,]     -32975.1       -32777.75       -191.4369
```

Opciones de tasa de interés

```
#Cálculo de matriz de pérdidas y ganancias Opciones Tasa de interés
#dimensión
m=ncol(plazos_oir)  #PASO CLAVE
X_s_oir=matrix(0,Ns,n_if[8]) #Factores de riesgo simulados con base en DeltaX_s x0*(1+Delta_Xs) #PASO C
V_oir=matrix(0,Ns,m)
Vfr1_oir=matrix(0,Ns,m)
Vfr2_oir=matrix(0,Ns,m)
Vfr3_oir=matrix(0,Ns,m)
PG_oir=matrix(0,Ns,m) #Pèrdidas y ganancias
PGfr1_oir=matrix(0,Ns,m)
PGfr2_oir=matrix(0,Ns,m)
PGfr3_oir=matrix(0,Ns,m)
PGT_oir=matrix(0,Ns,1)
PGfr1T_oir=matrix(0,Ns,1)
PGfr2T_oir=matrix(0,Ns,1)
PGfr3T_oir=matrix(0,Ns,1)

DeltaX_s_oir=DeltaX_s[,sum(n_if[1:7],1):sum(n_if[1:8])]  #PASO CLAVE
x0_oir=X_oir[1,] #PASO CLAVE

for (i in 1:Ns)
{
  X_s_oir[i,]=x0_oir*(1+DeltaX_s_oir[i,])
  #PASO CLAVE
  V_oir[i,]=  opctint(X_s_oir[i,(1:(n_if[8]/3))],X_s_oir[i,((n_if[8]/3+1):(n_if[8]/3*2))],K_oir,X_s_oir
  #PASO CLAVE
  Vfr1_oir[i,]=opctint(X_s_oir[i,(1:(n_if[8]/3))],x0_oir[((n_if[8]/3+1):(n_if[8]/3*2))],K_oir,x0_oir[((
  #PASO CLAVE
  Vfr2_oir[i,]=opctint(x0_oir[(1:(n_if[8]/3))],X_s_oir[i,((n_if[8]/3+1):(n_if[8]/3*2))],K_oir,x0_oir[((
  #PASO CLAVE
  Vfr3_oir[i,]=opctint(x0_oir[(1:(n_if[8]/3))],x0_oir[((n_if[8]/3+1):(n_if[8]/3*2))],K_oir,X_s_oir[i,((
  PG_oir[i,]=V_oir[i,]-V0_oir
  PGfr1_oir[i,]=Vfr1_oir[i,]-V0_oir
  PGfr2_oir[i,]=Vfr2_oir[i,]-V0_oir
  PGfr3_oir[i,]=Vfr3_oir[i,]-V0_oir
  PGT_oir[i,]=sum(PG_oir[i,])
  PGfr1T_oir[i,]=sum(PGfr1_oir[i,])
```

```
  PGfr2T_oir[i,]=sum(PGfr2_oir[i,])
  PGfr3T_oir[i,]=sum(PGfr3_oir[i,])
}



PG_oir[1:5,]
```

```
##              [,1]         [,2]
## [1,] -0.013297617  0.021174659
## [2,]  0.004296709 -0.023443216
## [3,]  0.036129313  0.003520552
## [4,]  0.006843204  0.017231792
## [5,] -0.012797097 -0.022911907
```

```
PGfr1_oir[1:5,]
```

```
##               [,1]          [,2]
## [1,]  0.0020457845  1.830327e-04
## [2,]  0.0022072775  2.628590e-04
## [3,]  0.0043344132  4.929508e-04
## [4,] -0.0007155997 -8.128916e-05
## [5,]  0.0039798572  4.306220e-04
```

```
PGfr2_oir[1:5,]
```

```
##              [,1]          [,2]
## [1,] -0.024188710  0.0209822099
## [2,]  0.013917184 -0.0178688921
## [3,]  0.015718333 -0.0004752578
## [4,] -0.008406576  0.0068984524
## [5,]  0.008904189 -0.0090267990
```

```
PGT_oir[1:5,]
```

```
## [1]  0.007877042 -0.019146508  0.039649865  0.024074996 -0.035709004
```

```
#VaR por posición
VaRCont_oir=matrix(0,1,m)
VaRfr1_oir=matrix(0,1,m)
VaRfr2_oir=matrix(0,1,m)
VaRfr3_oir=matrix(0,1,m)
CVaRCont_oir=matrix(0,1,m)
CVaRfr1_oir=matrix(0,1,m)
CVaRfr2_oir=matrix(0,1,m)
CVaRfr3_oir=matrix(0,1,m)
for (i in (1:m))
{
  VaRCont_oir[i]=quantile(PG_oir[,i],1-alpha,Ns)
  VaRfr1_oir[i]=quantile(PGfr1_oir[,i],1-alpha,Ns)
  VaRfr2_oir[i]=quantile(PGfr2_oir[,i],1-alpha,Ns)
  VaRfr3_oir[i]=quantile(PGfr3_oir[,i],1-alpha,Ns)
  CVaRfr1_oir[i]= mean(merge(which(PGfr1_oir[,i]<VaRfr1_oir[i]),cbind(seq(1,Ns),PGfr1_oir[,i]), by.x=1,
  CVaRfr2_oir[i]= mean(merge(which(PGfr2_oir[,i]<VaRfr2_oir[i]),cbind(seq(1,Ns),PGfr2_oir[,i]), by.x=1,
  CVaRfr3_oir[i]= mean(merge(which(PGfr3_oir[,i]<VaRfr3_oir[i]),cbind(seq(1,Ns),PGfr3_oir[,i]), by.x=1,
  CVaRCont_oir[i]= mean(merge(which(PG_oir[,i]<VaRCont_oir[i]),cbind(seq(1,Ns),PG_oir[,i]), by.x=1,by.y=
```

