Medición Riesgos Financieros
Práctica 2

Alumno
Manuel de la Llave*

Profesor
Laura García

14 de diciembre de 2020

*manudela@ucm.es

Índice general

1 Primer ejercicio **2**

2 Segundo ejercicio **4**

3 Tercer ejercicio **5**

A Figuras **8**

B Código R **11**

B.1 Ejercicio 1 11

B.2 Ejercicio 2 16

B.3 Ejercicio 3 18

Los scripts, adjuntados junto a este documento, también se pueden consultar en el anexo. Los datos (también adjuntos) han sido extraídos de Thomson Reuters (excepto el EURO STOXX 50, de Yahoo Finance) y para los ejercicios 2 y 3 hemos empleado la semilla `set.seed(1585)` para poder replicar los resultados, aunque no deberían diferir mucho aún sin ella.

1 Primer ejercicio

Obtenga los precios diarios del índice de mercado S&P500 y del oro, desde Enero de 2000 hasta hoy, y

a) Estime un modelo AR(1)-GJRGARCH(1,1) para las rentabilidades de cada uno de los dos activos, bajo el supuesto de Normalidad de las innovaciones.

   Para este ejercicio hemos escogido el oro que cotiza en el London Bullion Market, desde enero de 2000 hasta el 25 de noviembre de 2020. Los resultados de la estimación son los siguientes:

	x
mu	0.04
ar1	-0.00
omega	0.01
alpha1	0.07
beta1	0.94
gamma1	-0.03

	x
mu	0.02
ar1	-0.06
omega	0.02
alpha1	0.00
beta1	0.90
gamma1	0.16

Cuadro 1: Estimación GJR del oro (izquierda) y el SP500 (derecha)

b) Considere los modelos DCC-EWMA y DCC-GARCH para la estimación de las correlaciones entre este par de activos, también bajo el supuesto de Normalidad de las innovaciones.

DCC-GARCH			**DCC-EWMA**
α	β	Valor f. obj.	λ
0.0125319	09799955	5428.357	0.9898682

Cuadro 2: Parámetros estimados Correlación Condicional Dinámica (DCC)

c) ¿Ha tenido un efecto claro la crisis financiera de 2007-2008 en las correlaciones entre estos activos? ¿Y la crisis económica del coronavirus? Razone la respuesta.

   Como podemos ver en la figura 1, hay varios momentos donde la correlación entre ambos activos se vuelve negativa, incluidos los periodos de interés. Esto es debido a que en momentos de crisis el oro actúa como activo refugio para los inversores, por lo que su precio aumenta mientras que un activo como puede ser el índice SP500 cae en esas situaciones, por lo que tiene sentido que la correlación sea negativa.

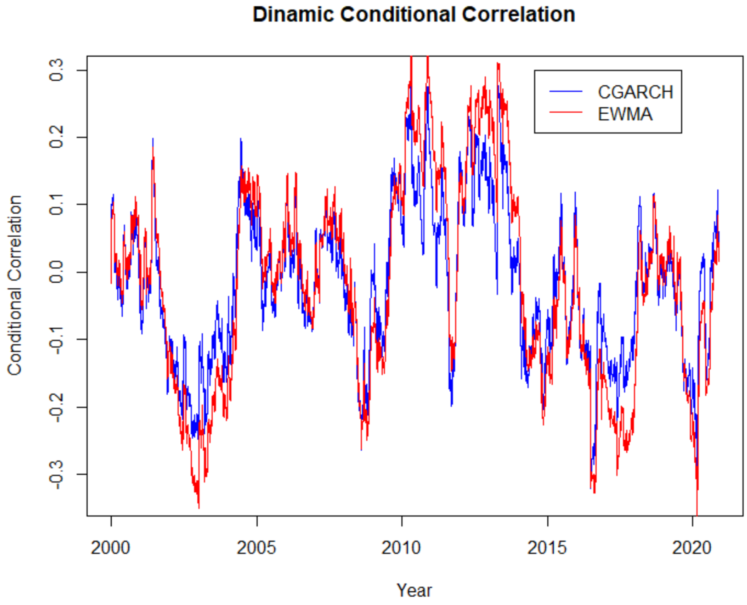

Figura 1: Correlación dinámica entre el SP500 y el oro

d) Calcule el VaR 2.5%, el VaR 1% y el VaR 0.5% a 1 día para toda la muestra utilizando el método paramétrico, con los dos modelos de correlación dinámica condicional, para una cartera en la que invierte un porcentaje igual en cada uno de los dos activos considerados. Presente gráficos temporales de las rentabilidades de la cartera y de los VaR con los dos modelos de correlación dinámica condicional y calcule los tres contrastes que propone Kratz et al. (2018). ¿Qué modelo de DCC es más adecuado para estimar el *tail risk* de las rentabilidades de la cartera? Razone la respuesta.

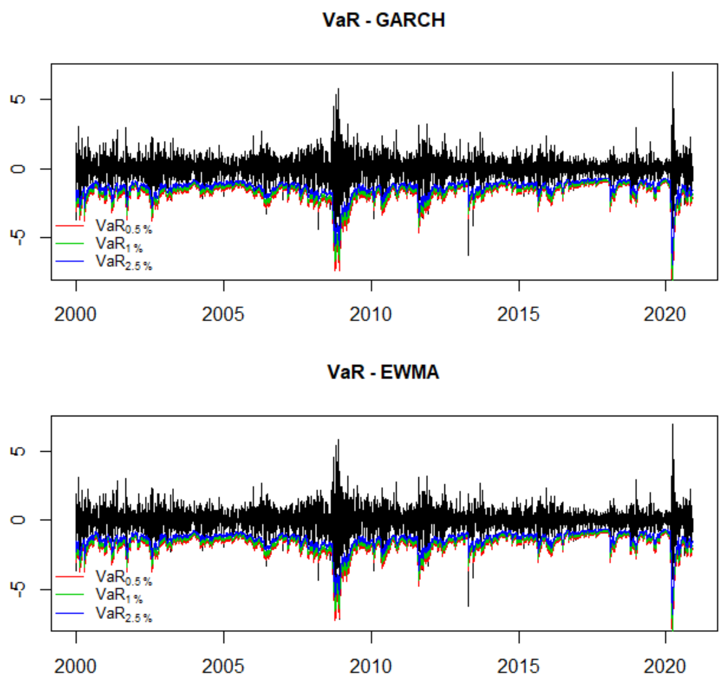

Figura 2: VaR paramétrico de una cartera equiponderada de oro y SP500

Rechazamos todos los contrastes propuestos (Pearson, Nass y de máxima verosimilitud) para hacer backtesting de un VaR multinomial, tanto para el modelo GARCH como para el EWMA, por lo que no podemos determinar cuál de los dos es mejor, tan sólo podemos decir que ninguno de los modelos propuestos es adecuado.

2 Segundo ejercicio

Obtenga los precios diarios del índice de mercado EURO STOXX 50 del periodo 2015-2020, y

a) Realice predicciones del VaR 2.5% y del ES 2.5% a 10 días mediante Simulación Histórica Filtrada (FHS) suponiendo que las rentabilidades tienen esperanza nula, varianza condicional no unitaria que se especifica mediante un GJR-GARCH(1,1) e innovaciones que siguen una distribución Student-t. Para FHS, utilice una ventana móvil de tamaño 1000 para estimar el modelo y genere mediante bootstrapping un amplio número de realizaciones para poder obtener una más precisa distribución empírica de las rentabilidades a 10 días sobre las que calcular el VaR y el ES a 10 días. Describa detalladamente el proceso realizado.

