Obligatorio Series Temporales

Tema: Analisis de la evolucion del precio del oro

Fecha: 08/07/2021

Autores: Adrian Arredondo y Alan Rytt

In [1]:

```
library(astsa)
options(repr.plot.width=15, repr.plot.height=8) #ajusta tamaño de graficas
library(dplyr)
#install.packages("xts")                          # Install & load xts package
library("xts")
library(lubridate)
library(forecast)
#install.packages("fGarch")
library(fGarch)
#install.packages("rugarch")
library(rugarch)
library(tseries)
#install.packages("fDMA")
library(fDMA)
#install.packages("dynlm")
library(dynlm)
#install.packages("FinTS")
library(FinTS)
#install.packages("magick")
library(magick)
```

```
Attaching package: 'dplyr'


The following objects are masked from 'package:stats':

    filter, lag


The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union


Warning message:
"package 'xts' was built under R version 3.6.3"
Loading required package: zoo

Warning message:
"package 'zoo' was built under R version 3.6.3"

Attaching package: 'zoo'


The following objects are masked from 'package:base':

    as.Date, as.Date.numeric



Attaching package: 'xts'


The following objects are masked from 'package:dplyr':

    first, last



Attaching package: 'lubridate'


The following object is masked from 'package:base':

    date


Warning message:
"package 'forecast' was built under R version 3.6.3"
Registered S3 method overwritten by 'quantmod':
  method            from
  as.zoo.data.frame zoo


Attaching package: 'forecast'


The following object is masked from 'package:astsa':

    gas
```

```
Warning message:
"package 'fGarch' was built under R version 3.6.3"
Loading required package: timeDate

Loading required package: timeSeries

Warning message:
"package 'timeSeries' was built under R version 3.6.3"

Attaching package: 'timeSeries'


The following object is masked from 'package:zoo':

    time<-


Loading required package: fBasics

Warning message:
"package 'fBasics' was built under R version 3.6.3"

Attaching package: 'fBasics'


The following object is masked from 'package:astsa':

    nyse


Warning message:
"package 'rugarch' was built under R version 3.6.3"
Loading required package: parallel


Attaching package: 'rugarch'


The following object is masked from 'package:stats':

    sigma


Warning message:
"package 'tseries' was built under R version 3.6.3"
Warning message:
"package 'fDMA' was built under R version 3.6.3"
Warning message:
"package 'dynlm' was built under R version 3.6.3"
Warning message:
"package 'FinTS' was built under R version 3.6.3"

Attaching package: 'FinTS'


The following object is masked from 'package:forecast':

    Acf


Warning message:
```

```
"package 'magick' was built under R version 3.6.3"
Linking to ImageMagick 6.9.12.3
Enabled features: cairo, freetype, fftw, ghostscript, heic, lcms, pango, r
aw, rsvg, webp
Disabled features: fontconfig, x11
```

Ingesta de datos

In [2]:

```
#Fuente de los datos: Kaggle
df_gold = read.csv("gold_price_data.csv")
head(df_gold,15)
tail(df_gold)
```

A data.frame: 15 × 2

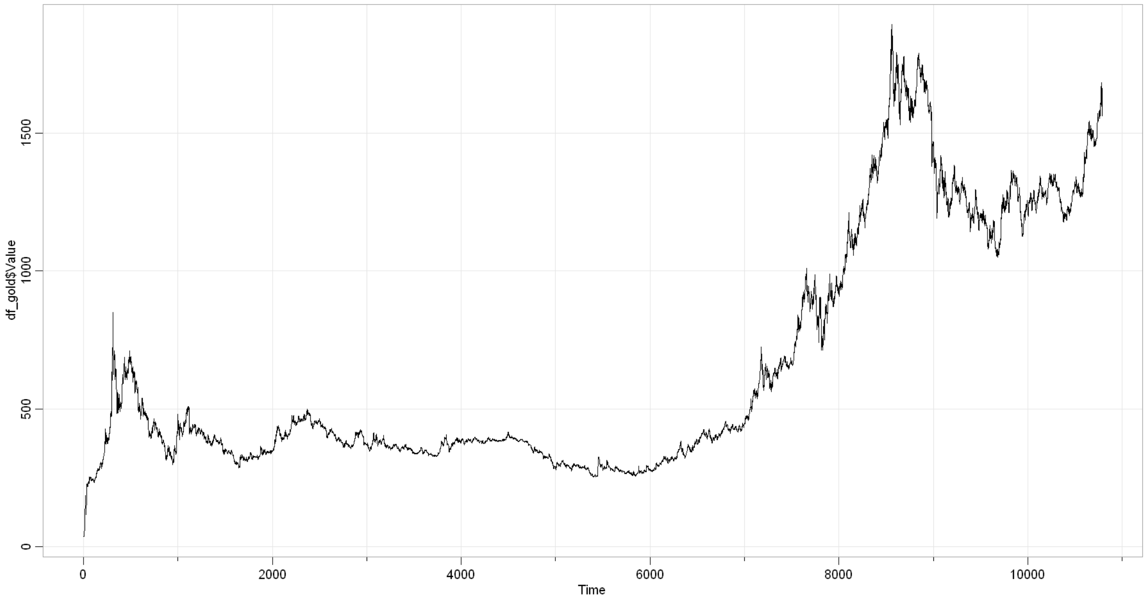
	Date	Value
	<fct>	<dbl>
1	1970-01-01	35.2
2	1970-04-01	35.1
3	1970-07-01	35.4
4	1970-10-01	36.2
5	1971-01-01	37.4
6	1971-04-01	38.9
7	1971-07-01	40.1
8	1971-10-01	42.0
9	1972-01-03	43.5
10	1972-04-03	48.3
11	1972-07-03	62.1
12	1972-10-02	65.5
13	1973-01-01	63.9
14	1973-04-02	84.4
15	1973-07-02	120.1

A data.frame: 6 × 2

	Date	Value
	<fct>	<dbl>
10782	2020-03-06	1683.65
10783	2020-03-09	1672.50
10784	2020-03-10	1655.70
10785	2020-03-11	1653.75
10786	2020-03-12	1570.70
10787	2020-03-13	1562.80

In [3]:

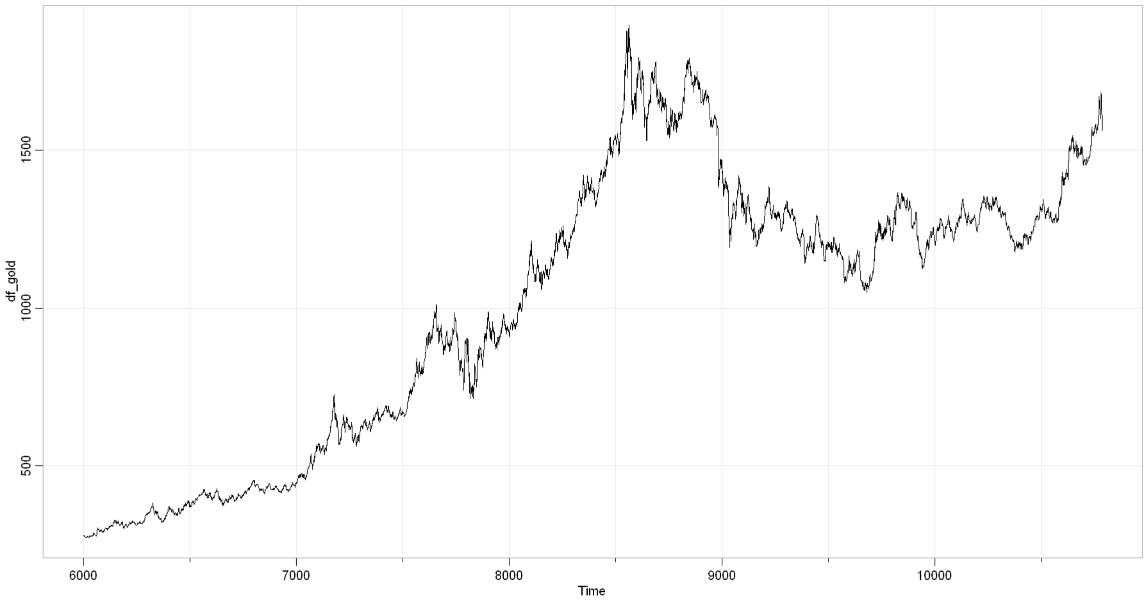
```
tsplot(df_gold$Value)
```



Observamos que la gráfica tiene dos tendencias muy diferentes. Analizando los datos encontramos que al principio de la serie los datos son trimestrales y que posteriormente son diarios. Es por esto por lo que optamos por tomar la serie del 6000 en adelante.

In [4]:

```
df_gold = ts(df_gold$Value[6000:length(df_gold$Value)],start = 6000)
tsplot(df_gold)
```

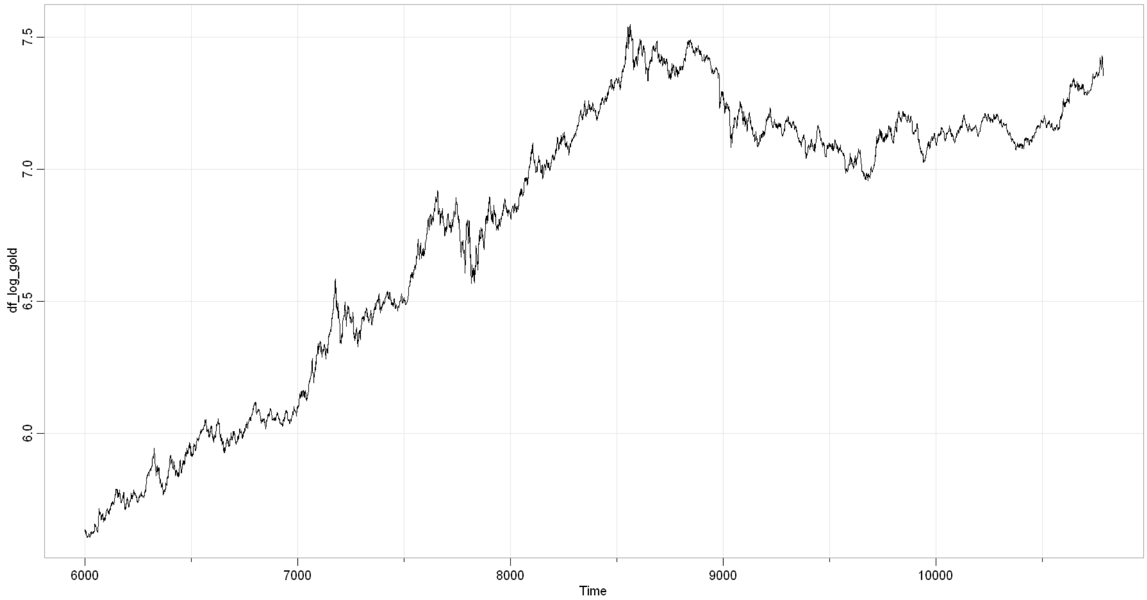


Transformación de los datos

Para minimizar el impacto de la tendencia exponencial le aplicamos logaritmo a la serie.