```
}
#Met
#VaRCont_oir
#VaRfr1_oir
#VaRfr2_oir
#CVaRCont_oir
#CVaRfr1_oir
#CVaRfr2_oir


#VaR Total
#Met
VaRTotal_oir=quantile(PGT_oir,1-alpha,Ns)
CVaRTotal_oir= mean(merge(which(PGT_oir<VaRTotal_oir),cbind(seq(1,Ns),PGT_oir), by.x=1,by.y=1)[,2])
VaRTotalfr1_oir=quantile(PGfr1T_oir,1-alpha,Ns)
CVaRTotalfr1_oir= mean(PGfr1T_oir[which(PGfr1T_oir<VaRTotalfr1_oir),])
VaRTotalfr2_oir=quantile(PGfr2T_oir,1-alpha,Ns)
CVaRTotalfr2_oir= mean(PGfr2T_oir[which(PGfr2T_oir<VaRTotalfr2_oir),])
VaRTotalfr3_oir=quantile(PGfr3T_oir,1-alpha,Ns)
CVaRTotalfr3_oir= mean(PGfr3T_oir[which(PGfr2T_oir<VaRTotalfr2_oir),])


cbind(VaRTotal_oir, VaRCont_oir)
```

```
##    VaRTotal_oir
## 2%  -0.07128067 -0.05528672 -0.04142505
```

```
cbind(CVaRTotal_oir, CVaRCont_oir)
```

```
##      CVaRTotal_oir
## [1,]   -0.08440817 -0.06454015 -0.0487354
```

```
cbind(VaRTotal_oir,VaRTotalfr1_oir,VaRTotalfr2_oir,VaRTotalfr3_oir)
```

```
##    VaRTotal_oir VaRTotalfr1_oir VaRTotalfr2_oir VaRTotalfr3_oir
## 2%  -0.07128067     -0.00762796     -0.02596427     -0.07371672
```

```
cbind(CVaRTotal_oir,CVaRTotalfr1_oir,CVaRTotalfr2_oir,CVaRTotalfr3_oir)
```

```
##      CVaRTotal_oir CVaRTotalfr1_oir CVaRTotalfr2_oir CVaRTotalfr3_oir
## [1,]   -0.08440817     -0.008887088       -0.0306031       0.01950426
```

Riesgo total del portafolio

```
#Medición de riesgo por factor de riesgo de todo el portafolios
#Acciones
#1. Acciones
#2. Forward de IPC
PGPort_ACC=PGfr2T_acc_div+ PGfr3T_fwind #Pérdidas y ganancias
VaRPort_ACC=quantile(PGPort_ACC,1-alpha,Ns) #VaR
CVaRPort_ACC= mean(PGPort_ACC[which(PGPort_ACC<VaRPort_ACC)]) #CVaR

print(VaRPort_ACC)
```

```
##        2%
## -9813.862
```

```
print(CVaRPort_ACC)
```

```
## [1] -11841
```

```
#Tasa de Interés
#1. Dado que swaps y bondes son de tasa de interés usaremos PGT_bd y PGT_sw
#2. Para futuros usaremos PGfr1T_fwtdc y PGfr2T_fwtdc
PGPort_TI=PGT_bd+PGT_sw+PGfr1T_fwtdc+PGfr2T_fwtdc +PGfr2T_fwind+PGfr1T_fwind +PGfr1T_oir +PGfr2T_oir #P
VaRPort_TI=quantile(PGPort_TI,1-alpha,Ns) #VaR
CVaRPort_TI= mean(PGPort_TI[which(PGPort_TI<VaRPort_TI)]) #CVaR
print(VaRPort_TI)
```

```
##        2%
## -27994.71
```

```
print(CVaRPort_TI)
```

```
## [1] -33876.55
```

```
#Tipo de cambio
#1. Dado que swaps y bondes son de tasa de interés no usamos nada
#2. Para futuros usamos sólo PGfr3T_fwtdc
PGPort_TDC=PGfr1T_acc_div+PGfr3T_fwtdc  #Pérdidas y ganancias
VaRPort_TDC=quantile(PGPort_TDC,1-alpha,Ns) #VaR
CVaRPort_TDC= mean(PGPort_TDC[which(PGPort_TDC<VaRPort_TDC)]) #CVaR
print(VaRPort_TDC)
```

```
##        2%
## -88.43817
```

```
print(CVaRPort_TDC)
```

```
## [1] -104.1414
```

```
#Volatilidad
#1. Sólo aplica la volatilidad de Opciones de tasa de interés

PGPort_VOL=PGfr3T_oir  #Pérdidas y ganancias
VaRPort_VOL=quantile(PGPort_VOL,1-alpha,Ns) #VaR
CVaRPort_VOL= mean(PGPort_VOL[which(PGPort_VOL<VaRPort_VOL)]) #CVaR

print(VaRPort_VOL)
```

```
##          2%
## -0.07371672
```

```
print(CVaRPort_VOL)
```

```
## [1] -0.0868671
```

```
#Medición de riesgo de todo el portafolios
#Sumar todos los PGT de todos los instrumentos

PGT_Port=PGPort_ACC+PGPort_TI+PGPort_TDC+PGPort_VOL
VaRTotal_Port=quantile(PGT_Port,1-alpha,Ns) #VaR
CVaRTotal_Port= mean(PGT_Port[which(PGT_Port<VaRTotal_Port)]) #CVaR
print(VaRTotal_Port)
```

```
##        2%
```

```
## -31392.93
```

```
print(CVaRTotal_Port)
```

```
## [1] -38031.69
```

```
print(V0T_port)
```

```
## [1] 1474917
```

‘‘‘