   Para programar el FHS con las condiciones que marcan el enunciado, hemos hecho un bucle que va actualizando el modelo GJR día a día, es decir, estimamos los coeficientes cada día que pasa. Dentro de ese bucle, tenemos otro que nos genera residuos estandarizados mediante bootstrapping, usando los residuos estimados en el modelo GJR, de manera que cada día no sólo tenemos nuevos residuos, sino que el bootstrap también se actualiza de manera aleatoria. Con los residuos obtenidos del bootsrapping y los parámetros del GJR, actualizamos otro bucle donde calculamos los rendimientos a 10 días mediante un modelo GJR,[1] que son sumados y posteriormente se calcula tanto el VaR como el ES a 10 días; tendremos tantos rendimientos a 10 días (simulados) como residuos hayamos simulado mediante bootstrapping (en nuestro caso N = 5000).

b) Presente un gráfico temporal de las rentabilidades, del VaR 2.5% y del ES 2.5% a 10 días, y calcule el contraste de cobertura incondicional de Kupiec para el VaR y los dos contrastes de Acerbi y Szekely para el ES.

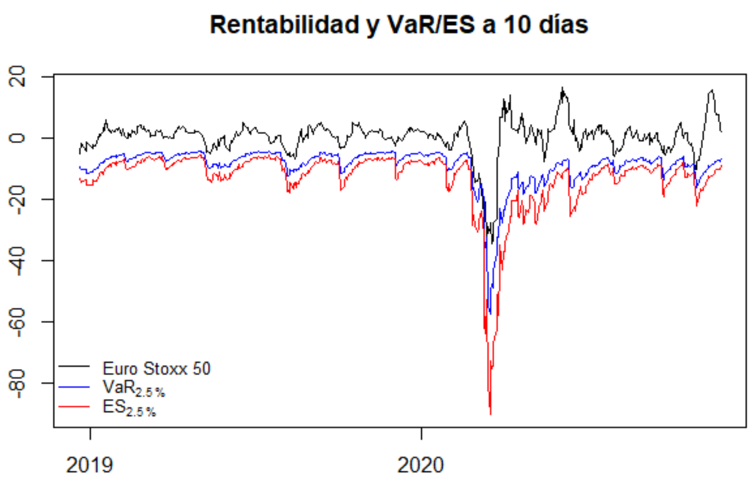

Figura 3: VaR y ES simulados sobre los rendimientos reales.

[1] $\hat{\sigma}_t^2 = \hat{\omega} + [\hat{\alpha} + \hat{\gamma} I(u_{t-i} < 0)]u_{t-1}^2 + \hat{\beta}\sigma_{t-1}^2$ con $u_t = \sigma_t z_t$, siendo z_t los residuos estandarizados.

Aunque a simple vista el ajuste parece bueno, el número de violaciones es muy pequeño (3 para el VaR y 0 para el ES), por lo que acabamos rechazando la hipótesis nula de todos los tests propuestos, algo que no debería ocurrir ya que rechazar el test de A&S para el ES implica una infravaloración del riesgo, cuando lo que ocurre es que lo estamos sobre estimando (si no, tendríamos muchas más violaciones), por lo que puede haber algún error en el código. También es de hacer notar los saltos o escalones que presentan tanto el VaR como el ES (similar a ciclos), algo que ocurre cuando trabajamos con ventanas móviles pero que el FHS debería eliminar.

c) ¿Se han obtenido buenas predicciones del VaR 2.5% y del ES 2.5% a 10 días con FHS? Justifique la respuesta.

   Como hemos comentado en el apartado anterior, no parece que las estimaciones sean buenas, ya que el FHS debería eliminar las *bland tails* pero no lo hace, produciéndose los escalones antes mentados. Haciendo pruebas con horizontes temporales menores ($h = 3$) obtenemos un número de violaciones más razonables (16 para el VaR y 3 para el ES) y de hecho aceptamos todos los tests. De nuevo, esto no debería suceder puesto que el FHS elimina los problemas que surgen al tratar de estimar horizontes relativamente largos, pero es posible que la implementación del código no sea buena.

3 Tercer ejercicio

Simulación de mixtura de Normales

a) Considere dos distribuciones Normales, ambas con esperanza cero, y una con varianza superior a la otra. Simule una mixtura de ambas (a más simulaciones, mayor precisión), con una probabilidad de mezcla más elevada para la distribución de mayor varianza. Observe la asimetría y la curtosis generadas. A continuación, varíe el ratio de varianzas de ambas distribuciones, y repita el ejercicio. Represente gráficamente el modo en que la asimetría y la curtosis varían con el ratio de varianzas. Repita el ejercicio fijando un ratio de varianzas y variando la probabilidad de mezcla.

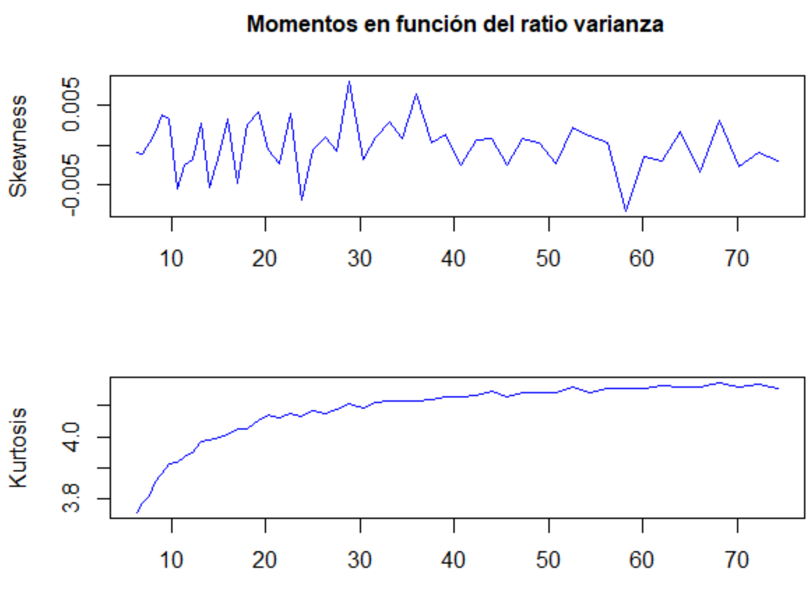

Figura 4: Evolución de la asimetría y la curtosis en función del ratio de varianza (media 0)

En esta figura podemos observar que si cambia el ratio de varianza, la asimetría no se ve afectada mientras que la curtosis aumenta de manera logarítmica.

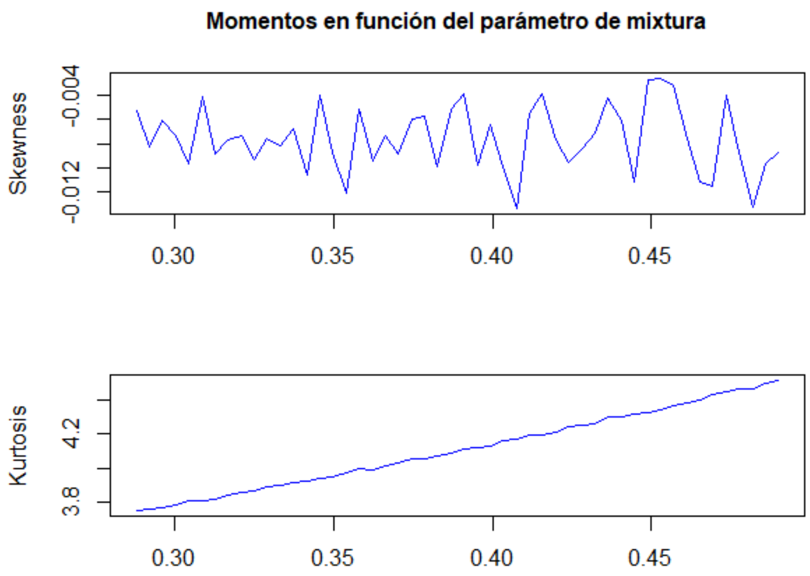

Figura 5: Evolución de la asimetría y la curtosis en función del parámetro de mezcla (media 0)

Cuando cambiamos el parámetro de mixtura (respetando que la proporción de la serie con mayor varianza sea superior, es decir, $max\{p\} = 0.49$), vemos que la asimetría tampoco cambia pero la evolución de la curtosis tiene una evolución lineal (crece más rápidamente).

b) Repita el apartado anterior asignando una esperanza matemática negativa a la distribución de mayor varianza, y una esperanza positiva a la de menor varianza. Comente los resultados obtenidos.