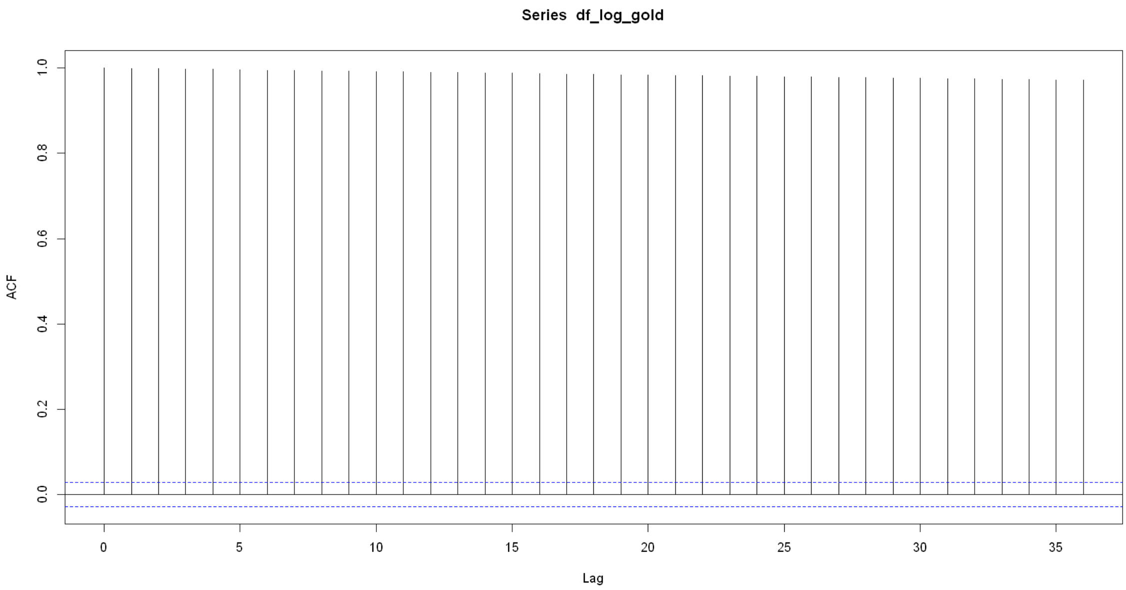
In [5]:

```
df_log_gold = log(df_gold)
tsplot(df_log_gold)
```



In [6]:

```
acf(df_log_gold)
```



A simple vista la serie no se muestra estacionaria. El ACF nos demuestra lo mismo ya que nunca llega a culminar un ciclo de estacionariedad.

Debido a que tenemos que dejar la serie estacional se nos plantean dos mecanismos:

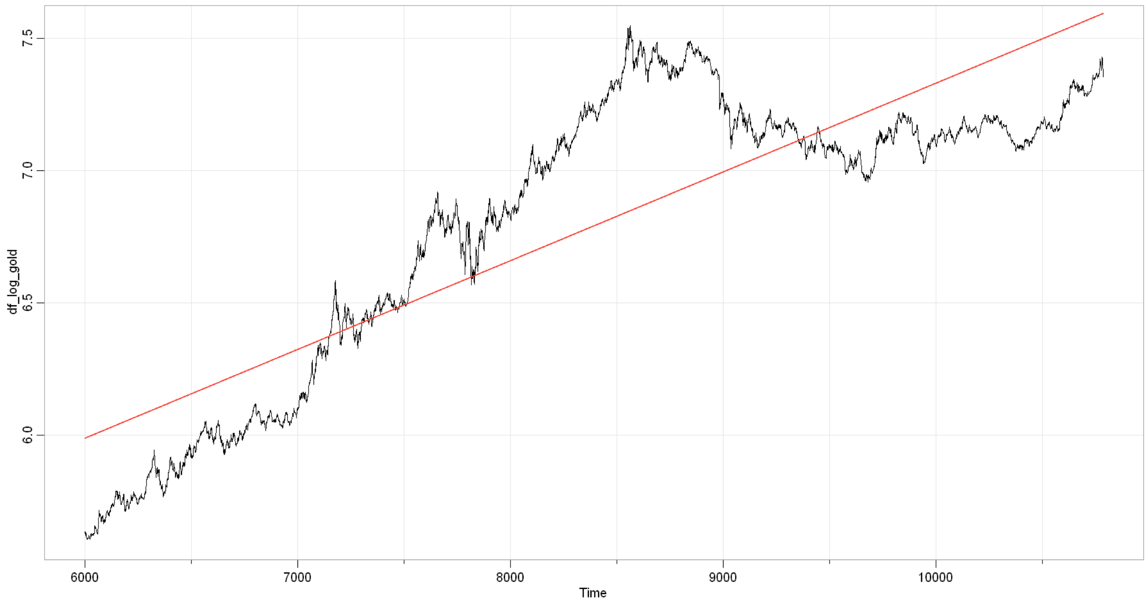
1) Ajustar una recta en el tiempo

2) Hacer la diferencia de los logaritmos entre un día y el otro

Ajustar una recta no permite estacionar la serie debido a que no tiene una tendencia lineal al alza en el tiempo. En otras palabras no podemos confirmar que la tendencia sea un componente fijo, sino que parecería ser variable día a día.

In [7]:

```
t = time(df_log_gold)
fit <- lm(df_log_gold ~ t)


predictions = ts(fitted(fit),start=6000)
tsplot(df_log_gold)
lines(predictions, col=2, lwd=2)
```



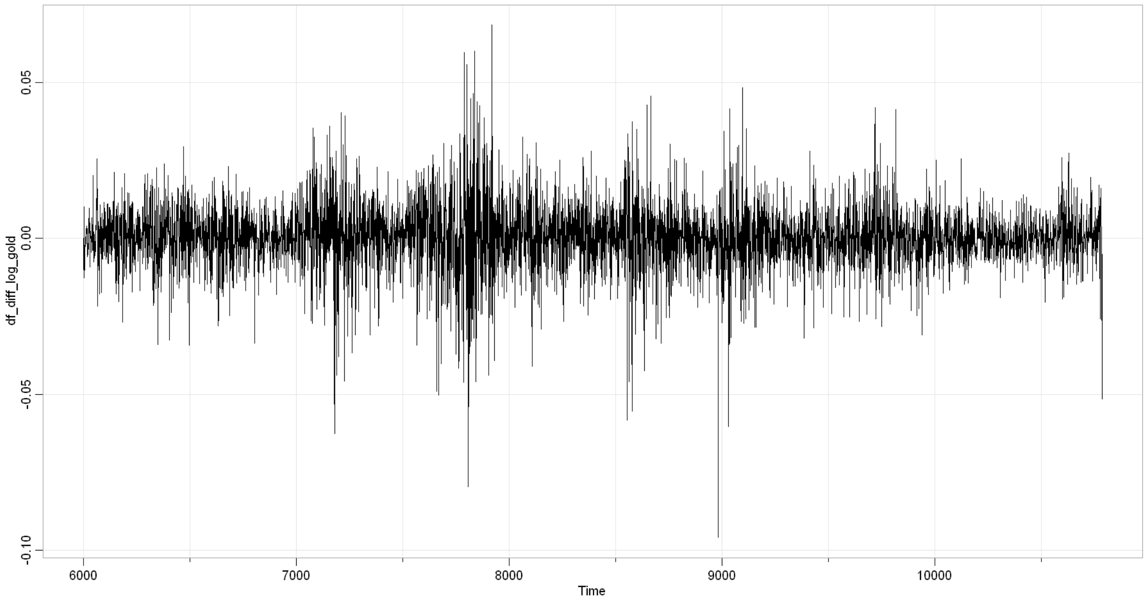
Como crece día a día irregularmente aplicamos la diferencia de logaritmos para ver el retorno o tasa de crecimiento diaria.

Este mecanismo permite centrar la serie en valores estacionarios cercanos a cero.

In [8]:

```
df_diff_log_gold = diff(log(df_gold))
tsplot(df_diff_log_gold)
mean(df_diff_log_gold)
```

0.000358966765199535



Observamos que la señal continúa teniendo picos marcados (alta varianza), por lo tanto entendemos que la serie tiene cierta estacionalidad en la media pero con una alta varianza.

La alta varianza es sinónimo de **heterocedasticidad**. En los modelos de regresión lineal se dice que hay heterocedasticidad cuando la varianza de los errores no es igual en todas las observaciones. Así, no se cumple uno de los requisitos básicos de las hipótesis de los modelos lineales.