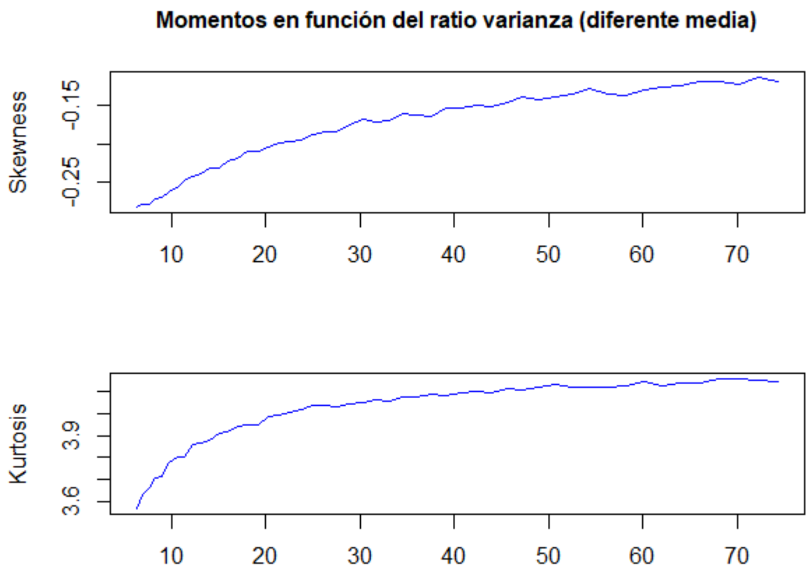

Figura 6: Evolución de la asimetría y la curtosis en función del parámetro de mezcla (media 1 y -1)

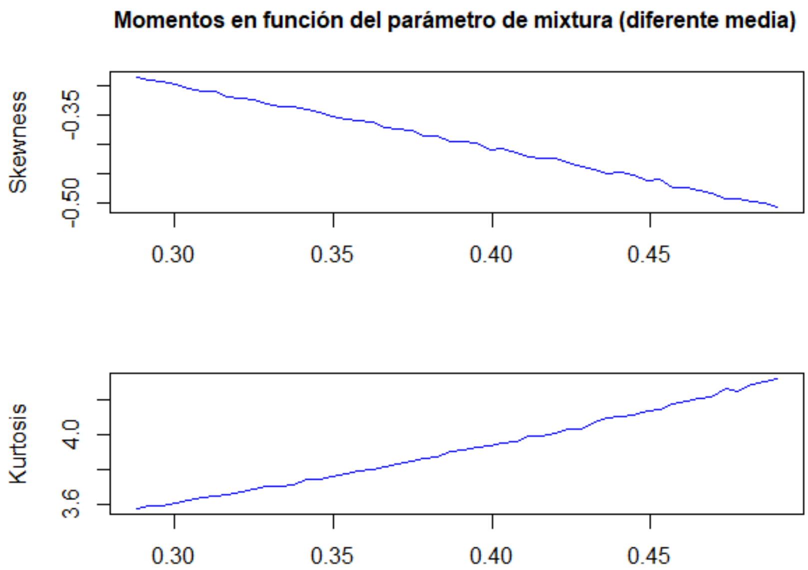

Figura 7: Evolución de la asimetría y la curtosis en función del parámetro de mezcla (media 1 y -1)

Aunque la curtosis mantenga un comportamiento parecido al apartado anterior, la asimetría sí que se ve afectada porque las series tengan una esperanza no nula. En el primer caso (figura 6), al aumentar el ratio de varianza esta disminuye, lo que tiene sentido si definimos la asimetría como $(\mu - \nu)/\sigma$, donde μ es la media, ν es la mediana y σ es la desviación típica; a una mayor desviación típica, tendremos una asimetría más cercana a 0. Este mismo efecto sucede, a la inversa, en el caso donde lo que cambia es p (figura 7). Aquí la desviación típica disminuye, ya que la serie que mayor volatilidad aportaba cada vez es menos relevante.

c) Estime el VaR 5% a horizontes 1, 5 y 20 días a partir de la mixtura utilizada en el apartado anterior. Vuelva a estimar el VaR a partir de una distribución Normal con esperanza y varianza iguales a las de dicha mixtura. Compare los resultados obtenidos.

En los gráficos presentados[2] en el anexo correspondiente, podemos ver que lo que ocurre está directamente relacionado con los comentarios del apartado anterior. En el caso donde el ratio de varianzas cambia, el VaR de la mixtura se hace relativamente más pequeño cuando mayor es el ratio (manteniendo fijo $p = 0.288$). Por otro lado, cuanto menos peso tiene la serie con mayor volatilidad (mayor p), mayor es el VaR de la mixtura respecto al de la normal. En línea con lo que comentábamos antes, al aumentar la asimetría (debido al aumento de p) lo que obtenemos es una cola izquierda más pesada y por lo tanto un VaR superior y viceversa.

Además se puede observar que cuanto mayor es el horizonte al que calculamos el VaR, menos se parecen la mixtura y la normal. En general, cuanto menor sea el horizonte temporal y menor sea el ratio de varianza, mejor ajuste encontramos (el valor de p sólo es relevante para horizontes más alejados de un día).

[2]Para la representación gráfica hemos tenido en cuenta sólo 3 ratios de varianza diferentes y 3 p diferentes de los 50 que hemos calculado: el menor, el mediano y el mayor.

A Figuras

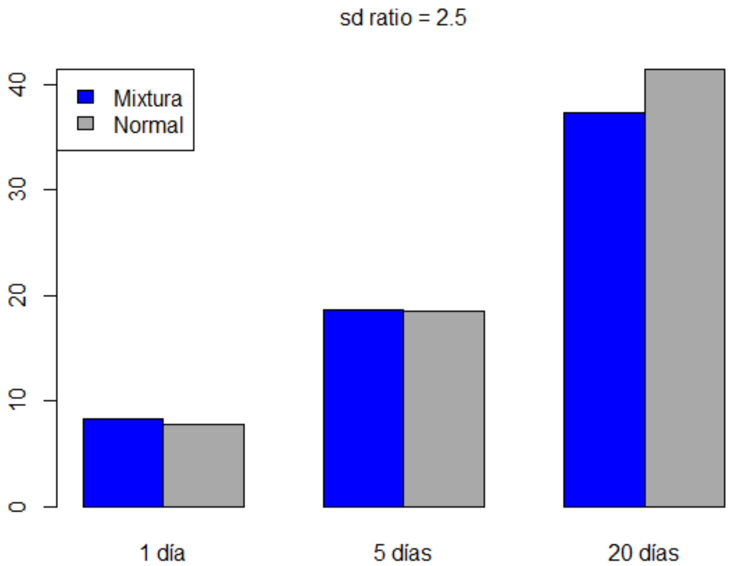

Figura 8: VaR de una mixtura vs VaR normal (sd ratio = 2.5)

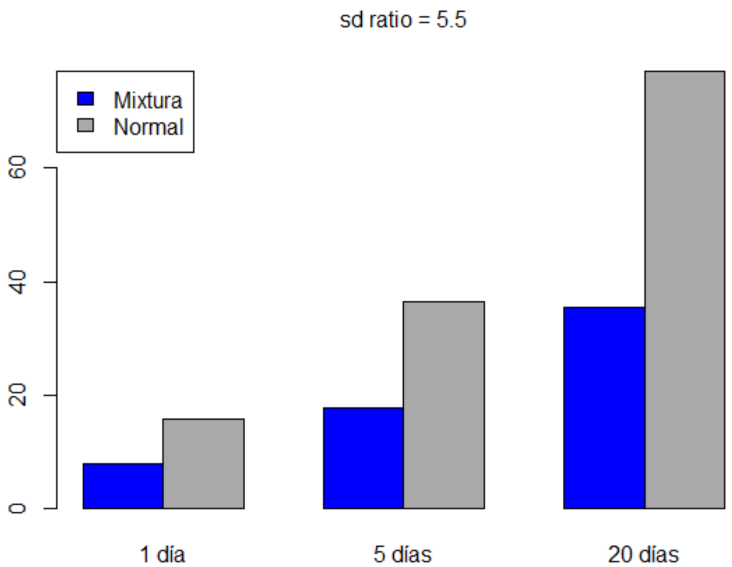

Figura 9: VaR de una mixtura vs VaR normal (sd ratio = 5.5)

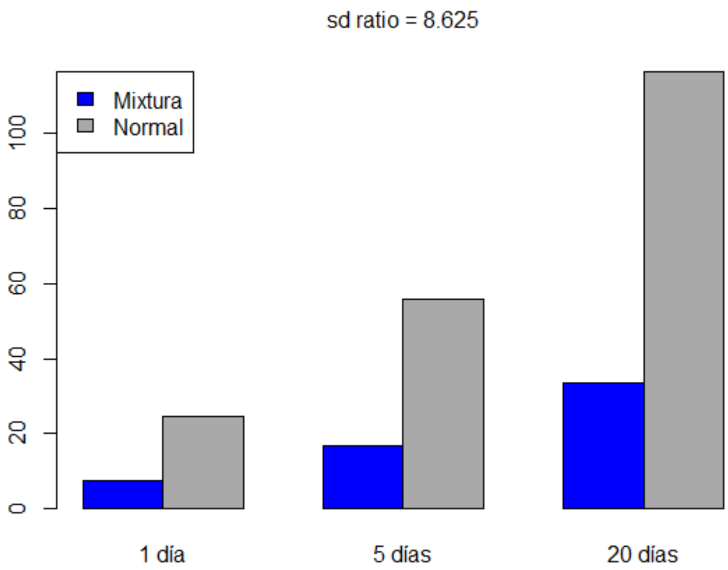

Figura 10: VaR de una mixtura vs VaR normal (sd ratio = 8.625)

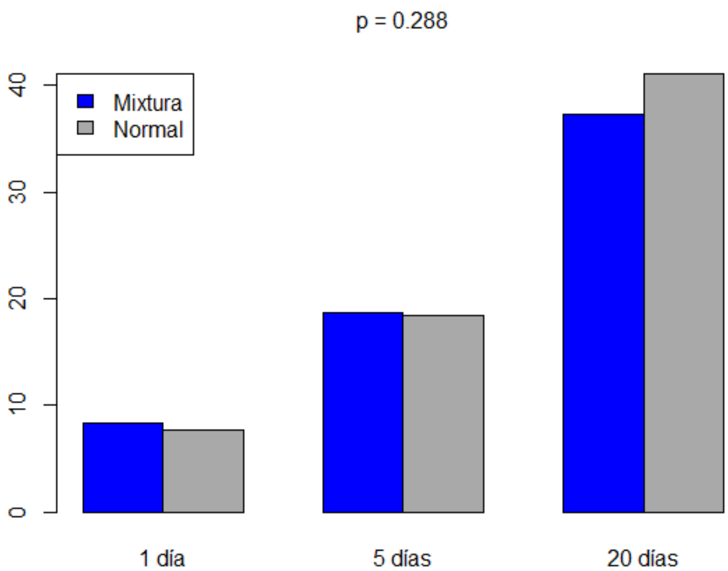

Figura 11: VaR de una mixtura vs VaR normal (p = 0.288)

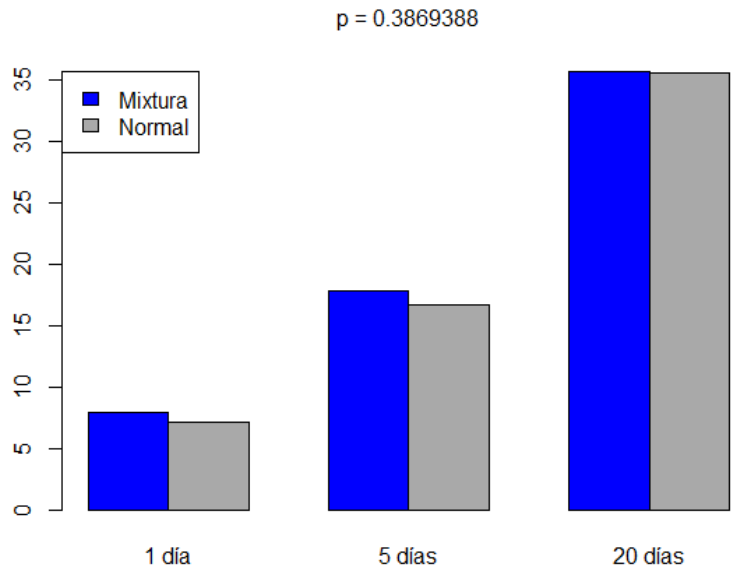

Figura 12: VaR de una mixtura vs VaR normal (p = 0.3869388)

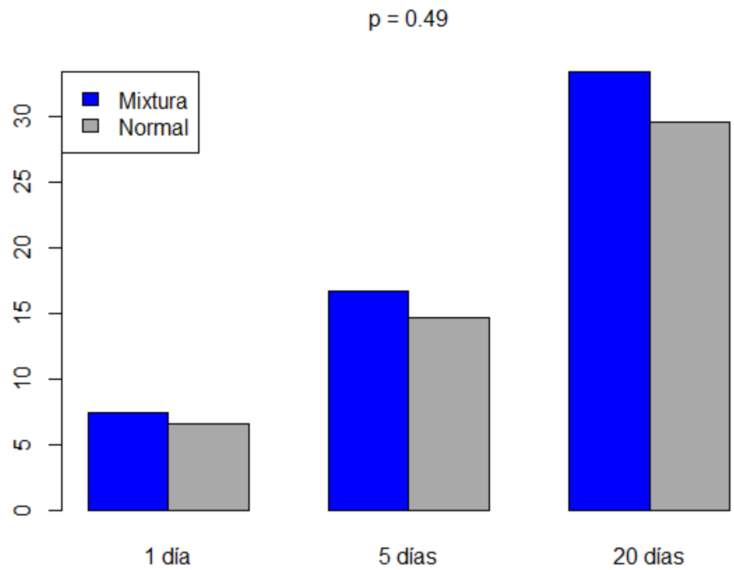

Figura 13: VaR de una mixtura vs VaR normal (p = 0.49)