Separo en Train y Test

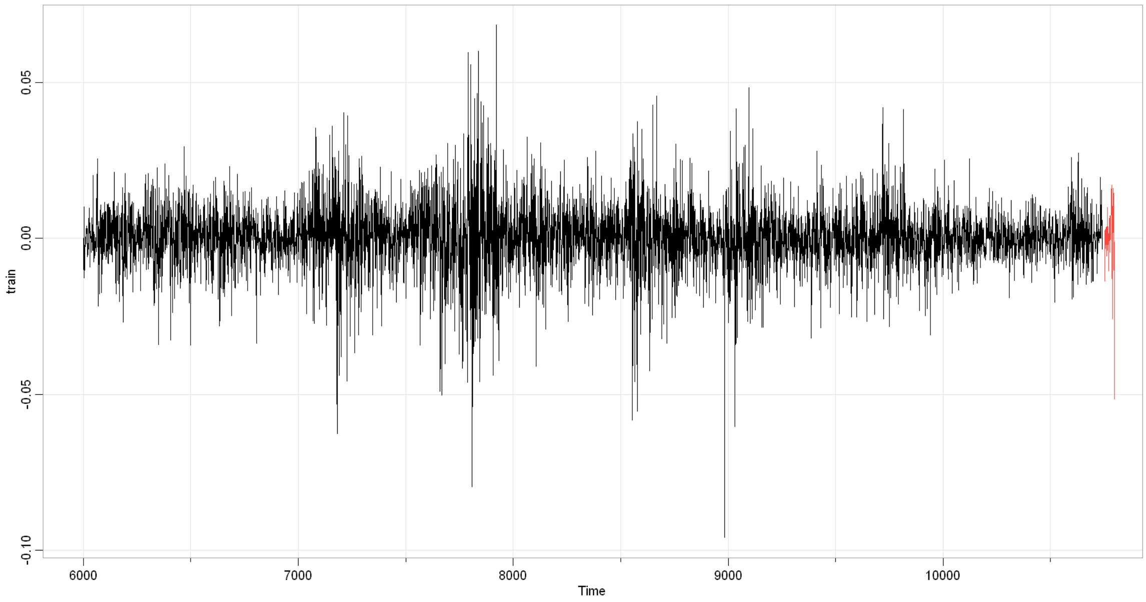
In [9]:

```
train = ts(df_diff_log_gold[0:4740],start = 6001)
test = ts(df_diff_log_gold[4740:length(df_diff_log_gold)],start = 10751)
```

In [10]:

```
tsplot(train)
lines(test,col=2) #Rojo
mean(train)
```

0.00036375774423983



In [11]:

```
#Aplicamos test para confirmar que es estacionario
adf.test(train)
```

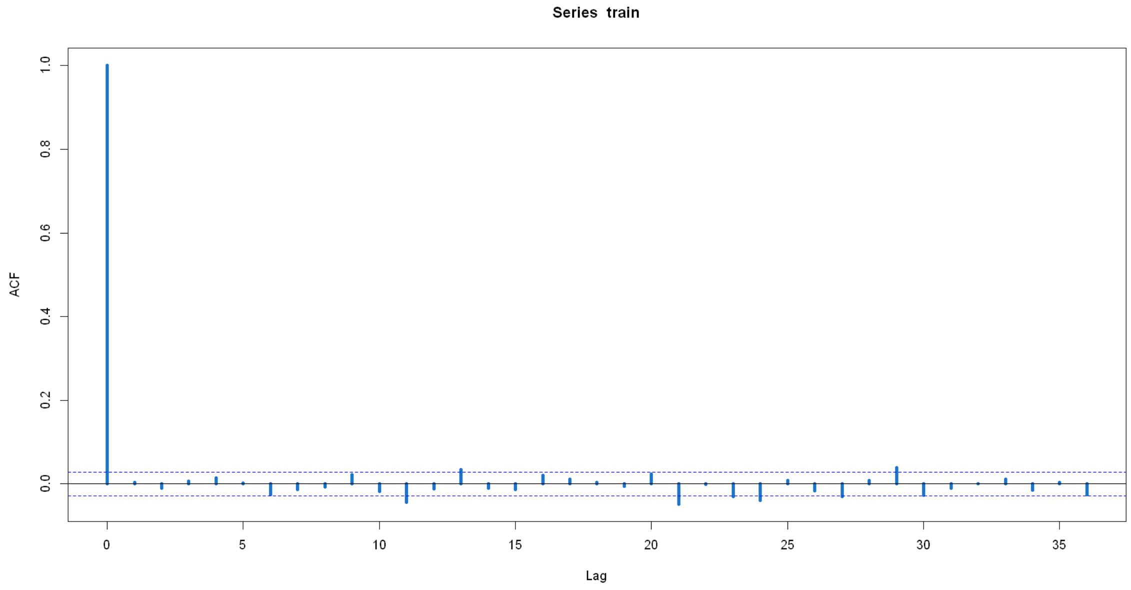
```
Warning message in adf.test(train):
"p-value smaller than printed p-value"

	Augmented Dickey-Fuller Test

data:  train
Dickey-Fuller = -16.863, Lag order = 16, p-value = 0.01
alternative hypothesis: stationary
```

Rechazamos la hipótesis nula (que no es estacionario) al observar un p-value menor que 0,05 (significativo).

Estudio de correlaciones

In [12]:

```
acf(train, lwd=4, col=4)
```



Luego de aplicar el diff, el resultado del ACF es cercano a ruido blanco, por lo tanto podemos estar en presencia de un paseo al azar con deriva.

In [13]:

```
#Aplicando el auto arima de R llegamos a la misma conclusión.
auto.arima(train)
```

```
Series: train
ARIMA(0,0,0) with non-zero mean

Coefficients:
       mean
      4e-04
s.e.  2e-04

sigma^2 estimated as 0.0001165:  log likelihood=14741.24
AIC=-29478.47   AICc=-29478.47   BIC=-29465.54
```

Descartamos la posibilidad de utilizar un modelo ARIMA ya que para su aplicación la señal debe ser estacionaria y con varianza constante; como acabamos de mostrar anteriormente si bien la señal es estacionaria en la media, no tiene varianza constante.

A continuación, analizaremos el comportamiento de la varianza utilizando ventanas de tiempo.

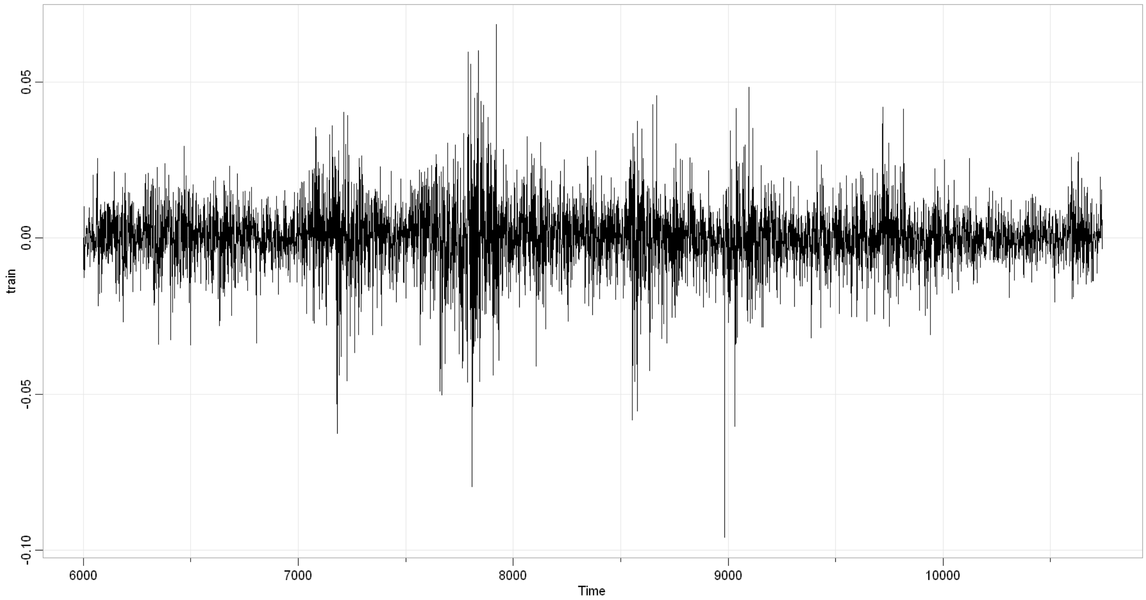
Analisis de varianza

In [14]:

```
sd(train)
var(train)
tsplot(train)
```

0.0107936391454068

0.000116502646001259

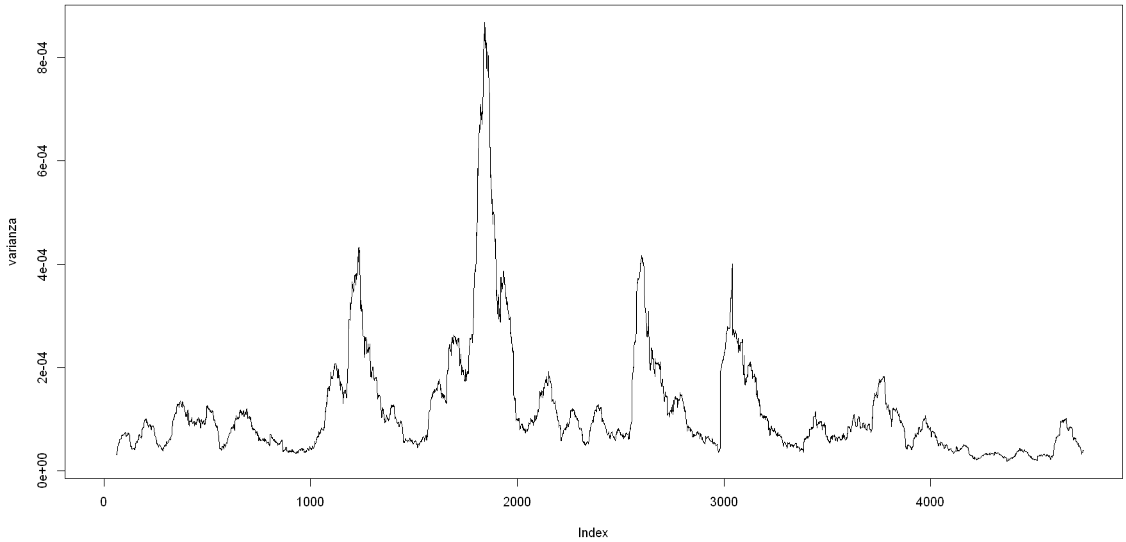


In [15]:

```
vent<- 60 #Tomamos una ventana de dos meses
vent2<- vent-1


varianza=c()
for(i in vent:length(train))
    {f=var(train[(i-vent2):i])
    varianza[i]=(f)}
```

In [16]:

```
plot(varianza,type="l")
```



Se puede observar como en ventanas de dos meses la varianza no es constante.