B Código R

B.1 Ejercicio 1

```r
## Paquetes necesarios

#install.packages("readxl")
#install.packages("rugarch")

library(readxl)
library(rugarch)


#### Importing data ####
data <- read_excel("data.xlsx");
dates <-as.Date(data$Date,format="%Y-%m-%d");

sp500 = data$`S&P 500 COMPOSITE - PRICE INDEX` # SP500 index at close
gold = data$`Gold Bullion LBM $/t oz DELAY`    # Gold - London Bullion
   Market


# Logarithmic returns (percentages)
r_sp <-  100*diff(log(sp500))
r_gold <- 100*diff(log(gold))
nobs <- length(r_sp)


#### GJR model ####
model.spec = ugarchspec(variance.model = list(model='gjrGARCH',
                                              garchOrder = c(1 , 1)),
                        mean.model = list(armaOrder = c(1 , 0)))
# A GJR and AR(1) for the conditional mean. Norm distr by default

# SP500

gjr_sp = ugarchfit(spec=model.spec, data=r_sp, solver='solnp') # Estimation
cond_sigma_sp = gjr_sp@fit$sigma # Conditonal variance
cond_mu_sp = gjr_sp@fit$coef[1]  # Conditonal mean

# Gold

gjr_gold = ugarchfit(spec=model.spec, data=r_gold, solver='solnp') #
   Estimation
cond_sigma_gold = gjr_gold@fit$sigma # Conditonal variance
cond_mu_gold = gjr_gold@fit$coef[1]  # Conditonal mean


#### DCC Models ####

# Standarized returns
Z_sp <- gjr_sp@fit$z
Z_gold <- gjr_gold@fit$z
cross <- Z_sp*Z_gold

# DCC-GARCH

q11_1 <- matrix(0,nobs)
```

```
q12_1 <- matrix(0,nobs)
q22_1 <- matrix(0,nobs)

Q1 <- function(param){
  alpha <- param[1]
  beta <- param[2]
  q11_1[1] = 1
  q12_1[1] = mean(cross)
  q22_1[1] = 1
  for (i in 2:nobs){
    q11_1[i] <- q11_1[1]*(1-alpha-beta)+alpha*Z_sp[i-1]^2+beta*q11_1[i-1]
    q12_1[i] <- q12_1[1]*(1-alpha-beta)+alpha*cross[i-1]+beta*q12_1[i-1]
    q22_1[i] <- q22_1[1]*(1-alpha-beta)+alpha*Z_gold[i-1]^2+beta*q22_1[i-1]
  }
  ccorr1 <- q12_1/sqrt(q11_1*q22_1)
  lnL_1 <- -0.5*log(1-ccorr1^2)-0.5*(Z_sp^2+Z_gold^2-2*ccorr1*cross)/(1-
     ccorr1^2)
  Q1 <- -sum(lnL_1)
}

optim1 <- nlminb(c(0.1,0.7), Q1)
print("Estimación de los parametros alfa y beta")
optim1$par
print("valor función objetivo")
optim1$objective

q11_1[1] = 1
q12_1[1] = mean(cross)
q22_1[1] = 1
alpha <- optim1$par[1]
beta <- optim1$par[2]
for (i in 2:nobs){
  q11_1[i] <- q11_1[1]*(1-alpha-beta)+alpha*Z_sp[i-1]^2+beta*q11_1[i-1]
  q12_1[i] <- q12_1[1]*(1-alpha-beta)+alpha*cross[i-1]+beta*q12_1[i-1]
  q22_1[i] <- q22_1[1]*(1-alpha-beta)+alpha*Z_gold[i-1]^2+beta*q22_1[i-1]
}
ccorr1 <- q12_1/sqrt(q11_1*q22_1)


# DCC-EWMA

q11_2 <- matrix(0,nobs)
q12_2 <- matrix(0,nobs)
q22_2 <- matrix(0,nobs)

Q2 <- function(param){
  lambda <- param[1]
  q11_2[1] = 1
  q12_2[1] = mean(cross)
  q22_2[1]=  1
  for (i in 2:nobs){
    q11_2[i] <- (1-lambda)*Z_sp[i-1]^2+lambda*q11_2[i-1]
    q12_2[i] <- (1-lambda)*cross[i-1]+lambda*q12_2[i-1]
    q22_2[i] <- (1-lambda)*Z_gold[i-1]^2+lambda*q22_2[i-1]
  }
  ccorr2 <- q12_2/sqrt(q11_2*q22_2)
  lnL_2 <- -0.5*log(1-ccorr2^2)-0.5*(Z_sp^2+Z_gold^2-2*ccorr2*cross)/(1-
     ccorr2^2)
```

```
  penalty <- (1+(max(lambda,1.0)-1))^50
  Q2 <- -sum(lnL_2)*penalty
}

optim2 <- optim(c(0.9), Q2)
print("Estimación del parametro lambda")
optim2$par

q11_2[1] = 1
q12_2[1] = mean(cross)
q22_2[1] = 1
lambda <- optim2$par
for (i in 2:nobs){
  q11_2[i] <- (1-lambda)*Z_sp[i-1]^2+lambda*q11_2[i-1]
  q12_2[i] <- (1-lambda)*cross[i-1]+lambda*q12_2[i-1]
  q22_2[i] <- (1-lambda)*Z_gold[i-1]^2+lambda*q22_2[i-1]
}
ccorr2 <- q12_2/sqrt(q11_2*q22_2)

# Dinamic Correlation Plot

plot(ccorr1 ~ dates[-1], type='l',col = "blue",
     main = "Dinamic Conditional Correlation",
     xlab="Year", ylab="Conditional Correlation",) # GARCH
lines(ccorr2 ~ dates[-1], col = "red") # EWMA
legend(dates[3800], 0.3,legend= c("CGARCH", "EWMA"), col=c("blue","red"),
   lty=1)

#### VaR ####

r_portfolio <- 0.5*r_sp + 0.5*r_gold
mu_portfolio <- 0.5*cond_mu_sp + 0.5*cond_mu_gold
taus <- c(.005, .01, .025)

# GARCH
sd_portfolio_garch <- sqrt(0.5^2*cond_sigma_sp^2 + 0.5^2*cond_sigma_gold^2
   +