In [17]:

```
y<-train
```

In [18]:

```
STDM=c()
for(i in vent:length(y))
    {f=sd(y[(i-vent2):i])
    STDM[i]=(f)}

STDM2<-STDM*2
```

In [19]:

```
MM=c()
for(i in vent:length(y))
    {f=mean(y[(i-vent2):i])
    MM[i]=(f)}
```

In [20]:

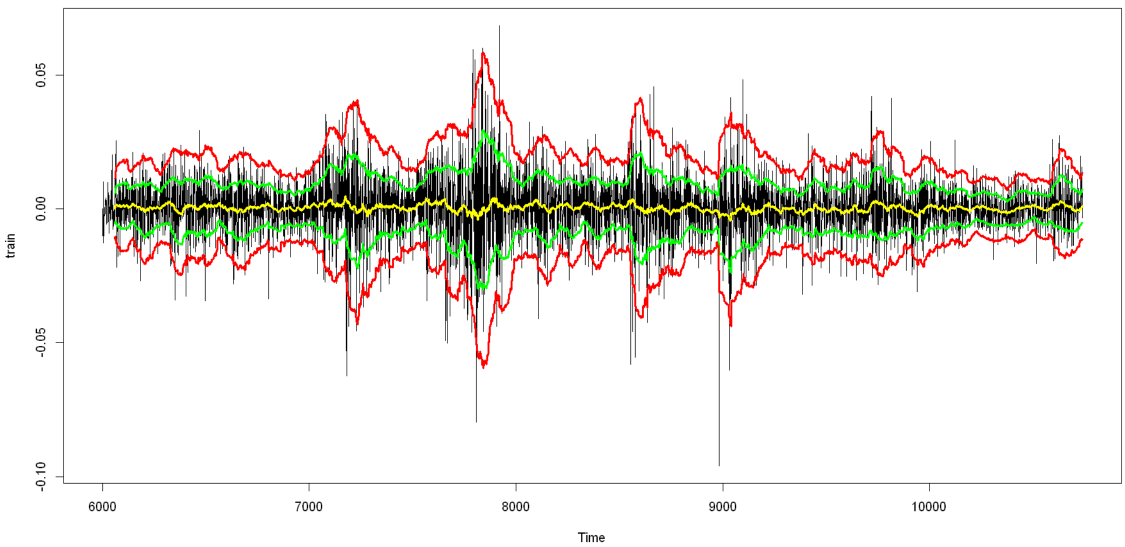
```
#desviacion típica
sd1= MM + STDM # un desvio estandar positivo
sd11= MM-STDM # un desvio estandar negativa
```

In [21]:

```
sd2<-MM+STDM2 # dos desvio estandar positivo
sd22<-MM-STDM2 # dos desvio estandar negativa
```

In [22]:

```
plot(train,type="l")
lines(ts(MM,start = 6000), col="yellow",lwd = 3) # Media
lines(ts(sd11,start = 6000), col="green",lwd = 3)# un desvio estandar negativa
lines(ts(sd1,start = 6000),col="green",lwd = 3) # un desvio estandar positivo
lines(ts(sd2, start = 6000),col="red",lwd = 3) # dos desvio estandar positivo
lines(ts(sd22, start = 6000),col="red",lwd = 3) # dos desvio estandar negativa
```



Confirmamos con el análisis anterior la volatilidad de la varianza; debido a esto es que podemos pensar en la aplicación de un modelo ARCH o GARCH.

Vamos a explorar la aplicabilidad de otros modelos que se basan en señales con alta varianza.

La prueba de efecto ARCH (Arch Test) es una prueba de ruido blanco para la serie de tiempo al cuadrado, en nuestro caso sería la señal al cuadrado. En otras palabras, es la investigación de un orden superior no lineal de autocorrelación.

Por ende, una prueba ARCH significativa nos indicaría que la volatilidad de la varianza puede ser capturable mediante el uso de estos modelos.

In [23]:

```
#Hacemos test de aplicacion de arch
archtest = archtest(ts = as.vector(train))
archtest
```

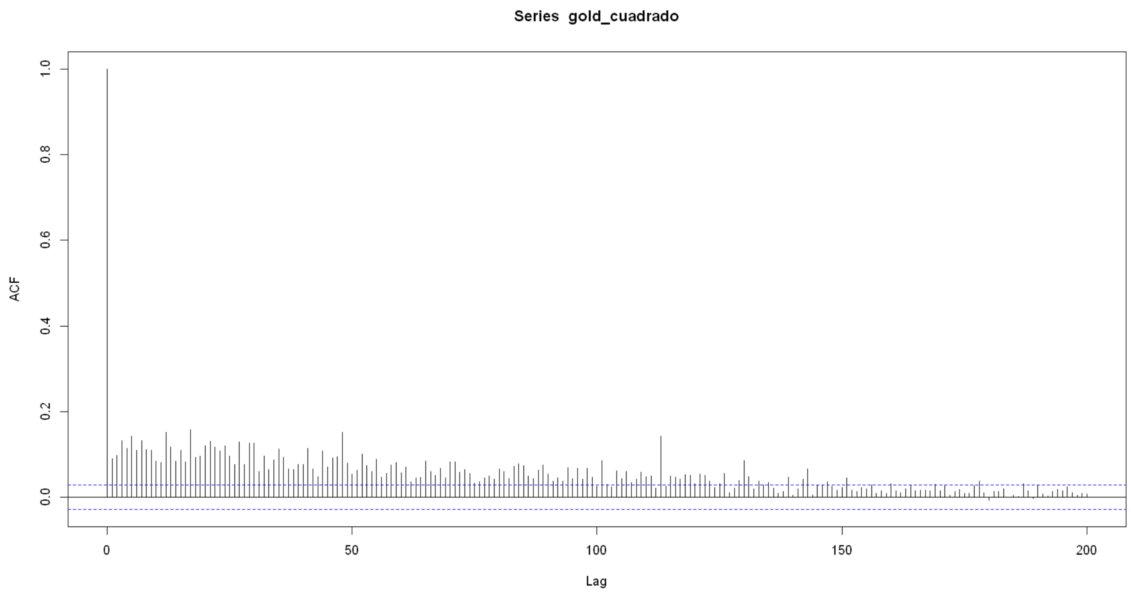
```
	Engle's LM ARCH Test

data:  as.vector(train)
statistic = 38.623, lag = 1, p-value = 5.142e-10
alternative hypothesis: ARCH effects of order 1 are present
```

Observamos que nuestra serie tiene efectos ARCH; el siguiente paso es estimar el orden de este modelo.

Orden ARCH

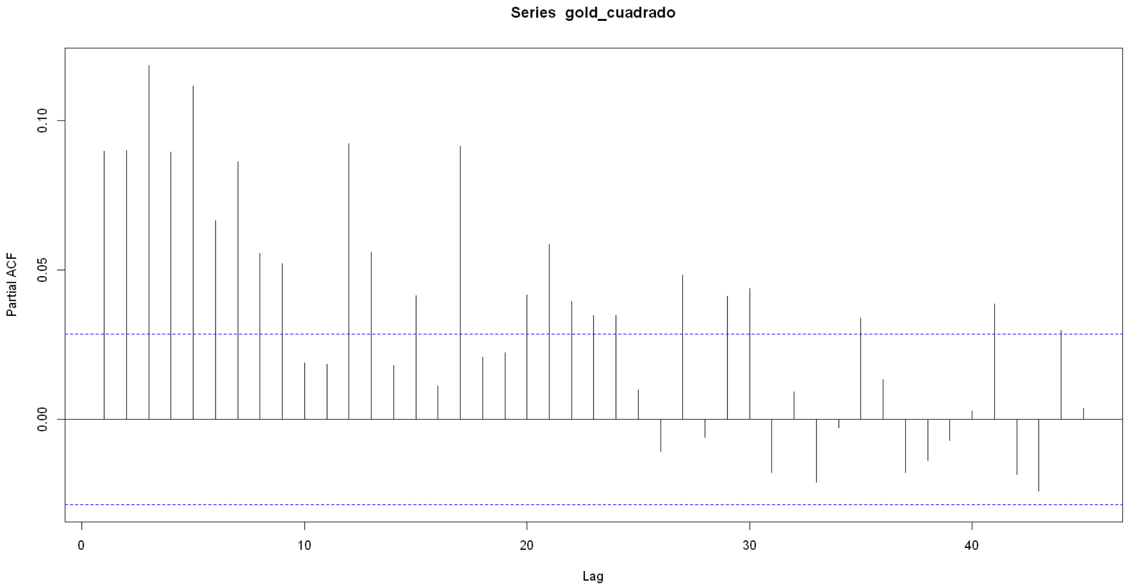
In [24]:

```
gold_cuadrado = train^2
acf(gold_cuadrado,lag.max = 200)
```



In [25]:

```
pacf(gold_cuadrado,lag.max = 45)
```



En función del PACF optamos por un orden 7 para la obtención de un modelo ARCH en nuestra señal. Al ejecutar este modelo tuvimos problemas de convergencia por lo que aumentamos el orden a 9.

Si bien mas allá del lag 10 existen algunas barras por arriba de las intervalos de confianza, entendemos que con esta selección lograremos capturar correctamente la varianza del modelo.

ARCH

Comenzamos por aplicar un modelo básico como es el ARCH (auto-regressive conditional heteroskedasticity).

Este tipo de modelo es similar al autorregresivo, con la diferencia de que usa los cuadrados de las perturbaciones anteriores.

$$y_t = \mu + \epsilon_t$$
$$\epsilon_t = \sigma_t z_t$$

Donde:

y_t = Serie

μ = Media estimada de la serie

ϵ_t = Residuos de retorno con respecto a un proceso medio. Siendo la esperanza de estos retornos 0.

σ_t = Desvio estandar dependiente del tiempo.

z_t = Ruido blanco

La variable aleatoria z_t es un ruido blanco. Las series σ_t^2 son modeladas por:

$$\sigma_t^2 = \omega + \alpha_1 \epsilon_{t-1}^2 + \cdots + \alpha_q \epsilon_{t-q}^2 = \omega + \sum_{i=1}^{q} \alpha_i \epsilon_{t-i}^2$$

donde $\omega > 0$ y $\alpha_i \geq 0, \ i > 0$

In [26]:

```
gold_arch9_spec <-  ugarchspec(variance.model = list(model="sGARCH",         #Other opt
ions are egarch, fgarch, etc.
                                                     garchOrder=c(9,0)), # You can modi
fy the order GARCH(m,s) here
                               mean.model = list(armaOrder = c(0, 0), include.mean = TR
UE, archm = FALSE, archpow = 1, arfima = FALSE, external.regressors = NULL, archex = FA
LSE),
                                distribution.model = "norm")         #Other distributio
n are "std" for t-distribution, and "ged" for General Error Distribution
gold_arch9 <- ugarchfit(spec=gold_arch9_spec,
                              data=train)
```

In [27]:

```
gold_arch9
```

```
*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model     : sGARCH(9,0)
Mean Model      : ARFIMA(0,0,0)
Distribution    : norm

Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.000332    0.000131   2.5374 0.011168
omega   0.000037    0.000003  14.0724 0.000000
alpha1  0.026989    0.011947   2.2590 0.023881
alpha2  0.069809    0.016245   4.2973 0.000017
alpha3  0.079209    0.015652   5.0607 0.000000
alpha4  0.106514    0.017680   6.0247 0.000000
alpha5  0.074648    0.015738   4.7432 0.000002
alpha6  0.069897    0.017525   3.9884 0.000067
alpha7  0.132438    0.020189   6.5598 0.000000
alpha8  0.057246    0.015654   3.6569 0.000255
alpha9  0.104588    0.019177   5.4539 0.000000

Robust Standard Errors:
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.000332    0.000147   2.2679 0.023336
omega   0.000037    0.000004   9.2376 0.000000
alpha1  0.026989    0.018253   1.4786 0.139247
alpha2  0.069809    0.026961   2.5893 0.009617
alpha3  0.079209    0.019873   3.9858 0.000067
alpha4  0.106514    0.027303   3.9012 0.000096
alpha5  0.074648    0.021636   3.4503 0.000560
alpha6  0.069897    0.027320   2.5584 0.010515
alpha7  0.132438    0.041628   3.1815 0.001465
alpha8  0.057246    0.020310   2.8186 0.004824
alpha9  0.104588    0.026926   3.8843 0.000103

LogLikelihood : 15134.11

Information Criteria
------------------------------------

Akaike       -6.3811
Bayes        -6.3661
Shibata      -6.3811
Hannan-Quinn -6.3758

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                      2.649  0.1036
Lag[2*(p+q)+(p+q)-1][2]     2.713  0.1679
Lag[4*(p+q)+(p+q)-1][5]     2.773  0.4501
d.o.f=0
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                         statistic p-value
```

```
Lag[1]                        0.092  0.7616
Lag[2*(p+q)+(p+q)-1][26]     13.860  0.4250
Lag[4*(p+q)+(p+q)-1][44]     28.427  0.1414
d.o.f=9

Weighted ARCH LM Tests
------------------------------------
             Statistic Shape Scale P-Value
ARCH Lag[10]   0.01698 0.500 2.000  0.8963
ARCH Lag[12]   0.03076 1.492 1.843  0.9983
ARCH Lag[14]   0.19292 2.466 1.738  0.9987

Nyblom stability test
------------------------------------
Joint Statistic:  3.061
Individual Statistics:
mu     0.3031
omega  1.5186
alpha1 0.1125
alpha2 0.3176
alpha3 0.2763
alpha4 0.5638
alpha5 0.7135
alpha6 0.2156
alpha7 0.1668
alpha8 0.5560
alpha9 0.3367

Asymptotic Critical Values (10% 5% 1%)
Joint Statistic:         2.49 2.75 3.27
Individual Statistic:    0.35 0.47 0.75

Sign Bias Test
------------------------------------
                   t-value     prob sig
Sign Bias           1.5408 0.123423
Negative Sign Bias  0.5798 0.562055
Positive Sign Bias  1.2029 0.229070
Joint Effect       13.2420 0.004142 ***


Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20     450.0    1.676e-83
2    30     515.5    1.886e-90
3    40     627.7   3.000e-107
4    50     717.8   1.275e-119


Elapsed time : 2.484192
```

GARCH

En este modelo la varianza condicional no solo depende de los cuadrados de las perturbaciones anteriores, sino que además toma en cuenta una componente similar a la media móvil, con la diferencia que dicha componente se calcula con las varianzas condicionales de periodos anteriores, es decir, de σ_t^2 pasados.

$$\sigma_t^2 = \omega + \alpha_1\epsilon_{t-1}^2 + \cdots + \alpha_q\epsilon_{t-q}^2 + \beta_1\sigma_{t-1}^2 + \cdots + \beta_p\sigma_{t-p}^2 = \omega + \sum_{i=1}^{q}\alpha_i\epsilon_{t-i}^2 + \sum_{i=1}^{p}\beta_i\sigma_{t-i}^2$$

Los coeficientes ω, α, β, los encontramos utilizando Máxima Verosimilitud.

In [28]:

```
gold_garch99_spec <-  ugarchspec(variance.model = list(model="sGARCH",         #Other options are egarch, fgarch, etc.
                                                     garchOrder=c(9,9)), # You can modify the order GARCH(m,s) here
                               mean.model = list(armaOrder = c(0, 0), include.mean = TRUE, archm = FALSE, archpow = 1, arfima = FALSE, external.regressors = NULL, archex = FALSE), #Specify your ARMA model implying your model should be stationary.
                                distribution.model = "norm")         #Other distribution are "std" for t-distribution, and "ged" for General Error Distribution
gold_garch99 <- ugarchfit(spec=gold_garch99_spec,
                              data=train)
```

In [29]:

```
gold_garch99
```

```
*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model     : sGARCH(9,9)
Mean Model      : ARFIMA(0,0,0)
Distribution    : norm

Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.000288    0.000128 2.261294 0.023741
omega   0.000004    0.000001 3.074121 0.002111
alpha1  0.035742    0.016893 2.115777 0.034364
alpha2  0.011413    0.049618 0.230015 0.818080
alpha3  0.043305    0.013099 3.305986 0.000946
alpha4  0.034881    0.019655 1.774713 0.075945
alpha5  0.008804    0.008245 1.067856 0.285585
alpha6  0.032254    0.025735 1.253308 0.210094
alpha7  0.083603    0.040940 2.042081 0.041143
alpha8  0.006607    0.035513 0.186046 0.852408
alpha9  0.038913    0.068709 0.566352 0.571155
beta1   0.000001    1.008199 0.000001 1.000000
beta2   0.000000    0.607405 0.000001 0.999999
beta3   0.000000    0.662817 0.000000 1.000000
beta4   0.000000    0.408993 0.000001 1.000000
beta5   0.000000    0.236434 0.000002 0.999999
beta6   0.386161    0.324409 1.190351 0.233908
beta7   0.092842    0.804623 0.115386 0.908139
beta8   0.196252    0.299749 0.654719 0.512649
beta9   0.000000    0.421084 0.000001 0.999999

Robust Standard Errors:
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.000288    0.000229 1.257110  0.20871
omega   0.000004    0.000011 0.412445  0.68001
alpha1  0.035742    0.064495 0.554175  0.57946
alpha2  0.011413    0.291103 0.039205  0.96873
alpha3  0.043305    0.067648 0.640153  0.52207
alpha4  0.034881    0.074312 0.469385  0.63880
alpha5  0.008804    0.045179 0.194870  0.84549
alpha6  0.032254    0.218436 0.147660  0.88261
alpha7  0.083603    0.280752 0.297782  0.76587
alpha8  0.006607    0.273574 0.024151  0.98073
alpha9  0.038913    0.431726 0.090135  0.92818
beta1   0.000001    5.797948 0.000000  1.00000
beta2   0.000000    4.075829 0.000000  1.00000
beta3   0.000000    4.078558 0.000000  1.00000
beta4   0.000000    2.887772 0.000000  1.00000
beta5   0.000000    1.464251 0.000000  1.00000
beta6   0.386161    1.984519 0.194587  0.84572
beta7   0.092842    4.683429 0.019824  0.98418
beta8   0.196252    1.755084 0.111819  0.91097
beta9   0.000000    2.899532 0.000000  1.00000

LogLikelihood : 15241.8

Information Criteria
------------------------------------
```

```
Akaike       -6.4227
Bayes        -6.3954
Shibata      -6.4227
Hannan-Quinn -6.4131

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                      4.236 0.03957
Lag[2*(p+q)+(p+q)-1][2]     4.237 0.06512
Lag[4*(p+q)+(p+q)-1][5]     4.421 0.20603
d.o.f=0
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                         statistic p-value
Lag[1]                      0.5554  0.4561
Lag[2*(p+q)+(p+q)-1][53]   13.5535  0.9964
Lag[4*(p+q)+(p+q)-1][89]   37.9935  0.8147
d.o.f=18

Weighted ARCH LM Tests
------------------------------------
             Statistic Shape Scale P-Value
ARCH Lag[19]    0.2209 0.500 2.000  0.6384
ARCH Lag[21]    0.6041 1.498 1.908  0.8883
ARCH Lag[23]    0.6554 2.489 1.834  0.9817

Nyblom stability test
------------------------------------
Joint Statistic:  373.578
Individual Statistics:
mu     0.46512
omega  0.19175
alpha1 0.06999
alpha2 0.25921
alpha3 0.16641
alpha4 0.33896
alpha5 0.35902
alpha6 0.27581
alpha7 0.10493
alpha8 0.25689
alpha9 0.21491
beta1  0.25517
beta2  0.25251
beta3  0.22527
beta4  0.22790
beta5  0.25138
beta6  0.26007
beta7  0.24181
beta8  0.25328
beta9  0.21031

Asymptotic Critical Values (10% 5% 1%)
Joint Statistic:         4.22 4.52 5.13
Individual Statistic:    0.35 0.47 0.75

Sign Bias Test
------------------------------------
```

```
                   t-value     prob sig
Sign Bias           1.7676 0.077186   *
Negative Sign Bias  0.2193 0.826430
Positive Sign Bias  1.3836 0.166541
Joint Effect       13.1431 0.004337 ***


Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20     382.2    2.221e-69
2    30     477.0    1.518e-82
3    40     590.3    1.259e-99
4    50     659.3   8.795e-108


Elapsed time : 2.791953
```

Observamos que alguno de los alphas son significativos y que ninguno de los beta son significativos. Por lo que agregar el componente de las varianzas condicionales de periodos anteriores no genera una mejora sustancial al modelo ARCH.

Debido a que el alpha 1 y 3 nos dio significativo es que probamos con un GARCH(3,1) y GARCH(1,1). Con esta nueva combinación de parámetros el beta es significativo.