                          2*0.5*0.5*ccorr1*cond_sigma_sp*cond_sigma_gold
                             )
VaR_GARCH <- matrix(0,nrow=nobs,ncol=length(taus))
b=1
for(a in taus){
  VaR_GARCH[,b] <- mu_portfolio + qnorm(a)*sd_portfolio_garch
  b=b+1}

# EMWA
sd_portfolio_EWMA <- sqrt(0.5^2*cond_sigma_sp^2 + 0.5^2*cond_sigma_gold^2 +
                          2*0.5*0.5*ccorr2*cond_sigma_sp*cond_sigma_gold)
VaR_EWMA <- matrix(0,nrow=nobs,ncol=length(taus))
b=1
for(a in taus){
  VaR_EWMA[,b] <- mu_portfolio + qnorm(a)*sd_portfolio_EWMA
  b=b+1}


# Plotting
par(mfrow=c(2,1),cex.lab=1, cex.axis=1, cex.main=1, mar = c(2, 4, 4, 4))
plot(r_portfolio ~ dates[-1],main="VaR - GARCH",
```

```r
     xlab="", ylab="",type="l",col=1,lwd=1,
     ylim = c(min(VaR_GARCH),max(r_portfolio)))
for (i in 1:length(taus)){
  lines(VaR_GARCH[,i] ~ dates[-1],col=i+1,lwd=1)
}
legend("bottomleft",c(expression("VaR"[0.5~'%']), expression("VaR"[1~'%']),
                      expression("VaR"[2.5~'%'])),
       ncol=1,merge=TRUE, col=2:(length(taus)+1),lty=1, cex=0.8,
       horiz=TRUE, bty = "n")

plot(r_portfolio ~ dates[-1],main="VaR - EWMA",
     xlab="",ylab="",type="l",col=1,lwd=1,
     ylim = c(min(VaR_GARCH),max(r_portfolio)))
par(new=T)
for (i in 1:length(taus)){
  lines(VaR_EWMA[,i] ~ dates[-1], col=i+1,lwd=1)
}
legend("bottomleft",c(expression("VaR"[0.5~'%']), expression("VaR"[1~'%']),
                      expression("VaR"[2.5~'%'])),
       ncol=1,merge=TRUE, col=2:(length(taus)+1),lty=1, cex=0.8,
       horiz=TRUE,bty = "n")

#### Kratz et al. Tests ####

# Pearson - GARCH

O_GARCH <- rep(0, length(taus)+1)
O_GARCH[1] <- sum(r_portfolio > VaR_GARCH[,3])
O_GARCH[2] <- sum(r_portfolio > VaR_GARCH[,2]) - O_GARCH[1]
O_GARCH[3] <- sum(r_portfolio > VaR_GARCH[,1]) - O_GARCH[2] - O_GARCH[1]
O_GARCH[4] <- sum(r_portfolio < VaR_GARCH[,1])

alpha <- rev(c(1 , (1-taus), 0))
Pearson_GARCH <- 0
for (i in length(taus)+1) {
  Sn = (O_GARCH[i] - nobs*(alpha[i+1] - alpha[i]))^2/(nobs*(alpha[i+1] -
      alpha[i]))
  Pearson_GARCH = Pearson_GARCH + Sn
}

if(Pearson_GARCH > qchisq(.95, df=length(O_GARCH))){
  print("Rechazamos H0 para Pearson - GARCH")
} else {
  print("No rechazamos H0 para Pearson - GARCH")
}


# Pearson - EWMA
O_EWMA <- rep(0, length(taus)+1)
O_EWMA[1] <- sum(r_portfolio > VaR_EWMA[,3])
O_EWMA[2] <- sum(r_portfolio > VaR_EWMA[,2]) - O_EWMA[1]
O_EWMA[3] <- sum(r_portfolio > VaR_EWMA[,1]) - O_EWMA[2] - O_EWMA[1]
O_EWMA[4] <- sum(r_portfolio < VaR_EWMA[,1])

Pearson_EWMA<- 0
for (i in length(taus)+1) {
  Sn = (O_EWMA[i] - nobs*(alpha[i+1] - alpha[i]))^2/(nobs*(alpha[i+1] -
      alpha[i]))
```

```
  Pearson_EWMA = Pearson_EWMA + Sn
}

if(Pearson_EWMA > qchisq(.95, df=length(O_EWMA))){
  print("Rechazamos H0 para Pearson - EWMA")
} else {
  print("No rechazamos H0 para Pearson - EWMA")
}

# Nass

N <- length(O_GARCH)

aux <- 0
for (i in N) {
  aux = aux + 1/(alpha[i+1] - alpha[i])
}
c <- 2*N / (2*N - (N^2 + 4*N + 1 + aux)/nobs)
df <- c*N


Nass_GARCH <- c*Pearson_GARCH
Nass_EWMA <- c*Pearson_EWMA

if(Nass_GARCH > qchisq(.95, df=df)){
  print("Rechazamos H0 para Nass - GARCH")
} else {
  print("No rechazamos H0 para Nass - GARCH")
}

if(Nass_EWMA > qchisq(.95, df=df)){
  print("Rechazamos H0 para Nass - EWMA")
} else {
  print("No rechazamos H0 para Nass - EWMA")
}

# LR - GARCH

LR_GARCH <- 0
for (i in length(O_GARCH)){
  if (O_GARCH[i] == 0){
    next
  }
  else{
    Sn_tilde = 2*O_GARCH[i]*log(O_GARCH[i]/(nobs*(alpha[i+1]-alpha[i])))
    LR_GARCH = LR_GARCH + Sn_tilde
  }
}

if(LR_GARCH > qchisq(.95, df=N)){
  print("Rechazamos H0 para LR - GARCH")
} else {
  print("No rechazamos H0 para LR - GARCH")
}

# LR - GARCH

LR_EWMA <- 0
```

```
for (i in length(O_EWMA)){
  if (O_EWMA[i] == 0){
    next
  }
  else{
  Sn_tilde = 2*O_EWMA[i]*log(O_EWMA[i]/(nobs*(alpha[i+1]-alpha[i])))
  LR_EWMA = LR_EWMA + Sn_tilde
  }
}
if(LR_EWMA > qchisq(.95, df=N)){
  print("Rechazamos H0 para LR - EWMA")
} else {
  print("No rechazamos H0 para LR - EWMA")
}
```