In [30]:

```
gold_garch11_spec <-  ugarchspec(variance.model = list(model="sGARCH",         #Other o
ptions are egarch, fgarch, etc.
                                                     garchOrder=c(1,1)), # You can modi
fy the order GARCH(m,s) here
                               mean.model = list(armaOrder = c(0, 0), include.mean = TR
UE, archm = FALSE, archpow = 1, arfima = FALSE, external.regressors = NULL, archex = FA
LSE), #Specify your ARMA model implying your model should be stationary.
                                distribution.model = "norm")          #Other distributio
n are "std" for t-distribution, and "ged" for General Error Distribution
gold_garch11 <- ugarchfit(spec=gold_garch11_spec,
                              data=train)
```

In [31]:

```
gold_garch11
```

```
*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model     : sGARCH(1,1)
Mean Model      : ARFIMA(0,0,0)
Distribution    : norm

Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.000316    0.000129   2.4584 0.013956
omega   0.000001    0.000001   1.0518 0.292913
alpha1  0.043546    0.007778   5.5988 0.000000
beta1   0.950770    0.007966 119.3542 0.000000

Robust Standard Errors:
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.000316    0.000136 2.325125 0.020065
omega   0.000001    0.000013 0.058458 0.953384
alpha1  0.043546    0.153101 0.284424 0.776086
beta1   0.950770    0.154098 6.169921 0.000000

LogLikelihood : 15224.26

Information Criteria
------------------------------------

Akaike       -6.4220
Bayes        -6.4166
Shibata      -6.4221
Hannan-Quinn -6.4201

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                      3.858 0.04951
Lag[2*(p+q)+(p+q)-1][2]     3.862 0.08213
Lag[4*(p+q)+(p+q)-1][5]     4.011 0.25274
d.o.f=0
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                        statistic p-value
Lag[1]                     0.1812  0.6703
Lag[2*(p+q)+(p+q)-1][5]    0.7223  0.9186
Lag[4*(p+q)+(p+q)-1][9]    2.1783  0.8827
d.o.f=2

Weighted ARCH LM Tests
------------------------------------
            Statistic Shape Scale P-Value
ARCH Lag[3] 0.0007989 0.500 2.000  0.9775
ARCH Lag[5] 0.1986596 1.440 1.667  0.9658
ARCH Lag[7] 1.6588840 2.315 1.543  0.7890

Nyblom stability test
------------------------------------
```

```
Joint Statistic:  431.5044
Individual Statistics:
mu      0.4021
omega  43.2409
alpha1  0.1894
beta1   0.2220

Asymptotic Critical Values (10% 5% 1%)
Joint Statistic:         1.07 1.24 1.6
Individual Statistic:    0.35 0.47 0.75

Sign Bias Test
------------------------------------
                   t-value     prob sig
Sign Bias          1.59879 0.109934
Negative Sign Bias 0.03915 0.968770
Positive Sign Bias 1.28671 0.198258
Joint Effect      11.87160 0.007836 ***


Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20     380.1    5.869e-69
2    30     479.1    5.531e-83
3    40     563.7    3.136e-94
4    50     611.6    3.355e-98


Elapsed time : 0.36604
```

In [32]:

```
gold_garch31_spec <-  ugarchspec(variance.model = list(model="sGARCH",         #Other o
ptions are egarch, fgarch, etc.
                                                      garchOrder=c(3,1)), # You can modi
fy the order GARCH(m,s) here
                               mean.model = list(armaOrder = c(0, 0), include.mean = TR
UE, archm = FALSE, archpow = 1, arfima = FALSE, external.regressors = NULL, archex = FA
LSE), #Specify your ARMA model implying your model should be stationary.
                                distribution.model = "norm")          #Other distributio
n are "std" for t-distribution, and "ged" for General Error Distribution
gold_garch31 <- ugarchfit(spec=gold_garch31_spec,
                              data=train)
```

In [33]:

```
gold_garch31
```

```
*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model     : sGARCH(3,1)
Mean Model      : ARFIMA(0,0,0)
Distribution    : norm

Optimal Parameters
------------------------------------
        Estimate  Std. Error    t value Pr(>|t|)
mu      0.000328    0.000129   2.539177 0.011111
omega   0.000001    0.000000   1.584956 0.112976
alpha1  0.042009    0.012155   3.456120 0.000548
alpha2  0.000005    0.020916   0.000258 0.999794
alpha3  0.002132    0.015991   0.133354 0.893913
beta1   0.950351    0.005761 164.949980 0.000000

Robust Standard Errors:
        Estimate  Std. Error   t value Pr(>|t|)
mu      0.000328    0.000172  1.908268 0.056357
omega   0.000001    0.000005  0.151976 0.879206
alpha1  0.042009    0.045487  0.923536 0.355728
alpha2  0.000005    0.059057  0.000091 0.999927
alpha3  0.002132    0.068362  0.031194 0.975115
beta1   0.950351    0.072611 13.088171 0.000000

LogLikelihood : 15223.94

Information Criteria
------------------------------------

Akaike       -6.4211
Bayes        -6.4129
Shibata      -6.4211
Hannan-Quinn -6.4182

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                      3.919 0.04773
Lag[2*(p+q)+(p+q)-1][2]     3.924 0.07904
Lag[4*(p+q)+(p+q)-1][5]     4.071 0.24536
d.o.f=0
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                         statistic p-value
Lag[1]                      0.2273  0.6335
Lag[2*(p+q)+(p+q)-1][11]    3.0477  0.8630
Lag[4*(p+q)+(p+q)-1][19]    5.4179  0.9153
d.o.f=4

Weighted ARCH LM Tests
------------------------------------
            Statistic Shape Scale P-Value
ARCH Lag[5]    0.3232 0.500 2.000  0.5697
ARCH Lag[7]    2.7142 1.473 1.746  0.3659
```

```
ARCH Lag[9]    3.6262 2.402 1.619  0.4559

Nyblom stability test
------------------------------------
Joint Statistic:  569.4606
Individual Statistics:
mu      0.3694
omega  42.7990
alpha1  0.1705
alpha2  0.1938
alpha3  0.2166
beta1   0.2046

Asymptotic Critical Values (10% 5% 1%)
Joint Statistic:         1.49 1.68 2.12
Individual Statistic:    0.35 0.47 0.75

Sign Bias Test
------------------------------------
                    t-value     prob sig
Sign Bias           1.57559 0.115188    
Negative Sign Bias  0.08383 0.933196    
Positive Sign Bias  1.27145 0.203632    
Joint Effect       11.73052 0.008366 ***


Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20     380.5    4.837e-69
2    30     481.8    1.558e-83
3    40     565.0    1.763e-94
4    50     598.0    1.856e-95


Elapsed time : 0.4476631
```

En función de los resultados anteriores encontramos que los modelos GARCH funcionan muy bien, pero son simétricos. En estos, la varianza condicional depende de la magnitud de las innovaciones retardadas, pero no de su signo. Para solucionar este problema es que surgen los modelos EGARCH que agregan una nueva componente permitiendo capturar la variación del signo en la varianza.

EGARCH

$$\log \sigma_t^2 = \omega + \sum_{k=1}^{q} g(Z_{t-k}) + \sum_{k=1}^{p} \beta_k \log \sigma_{t-k}^2$$

Donde $g(Z_t) = \alpha Z_t + \gamma(|Z_t| - E(|Z_t|))$

Z_t = ϵ_t / σ_t

σ_t^2 es la varianza condicional

ω (contante), β (Efecto GARCH), α (Efecto ARCH) y γ (nuevo componente EGARCH) son los coeficientes.

Z_t puede ser la variable normal estandar o que venga por la distribucion generalizada del error. En nuestro θ deja de tener relevancia al estar trabajando con una distribución normal.

$g(Z_t)$ permite mantener el signo y la magnitud de Z_t, manteniendo separados los efectos en la volatilidad.

Como $\log \sigma_t^2$ puede ser negativo, no hay restricciones para este parametro.

A continuación, vamos a simular con distintos modelos para determinar el orden del p y el q.

Como estos modelos logran capturar mejor el comportamiento de la señal, esperamos encontrar un modelo EGARCH que con menos parámetros logre mejores resultados que un modelo GARCH.

EGARCH (1,1)

In [34]:

```
gold_egarch11_spec <-  ugarchspec(variance.model = list(model="eGARCH",         #Other options are egarch, fgarch, etc.
                                                    garchOrder=c(1,1)), # You can modify the order GARCH(m,s) here
                               mean.model = list(armaOrder=c(0,0)), #Specify your ARMA model implying your model should be stationary.
                               distribution.model = "norm")         #Other distribution are "std" for t-distribution, and "ged" for General Error Distribution
gold_egarch11 <- ugarchfit(spec=gold_egarch11_spec,
                              data=train)
```

In [35]:

```
gold_egarch11
```

```
*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model     : eGARCH(1,1)
Mean Model      : ARFIMA(0,0,0)
Distribution    : norm

Optimal Parameters
------------------------------------
        Estimate  Std. Error   t value Pr(>|t|)
mu      0.000437    0.000126    3.4691 0.000522
omega  -0.056711    0.001371  -41.3515 0.000000
alpha1  0.007693    0.004957    1.5519 0.120679
beta1   0.993106    0.000122 8150.8229 0.000000
gamma1  0.098242    0.003968   24.7614 0.000000

Robust Standard Errors:
        Estimate  Std. Error    t value Pr(>|t|)
mu      0.000437    0.000134    3.26897 0.001079
omega  -0.056711    0.002328  -24.36458 0.000000
alpha1  0.007693    0.013373    0.57521 0.565147
beta1   0.993106    0.000249 3986.83808 0.000000
gamma1  0.098242    0.006640   14.79588 0.000000

LogLikelihood : 15209.28

Information Criteria
------------------------------------

Akaike       -6.4153
Bayes        -6.4085
Shibata      -6.4153
Hannan-Quinn -6.4129

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                      4.028 0.04475
Lag[2*(p+q)+(p+q)-1][2]     4.031 0.07398
Lag[4*(p+q)+(p+q)-1][5]     4.272 0.22205
d.o.f=0
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                        statistic p-value
Lag[1]                     0.8533  0.3556
Lag[2*(p+q)+(p+q)-1][5]    1.1239  0.8310
Lag[4*(p+q)+(p+q)-1][9]    3.1626  0.7318
d.o.f=2

Weighted ARCH LM Tests
------------------------------------
            Statistic Shape Scale P-Value
ARCH Lag[3]  0.009361 0.500 2.000  0.9229
ARCH Lag[5]  0.113526 1.440 1.667  0.9843
ARCH Lag[7]  2.434936 2.315 1.543  0.6259
```

```
Nyblom stability test
------------------------------------
Joint Statistic:  1.3863
Individual Statistics:
mu     0.37663
omega  0.16031
alpha1 0.23545
beta1  0.16073
gamma1 0.06445

Asymptotic Critical Values (10% 5% 1%)
Joint Statistic:         1.28 1.47 1.88
Individual Statistic:    0.35 0.47 0.75

Sign Bias Test
------------------------------------
                   t-value     prob sig
Sign Bias           1.3632 0.172881
Negative Sign Bias  0.5502 0.582185
Positive Sign Bias  1.2273 0.219787
Joint Effect       11.6760 0.008579 ***


Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20     386.5    2.862e-70
2    30     477.1    1.464e-82
3    40     482.3    8.673e-78
4    50     485.9    3.120e-73


Elapsed time : 0.8488121
```

EGARCH (1,2)

In [36]:

```
gold_egarch12_spec <-  ugarchspec(variance.model = list(model="eGARCH",         #Other options are egarch, fgarch, etc.
                                                      garchOrder=c(1,2)), # You can modify the order GARCH(m,s) here
                               mean.model = list(armaOrder=c(0,0)), #Specify your ARMA model implying your model should be stationary.
                               distribution.model = "norm")         #Other distribution are "std" for t-distribution, and "ged" for General Error Distribution
gold_egarch12 <- ugarchfit(spec=gold_egarch12_spec,
                              data=train)
```

In [37]:

```
gold_egarch12
```

```
*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model     : eGARCH(1,2)
Mean Model      : ARFIMA(0,0,0)
Distribution    : norm

Optimal Parameters
------------------------------------
        Estimate  Std. Error     t value Pr(>|t|)
mu      0.000438    0.000086      5.0684 0.000000
omega  -0.056476    0.012217     -4.6228 0.000004
alpha1  0.007647    0.005018      1.5238 0.127550
beta1   0.999997    0.000009 105975.4773 0.000000
beta2  -0.006860    0.001369     -5.0120 0.000001
gamma1  0.097554    0.008923     10.9327 0.000000

Robust Standard Errors:
        Estimate  Std. Error     t value Pr(>|t|)
mu      0.000438    0.000132     3.32396 0.000888
omega  -0.056476    0.025271    -2.23482 0.025429
alpha1  0.007647    0.013408     0.57031 0.568468
beta1   0.999997    0.000012 82373.58837 0.000000
beta2  -0.006860    0.002970    -2.30980 0.020899
gamma1  0.097554    0.018848     5.17575 0.000000

LogLikelihood : 15208.92

Information Criteria
------------------------------------

Akaike       -6.4147
Bayes        -6.4066
Shibata      -6.4147
Hannan-Quinn -6.4119

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                      4.033 0.04462
Lag[2*(p+q)+(p+q)-1][2]     4.035 0.07378
Lag[4*(p+q)+(p+q)-1][5]     4.276 0.22158
d.o.f=0
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                         statistic p-value
Lag[1]                      0.8666  0.3519
Lag[2*(p+q)+(p+q)-1][8]     2.5557  0.7660
Lag[4*(p+q)+(p+q)-1][14]    5.5469  0.6993
d.o.f=3

Weighted ARCH LM Tests
------------------------------------
            Statistic Shape Scale P-Value
ARCH Lag[4]    0.1139 0.500 2.000  0.7358
ARCH Lag[6]    0.2325 1.461 1.711  0.9614
```

```
ARCH Lag[8]    3.5205 2.368 1.583  0.4510

Nyblom stability test
------------------------------------
Joint Statistic:  2.1419
Individual Statistics:
mu     0.38035
omega  0.16068
alpha1 0.23434
beta1  0.16103
beta2  0.16112
gamma1 0.06448

Asymptotic Critical Values (10% 5% 1%)
Joint Statistic:         1.49 1.68 2.12
Individual Statistic:    0.35 0.47 0.75

Sign Bias Test
------------------------------------
                   t-value     prob sig
Sign Bias           1.3615 0.173405
Negative Sign Bias  0.5614 0.574526
Positive Sign Bias  1.2206 0.222291
Joint Effect       11.6698 0.008604 ***


Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20     386.0    3.501e-70
2    30     475.2    3.544e-82
3    40     484.0    3.916e-78
4    50     485.2    4.273e-73


Elapsed time : 1.320657
```

EGARCH (2,1)

In [38]:

```
gold_egarch21_spec <-  ugarchspec(variance.model = list(model="eGARCH",         #Other
 options are egarch, fgarch, etc.
                                                     garchOrder=c(2,1)), # You can modi
fy the order GARCH(m,s) here
                               mean.model = list(armaOrder=c(0,0)), #Specify your ARMA
 model implying your model should be stationary.
                               distribution.model = "norm")         #Other distribution
are "std" for t-distribution, and "ged" for General Error Distribution
gold_egarch21 <- ugarchfit(spec=gold_egarch21_spec,
                              data=train)
```

In [39]:

```
gold_egarch21
```

```
*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model     : eGARCH(2,1)
Mean Model      : ARFIMA(0,0,0)
Distribution    : norm

Optimal Parameters
------------------------------------
        Estimate  Std. Error    t value Pr(>|t|)
mu      0.000463    0.000130     3.5638 0.000365
omega  -0.062782    0.001139   -55.0981 0.000000
alpha1 -0.135071    0.010821   -12.4822 0.000000
alpha2  0.151541    0.013942    10.8696 0.000000
beta1   0.992484    0.000035 28398.9614 0.000000
gamma1  0.049689    0.026880     1.8485 0.064524
gamma2  0.047958    0.027235     1.7609 0.078258

Robust Standard Errors:
        Estimate  Std. Error    t value Pr(>|t|)
mu      0.000463    0.000142    3.25092  0.00115
omega  -0.062782    0.002250  -27.90349  0.00000
alpha1 -0.135071    0.020946   -6.44855  0.00000
alpha2  0.151541    0.025403    5.96549  0.00000
beta1   0.992484    0.000134 7398.08433  0.00000
gamma1  0.049689    0.054984    0.90370  0.36616
gamma2  0.047958    0.059216    0.80988  0.41801

LogLikelihood : 15235.18

Information Criteria
------------------------------------

Akaike       -6.4254
Bayes        -6.4158
Shibata      -6.4254
Hannan-Quinn -6.4220

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                      3.311 0.06883
Lag[2*(p+q)+(p+q)-1][2]     3.316 0.11523
Lag[4*(p+q)+(p+q)-1][5]     3.524 0.31967
d.o.f=0
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                         statistic p-value
Lag[1]                      0.8484  0.3570
Lag[2*(p+q)+(p+q)-1][8]     2.7843  0.7249
Lag[4*(p+q)+(p+q)-1][14]    6.1402  0.6199
d.o.f=3

Weighted ARCH LM Tests
------------------------------------
            Statistic Shape Scale P-Value
```

```
ARCH Lag[4]    0.1703 0.500 2.000  0.6798
ARCH Lag[6]    0.2837 1.461 1.711  0.9492
ARCH Lag[8]    3.7870 2.368 1.583  0.4078

Nyblom stability test
------------------------------------
Joint Statistic:  1.9277
Individual Statistics:
mu     0.43520
omega  0.15275
alpha1 0.31103
alpha2 0.31242
beta1  0.15110
gamma1 0.04451
gamma2 0.06636

Asymptotic Critical Values (10% 5% 1%)
Joint Statistic:         1.69 1.9 2.35
Individual Statistic:    0.35 0.47 0.75

Sign Bias Test
------------------------------------
                   t-value   prob sig
Sign Bias           1.0462 0.2955
Negative Sign Bias  0.6491 0.5163
Positive Sign Bias  0.4325 0.6654
Joint Effect        1.1452 0.7662


Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20     360.1    8.287e-65
2    30     444.1    8.138e-76
3    40     434.6    2.856e-68
4    50     449.3    4.311e-66


Elapsed time : 1.334938
```

EGARCH (3,1)

In [40]:

```
gold_egarch31_spec <-  ugarchspec(variance.model = list(model="eGARCH",         #Other options are egarch, fgarch, etc.
                                                     garchOrder=c(3,1)), # You can modify the order GARCH(m,s) here
                               mean.model = list(armaOrder=c(0,0)), #Specify your ARMA model implying your model should be stationary.
                               distribution.model = "norm")         #Other distribution are "std" for t-distribution, and "ged" for General Error Distribution
gold_egarch31 <- ugarchfit(spec=gold_egarch31_spec,
                              data=train)
```

In [41]:

```
gold_egarch31
```

```
*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model     : eGARCH(3,1)
Mean Model      : ARFIMA(0,0,0)
Distribution    : norm

Optimal Parameters
------------------------------------
        Estimate  Std. Error     t value Pr(>|t|)
mu      0.000463    0.000130  3.5545e+00 0.000379
omega  -0.067558    0.001185 -5.7016e+01 0.000000
alpha1 -0.132791    0.019413 -6.8402e+00 0.000000
alpha2  0.113628    0.029264  3.8828e+00 0.000103
alpha3  0.038695    0.021584  1.7927e+00 0.073019
beta1   0.991946    0.000009  1.1414e+05 0.000000
gamma1  0.056184    0.005342  1.0517e+01 0.000000
gamma2  0.000381    0.029192  1.3052e-02 0.989586
gamma3  0.043791    0.030881  1.4180e+00 0.156181

Robust Standard Errors:
        Estimate  Std. Error     t value Pr(>|t|)
mu      0.000463    0.000142  3.2598e+00 0.001115
omega  -0.067558    0.001935 -3.4906e+01 0.000000
alpha1 -0.132791    0.062391 -2.1284e+00 0.033306
alpha2  0.113628    0.056232  2.0207e+00 0.043309
alpha3  0.038695    0.028336  1.3656e+00 0.172071
beta1   0.991946    0.000022  4.5580e+04 0.000000
gamma1  0.056184    0.017159  3.2742e+00 0.001060
gamma2  0.000381    0.039474  9.6530e-03 0.992298
gamma3  0.043791    0.046622  9.3927e-01 0.347590

LogLikelihood : 15238.33

Information Criteria
------------------------------------

Akaike       -6.4259
Bayes        -6.4136
Shibata      -6.4259
Hannan-Quinn -6.4216

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                      3.328 0.06811
Lag[2*(p+q)+(p+q)-1][2]     3.341 0.11348
Lag[4*(p+q)+(p+q)-1][5]     3.559 0.31436
d.o.f=0
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                         statistic p-value
Lag[1]                      0.7681  0.3808
Lag[2*(p+q)+(p+q)-1][11]    4.2741  0.6916
Lag[4*(p+q)+(p+q)-1][19]    7.3887  0.7429
d.o.f=4
```

```
Weighted ARCH LM Tests
------------------------------------
            Statistic Shape Scale P-Value
ARCH Lag[5]   0.01018 0.500 2.000  0.9196
ARCH Lag[7]   3.97822 1.473 1.746  0.2008
ARCH Lag[9]   5.23681 2.402 1.619  0.2430

Nyblom stability test
------------------------------------
Joint Statistic:  2.0346
Individual Statistics:
mu     0.46432
omega  0.16084
alpha1 0.31539
alpha2 0.31480
alpha3 0.30309
beta1  0.15858
gamma1 0.04949
gamma2 0.06524
gamma3 0.07725

Asymptotic Critical Values (10% 5% 1%)
Joint Statistic:         2.1 2.32 2.82
Individual Statistic:    0.35 0.47 0.75

Sign Bias Test
------------------------------------
                   t-value   prob sig
Sign Bias           1.0964 0.2730
Negative Sign Bias  0.7797 0.4356
Positive Sign Bias  0.3227 0.7470
Joint Effect        1.3574 0.7156


Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20     357.3    3.098e-64
2    30     442.4    1.816e-75
3    40     431.5    1.173e-67
4    50     447.8    8.512e-66


Elapsed time : 1.914667
```