B.2 Ejercicio 2

```
## Paquetes necesarios

#install.packages("readxl")
#install.packages("rugarch")

library(readxl)
library(rugarch)


#### Importing data ####
data <- read_excel("datastoxx.xlsx");
dates <-as.Date(data$Date,format="%Y-%m-%d");

set.seed(1585) # For replication (should not differ a lot)
ES50 = data$`Stoxx 50` # EURO STOXX 50 index at close
r_ES50 <-  100*diff(log(ES50)) # Logarithmic returns (percentages)
T <- length(r_ES50)
h <- 10   # Days ahead
W <- 1000 # Window length
N <- 5000 # Simulations
tau <- 0.025 # Significance level

## GJR model specifications
model.spec = ugarchspec(variance.model = list(model = 'gjrGARCH' ,
                                              garchOrder = c(1 , 1)),
                        mean.model=list(armaOrder=c(0,0), include.mean=
                            FALSE),
                        distribution.model = "std")
# A GJR and 0 for the conditional mean. t-Student distribution


#### Initializing loop ####
boot_resid_est <- matrix(0, nrow=N,ncol=h)
r_sim <- matrix(0,nrow=N,ncol = h+1)
sigma_sim <- matrix(0,nrow=N,ncol=h+1)
resid_sim <- matrix(0,nrow=N,ncol=h+1)
r_h <- matrix(0,nrow=N,ncol=(T-W)) # Simulated returns h days ahead
VaR_h <- rep(0, (T-W))
ES_h <- rep(0, (T-W))

for (k in 1:(T-W)) {
  # GJR estimation
```

```
  model <- ugarchfit(spec=model.spec,data=r_ES50[k:(W+k-1)], solver = '
      solnp')
  cond_sigma <-  model@fit$sigma # Conditonal sd
  resid <- model@fit$residuals # Residuals
  resid_est <- model@fit$z # Standardized innovations
  omega <- model@fit$coef[1] # GARCH(1,1) parameters
  alpha <- model@fit$coef[2]
  beta <- model@fit$coef[3]
  gamma <- model@fit$coef[4]

  # Bootstrapping
  uni <- matrix(floor(W*runif(N*h) + 1),N,h)
  for (j in 1:h) {
    for (i in 1:N) {
      boot_resid_est[i,j] <- resid_est[uni[i,j]]
    }
  }

  ## Simulation
  # Initial values
  r_sim[,1] <- r_ES50[W+k-1]
  sigma_sim[,1] <- cond_sigma[W]
  resid_sim[,1] <- resid[W]
  # Loop
  for (i in 1:N) {
    for (j in 1:h) {
        if (resid_sim[i,j] <= 0){
          sigma_sim[i,j+1] = sqrt(omega + (alpha + gamma)*resid_sim[i,j]^2
              +
                                    beta*sigma_sim[i,j]^2)
        } else {
          sigma_sim[i,j+1] = sqrt(omega + alpha*resid_sim[i,j]^2 +
                                    beta*sigma_sim[i,j]^2)
        }
        resid_sim[i,j+1] <- sigma_sim[i,j]*boot_resid_est[i,j]
      }
  }
  r_sim <- resid_sim[,-1]
  r_h[,k] <- rowSums(r_sim)

  ## VaR and ES
  VaR_h[k] <- quantile(r_h[,k], probs=tau)
  ES_h[k] <- sum(r_h[r_h[,k]<VaR_h[k],k])/(N*tau)
}

## Returns in h days
r_ES50_h <- rep(0,length(VaR_h))
for (i in 1:length(VaR_h)) {
  r_ES50_h[i] <- sum(r_ES50[(W-h+i):(W+i)])
}


#### Plotting ####

plot(r_ES50_h ~ dates[(W+2):length(dates)],
     xlab="", ylab="",type='l',col = "black",
     ylim = c(min(ES_h),max(r_ES50_h)),
     main = "Rentabilidad y VaR/ES a 10 días")
```

```r
lines(VaR_h ~ dates[(W+2):length(dates)], col = "blue")
lines(ES_h ~ dates[(W+2):length(dates)], col = "red")
legend("bottomleft",c(expression("Euro Stoxx 50"),expression("VaR"[2.5~'%'
   ]),
                     expression("ES"[2.5~'%'])), ncol=1,merge=TRUE,
       col=c("black","blue","red"),lty=1, cex=0.8, bty = "n")

#### Backtesting ####
# Unconditional Coverage Test (Kupiec) for VaR
T1 <- sum(r_ES50_h < VaR_h) # Number of violations
T0 <- length(r_ES50_h) - T1
FR <- T1/length(r_ES50_h) # Failure rate

Kupiec <- -2*log(((1-tau)^T0*tau^T1)/((1-FR)^T0*FR^T1))
crit_val_kupiec <- qchisq(1 - tau, df=1)

if (Kupiec > crit_val_kupiec){
print("Rechazamos hipótesis nula en el test de Kupiec")      # tau != FR
} else {
  print("No rechazamos hipótesis nula en el test de Kupiec") # tau = FR
}

#### Acerbi & Szekely Test for ES (2 statistics) ####


sum <- sum(ifelse(r_ES50_h<VaR_h,1,0)*r_ES50_h/ES_h)
Z1 <- sum/T1 - 1
Z2 <- 1 - (1-Z1)*T1/(length(r_ES50_h)*tau)


sum_sim <- rep(0,N)
Z1_sim <- sum_sim
Z2_sim <- sum_sim
for (i in 1:N) {
  sum_sim[i] <- sum(ifelse(r_h[i,]<VaR_h,1,0)*r_h[i,]/ES_h)
  Z1_sim[i] <- sum_sim[i]/T1 - 1
  Z2_sim[i] <- 1 - (1-Z1_sim[i])*T1/(length(r_h[1,])*tau)
}

pvalue_Z1 <- sum(Z1_sim< Z1)/N
pvalue_Z2 <- sum(Z2_sim< Z2)/N

if (pvalue_Z1 < tau){
  print("Rechazamos hipótesis nula para Z1")     # ES_sim = ES
} else {
  print("No rechazamos hipótesis nula para Z1")  # ES_sim \leq  ES
}

if (pvalue_Z2 < tau){
  print("Rechazamos hipótesis nula para Z2")     # ES_sim = ES
} else {
  print("No rechazamos hipótesis nula para Z2")  # ES_sim \leq  ES
}
```

B.3 Ejercicio 3

```r
## Paquetes necesarios

#install.packages("moments")
```

```r
library(moments)

set.seed(1585) # For replication
T <- 1.0e6 # Number of simulations
p_ini <- 0.288 # Mix parameter.
mean1 <-  0.0
mean2 <-  0.0
sd1_ini <-  2
sd2_ini <-  5
n <- 50 # Number of variances/parameters

#### Moving var ratio ####
sd1 <- rep(sd1_ini, n)
sd2 <- seq(from=sd2_ini, by=0.25, length.out=n)
sd_ratio <-  sd2/sd1
y1 <- matrix(0,nrow=length(sd1), ncol=T)
y2 <- y1
for (i in 1:length(sd1)) {
  y1[i,] <- rnorm(T, mean=mean1, sd=sd1[i])
  y2[i,] <- rnorm(T, mean=mean2, sd=sd2[i])
}

M1 <- matrix(0,nrow=length(sd1), ncol=T) # Mixture
skew1 <- rep(0,length(sd1))
kurt1 <- skew1
for (i in 1:length(sd1)) {
  U <- runif(T)  # Uniform for mixing
  for (j in 1:T) {
    if (U[j] <= p_ini){
      M1[i,j] <- y1[i,j]
    } else {
      M1[i,j] <- y2[i,j]
    }
  }
  skew1[i] <- skewness(M1[i,])
  kurt1[i] <- kurtosis(M1[i,])
}

## Ploting
var_ratio <- sd_ratio^2
par(mfrow=c(2,1),cex.lab=1, cex.axis=1, cex.main=1, mar = c(2, 4, 4, 4))
plot(skew1 ~ var_ratio, type='l',col = "blue",
     main="Momentos en función del ratio varianza", xlab="", ylab="Skewness
        ")
plot(kurt1 ~ var_ratio, type='l',col = "blue",
     main = "", xlab = "", ylab = "Kurtosis")


#### Moving mix parameter ####
p <- seq(from=p_ini, to=0.49, length.out=n)
x1 <- rnorm(T, mean=mean1, sd=sd1_ini)
x2 <- rnorm(T, mean=mean2, sd=sd2_ini)
M2 <- matrix(0,nrow=length(p), ncol=T) # Mixture
skew2 <- rep(0,length(p))
kurt2 <- skew2
for (i in 1:length(p)) {
  U <- runif(T)  # Uniform for mixing
```

```
  for (j in 1:T) {
    if (U[j] <= p[i]){
      M2[i,j] <- x1[j]
    } else {
      M2[i,j] <- x2[j]
    }
  }
  skew2[i] <- skewness(M2[i,])
  kurt2[i] <- kurtosis(M2[i,])
}

## Ploting

par(mfrow=c(2,1),cex.lab=1, cex.axis=1, cex.main=1, mar = c(2, 4, 4, 4))
plot(skew2 ~ p, type='l',col = "blue",
     main = "Momentos en función del parámetro de mixtura",
     xlab="", ylab="Skewness")
plot(kurt2 ~ p, type='l',col = "blue", main = "", xlab = "", ylab = "
    Kurtosis")


## Same, with different means

mean1_2 <-  1.0
mean2_2 <-  -1.0

#### Moving var ratio (different means) ####
y1_2 <- matrix(0,nrow=length(sd1), ncol=T)
y2_2 <- y1_2
for (i in 1:length(sd1)) {
  y1_2[i,] <- rnorm(T, mean=mean1_2, sd=sd1[i])
  y2_2[i,] <- rnorm(T, mean=mean2_2, sd=sd2[i])
}

M1_2 <- matrix(0,nrow=length(sd1), ncol=T) # Mixture
skew1_2 <- rep(0,length(sd1))
kurt1_2 <- skew1_2
for (i in 1:length(sd1)) {
  U <- runif(T)  # Uniform for mixing
  for (j in 1:T) {
    if (U[j] <= p_ini){
      M1_2[i,j] <- y1_2[i,j]
    } else {
      M1_2[i,j] <- y2_2[i,j]
    }
  }
  skew1_2[i] <- skewness(M1_2[i,])
  kurt1_2[i] <- kurtosis(M1_2[i,])
}

# Ploting
par(mfrow=c(2,1),cex.lab=1, cex.axis=1, cex.main=1, mar = c(2, 4, 4, 4))
plot(skew1_2 ~ var_ratio, type='l',col = "blue",
     main = "Momentos en función del ratio varianza (diferente media)",
     xlab = "", ylab = "Skewness")
plot(kurt1_2 ~ var_ratio, type='l',col="blue",main="",xlab="", ylab="
    Kurtosis")
```

```
#### Moving mix parameter (different means) ####
x1_2 <- rnorm(T, mean=mean1_2, sd=sd1_ini)
x2_2 <- rnorm(T, mean=mean2_2, sd=sd2_ini)
M2_2 <- matrix(0,nrow=length(p), ncol=T) # Mixture
skew2_2 <- rep(0,length(p))
kurt2_2 <- skew2_2
for (i in 1:length(p)) {
  U <- runif(T)  # Uniform for mixing
  for (j in 1:T) {
    if (U[j] <= p[i]){
      M2_2[i,j] <- x1_2[j]
    } else {
      M2_2[i,j] <- x2_2[j]
    }
  }
  skew2_2[i] <- skewness(M2_2[i,])
  kurt2_2[i] <- kurtosis(M2_2[i,])
}

# Ploting
par(mfrow=c(2,1),cex.lab=1, cex.axis=1, cex.main=1, mar = c(2, 4, 4, 4))
plot(skew2_2 ~ p, type='l',col = "blue",
     main = "Momentos en función del parámetro de mixtura (diferente media)
        ",
     xlab = "", ylab = "Skewness")
plot(kurt2_2 ~ p, type='l',col="blue", main="", xlab="", ylab="Kurtosis")


#### VaR ####

tau <- 0.05 # Significance level
h <- c(1,5,20) # Days ahead

## Moving var ratio (different means)

var_ratio_index <- c(1,round(n/2),n)
sd_M_1 <- rep(0,length(var_ratio_index))
mean_M_1 <- rep(0,length(var_ratio_index))
VaR_M_1 <- matrix(0,nrow = length(h),ncol = length(var_ratio_index))
VaR_N_1 <- matrix(0,nrow = length(h),ncol = length(var_ratio_index))

for (i in 1:length(var_ratio_index)) {
  sd_M_1[i] <- sd(M1_2[var_ratio_index[i],])
  mean_M_1[i] <- mean(M1_2[var_ratio_index[i],])
  for (j in 1:length(h)) {
    VaR_M_1[i,j] <- -sqrt(h[j])*quantile(M2_2[var_ratio_index[i],],tau)
    VaR_N_1[i,j] <- -sd_M_1[i]*sqrt(h[j])*qnorm(tau) - h[j]*mean_M_1[i]
  }
}

# Ploting
VaR1_1 <- matrix(c(VaR_M_1[1,],VaR_N_1[1,]),ncol=3,byrow=TRUE)
colnames(VaR1_1) <- c("1 día", "5 días", "20 días")
rownames(VaR1_1) <- c("Mixtura", "Normal")
VaR1_1 <- as.table(VaR1_1)

VaR2_1 <- matrix(c(VaR_M_1[2,],VaR_N_1[2,]),ncol=3,byrow=TRUE)
```

```r
colnames(VaR2_1) <- c("1 día", "5 días", "20 días")
rownames(VaR2_1) <- c("Mixtura", "Normal")
VaR2_1 <- as.table(VaR2_1)

VaR3_1 <- matrix(c(VaR_M_1[3,],VaR_N_1[3,]),ncol=3,byrow=TRUE)
colnames(VaR3_1) <- c("1 día", "5 días", "20 días")
rownames(VaR3_1) <- c("Mixtura", "Normal")
VaR3_1 <- as.table(VaR3_1)


par(mfrow=c(1,1),cex.lab=1, cex.axis=1, cex.main=1, mar = c(2, 4, 4, 4))
barplot(VaR1_1,
        main = expression("sd ratio = 2.5"),
        xlab = "Días",
        col = c("blue","darkgrey"),
        beside = TRUE
)
legend("topleft",
       c("Mixtura","Normal"),
       fill = c("blue","darkgrey")
)
barplot(VaR2_1,
        main = expression("sd ratio = 5.5"),
        xlab = "Días",
        col = c("blue","darkgrey"),
        beside = TRUE
)
legend("topleft",
       c("Mixtura","Normal"),
       fill = c("blue","darkgrey")
)
barplot(VaR3_1,
        main = expression("sd ratio = 8.625"),
        xlab = "Días",
        col = c("blue","darkgrey"),
        beside = TRUE
)
legend("topleft",
       c("Mixtura","Normal"),
       fill = c("blue","darkgrey")
)

## Moving mix parameter (different means)

p_index <- c(1,round(n/2),n)
sd_M_2 <- rep(0,length(p_index))
mean_M_2 <- rep(0,length(p_index))
VaR_M_2 <- matrix(0,nrow = length(h),ncol = length(p_index))
VaR_N_2 <- matrix(0,nrow = length(h),ncol = length(p_index))

for (i in 1:length(p_index)) {
  sd_M_2[i] <- sd(M2_2[p_index[i],])
  mean_M_2[i] <- mean(M2_2[p_index[i],])
  for (j in 1:length(h)) {
    VaR_M_2[i,j] <- -sqrt(h[j])*quantile(M2_2[p_index[i],],tau)
    VaR_N_2[i,j] <- -sd_M_2[i]*sqrt(h[j])*qnorm(tau) - h[j]*mean_M_2[i]
  }
}
```

```r
# Ploting
VaR1_2 <- matrix(c(VaR_M_2[1,],VaR_N_2[1,]),ncol=3,byrow=TRUE)
colnames(VaR1_2) <- c("1 día", "5 días", "20 días")
rownames(VaR1_2) <- c("Mixtura", "Normal")
VaR1_2 <- as.table(VaR1_2)

VaR2_2 <- matrix(c(VaR_M_2[2,],VaR_N_2[2,]),ncol=3,byrow=TRUE)
colnames(VaR2_2) <- c("1 día", "5 días", "20 días")
rownames(VaR2_2) <- c("Mixtura", "Normal")
VaR2_2 <- as.table(VaR2_2)

VaR3_2 <- matrix(c(VaR_M_2[3,],VaR_N_2[3,]),ncol=3,byrow=TRUE)
colnames(VaR3_2) <- c("1 día", "5 días", "20 días")
rownames(VaR3_2) <- c("Mixtura", "Normal")
VaR3_2 <- as.table(VaR3_2)


barplot(VaR1_2,
        main = expression("p = 0.288"),
        xlab = "Días",
        col = c("blue","darkgrey"),
        beside = TRUE
)
legend("topleft",
       c("Mixtura","Normal"),
       fill = c("blue","darkgrey")
)
barplot(VaR2_2,
        main = expression("p = 0.3869388"),
        xlab = "Días",
        col = c("blue","darkgrey"),
        beside = TRUE
)
legend("topleft",
       c("Mixtura","Normal"),
       fill = c("blue","darkgrey")
)
barplot(VaR3_2,
        main = expression("p = 0.49"),
        xlab = "Días",
        col = c("blue","darkgrey"),
        beside = TRUE
)
legend("topleft",
       c("Mixtura","Normal"),
       fill = c("blue","darkgrey")
)
```