EGARCH (3,2)

In [42]:

```
gold_egarch32_spec <-  ugarchspec(variance.model = list(model="eGARCH",         #Other
 options are egarch, fgarch, etc.
                                                      garchOrder=c(3,2)), # You can modi
fy the order GARCH(m,s) here
                              mean.model = list(armaOrder=c(0,0)), #Specify your ARMA
 model implying your model should be stationary.
                              distribution.model = "norm")         #Other distribution
are "std" for t-distribution, and "ged" for General Error Distribution
gold_egarch32 <- ugarchfit(spec=gold_egarch32_spec,
                             data=train)
```

In [43]:

```
gold_egarch32
```

```
*---------------------------------*
*          GARCH Model Fit        *
*---------------------------------*

Conditional Variance Dynamics
-----------------------------------
GARCH Model     : eGARCH(3,2)
Mean Model      : ARFIMA(0,0,0)
Distribution    : norm

Optimal Parameters
------------------------------------
        Estimate  Std. Error     t value Pr(>|t|)
mu      0.000463    0.000130     3.55237 0.000382
omega  -0.067103    0.016111    -4.16497 0.000031
alpha1 -0.132761    0.012495   -10.62520 0.000000
alpha2  0.114674    0.024568     4.66751 0.000003
alpha3  0.037491    0.021673     1.72986 0.083656
beta1   0.999984    0.000037 27291.33087 0.000000
beta2  -0.007983    0.001763    -4.52785 0.000006
gamma1  0.056229    0.027907     2.01483 0.043922
gamma2 -0.000152    0.041877    -0.00364 0.997096
gamma3  0.043519    0.030974     1.40503 0.160011

Robust Standard Errors:
        Estimate  Std. Error     t value Pr(>|t|)
mu      0.000463    0.000144    3.212833 0.001314
omega  -0.067103    0.022271   -3.013017 0.002587
alpha1 -0.132761    0.018824   -7.052886 0.000000
alpha2  0.114674    0.042281    2.712202 0.006684
alpha3  0.037491    0.033554    1.117343 0.263848
beta1   0.999984    0.000127 7893.368487 0.000000
beta2  -0.007983    0.002388   -3.342624 0.000830
gamma1  0.056229    0.062406    0.901022 0.367577
gamma2 -0.000152    0.082016   -0.001858 0.998517
gamma3  0.043519    0.040254    1.081119 0.279644

LogLikelihood : 15238.34

Information Criteria
------------------------------------

Akaike       -6.4255
Bayes        -6.4118
Shibata      -6.4255
Hannan-Quinn -6.4207

Weighted Ljung-Box Test on Standardized Residuals
------------------------------------
                        statistic p-value
Lag[1]                      3.327 0.06815
Lag[2*(p+q)+(p+q)-1][2]     3.340 0.11354
Lag[4*(p+q)+(p+q)-1][5]     3.559 0.31446
d.o.f=0
H0 : No serial correlation

Weighted Ljung-Box Test on Standardized Squared Residuals
------------------------------------
                         statistic p-value
Lag[1]                      0.7674  0.3810
Lag[2*(p+q)+(p+q)-1][14]    5.7120  0.6774
```

```
Lag[4*(p+q)+(p+q)-1][24]    9.2527  0.7750
d.o.f=5

Weighted ARCH LM Tests
------------------------------------
             Statistic Shape Scale P-Value
ARCH Lag[6]     0.1246 0.500 2.000  0.7241
ARCH Lag[8]     4.9821 1.480 1.774  0.1284
ARCH Lag[10]    6.9782 2.424 1.650  0.1230

Nyblom stability test
------------------------------------
Joint Statistic:  2.3886
Individual Statistics:
mu     0.46454
omega  0.16099
alpha1 0.31545
alpha2 0.31468
alpha3 0.30346
beta1  0.15871
beta2  0.15887
gamma1 0.04971
gamma2 0.06537
gamma3 0.07718

Asymptotic Critical Values (10% 5% 1%)
Joint Statistic:         2.29 2.54 3.05
Individual Statistic:    0.35 0.47 0.75

Sign Bias Test
------------------------------------
                   t-value   prob sig
Sign Bias           1.0969 0.2728
Negative Sign Bias  0.7802 0.4353
Positive Sign Bias  0.3229 0.7468
Joint Effect        1.3586 0.7153


Adjusted Pearson Goodness-of-Fit Test:
------------------------------------
  group statistic p-value(g-1)
1    20     357.2    3.317e-64
2    30     442.5    1.721e-75
3    40     431.9    1.006e-67
4    50     447.7    9.094e-66


Elapsed time : 2.259945
```

Comparo AIC

In [44]:

```
print(paste0("AIC ARCH(9):      ", round(infocriteria(gold_arch9)[1],3)))
print(paste0("AIC SGARCH(9,9): ", round(infocriteria(gold_garch99)[1],3)))
print(paste0("AIC SGARCH(1,1): ", round(infocriteria(gold_garch11)[1],3)))
print(paste0("AIC SGARCH(3,1): ", round(infocriteria(gold_garch31)[1],3)))
print(paste0("AIC EGARCH(1,1): ", round(infocriteria(gold_egarch11)[1],3)))
print(paste0("AIC EGARCH(1,2): ", round(infocriteria(gold_egarch12)[1],3)))
print(paste0("AIC EGARCH(2,1): ", round(infocriteria(gold_egarch21)[1],3)))
print(paste0("AIC EGARCH(3,1): ", round(infocriteria(gold_egarch31)[1],3)))
print(paste0("AIC EGARCH(3,2): ", round(infocriteria(gold_egarch32)[1],3)))
```

```
[1] "AIC ARCH(9):      -6.381"
[1] "AIC SGARCH(9,9): -6.423"
[1] "AIC SGARCH(1,1): -6.422"
[1] "AIC SGARCH(3,1): -6.421"
[1] "AIC EGARCH(1,1): -6.415"
[1] "AIC EGARCH(1,2): -6.415"
[1] "AIC EGARCH(2,1): -6.425"
[1] "AIC EGARCH(3,1): -6.426"
[1] "AIC EGARCH(3,2): -6.425"
```

Comparo BIC

In [45]:

```
print(paste0("BIC ARCH(9):      ", round(infocriteria(gold_arch9)[2],3)))
print(paste0("BIC SGARCH(9,9): ", round(infocriteria(gold_garch99)[2],3)))
print(paste0("BIC SGARCH(1,1): ", round(infocriteria(gold_garch11)[2],3)))
print(paste0("BIC SGARCH(3,1): ", round(infocriteria(gold_garch31)[2],3)))
print(paste0("BIC EGARCH(1,1): ", round(infocriteria(gold_egarch11)[2],3)))
print(paste0("BIC EGARCH(1,2): ", round(infocriteria(gold_egarch12)[2],3)))
print(paste0("BIC EGARCH(2,1): ", round(infocriteria(gold_egarch21)[2],3)))
print(paste0("BIC EGARCH(3,1): ", round(infocriteria(gold_egarch31)[2],3)))
print(paste0("BIC EGARCH(3,2): ", round(infocriteria(gold_egarch32)[2],3)))
```

```
[1] "BIC ARCH(9):      -6.366"
[1] "BIC SGARCH(9,9): -6.395"
[1] "BIC SGARCH(1,1): -6.417"
[1] "BIC SGARCH(3,1): -6.413"
[1] "BIC EGARCH(1,1): -6.408"
[1] "BIC EGARCH(1,2): -6.407"
[1] "BIC EGARCH(2,1): -6.416"
[1] "BIC EGARCH(3,1): -6.414"
[1] "BIC EGARCH(3,2): -6.412"
```

Observando los resultados vemos que el modelo EGARCH (2,1) esta dentro de los mejores AIC y tiene el mejor valor de BIC en comparación al resto de los modelos EGARCH.

Por otro lado, el SGARCH(1,1) tiene el mejor valor de AIC comparado con el resto de los SGARCH y tiene el mejor valor de BIC.

Por lo tanto, viendo que para el AIC el EGARCH(2,1) es superior al SGARCH(1,1) por 0,003 pero para el BIC el SGARCH(1,1) es superior al EGARCH(2,1) por 0.001 es que optamos por continuar nuestro camino con el EGARCH(2,1).

Fundamentamos este camino por dos principales motivos adicionales.

- El primero es que entendemos que un modelo que permite capturar el comportamiento de la varianza positiva o negativa de los retornos es mejor en señales financieras; debido a que los valores al alza o a la baja atraen periodos de tranquilidad o de volatilidad en la varianza.
- El segundo motivo es que la literatura establece que el EGARCH es el modelo que mejor se adapta al tratar de estimar la varianza de los retornos de una serie financiera.

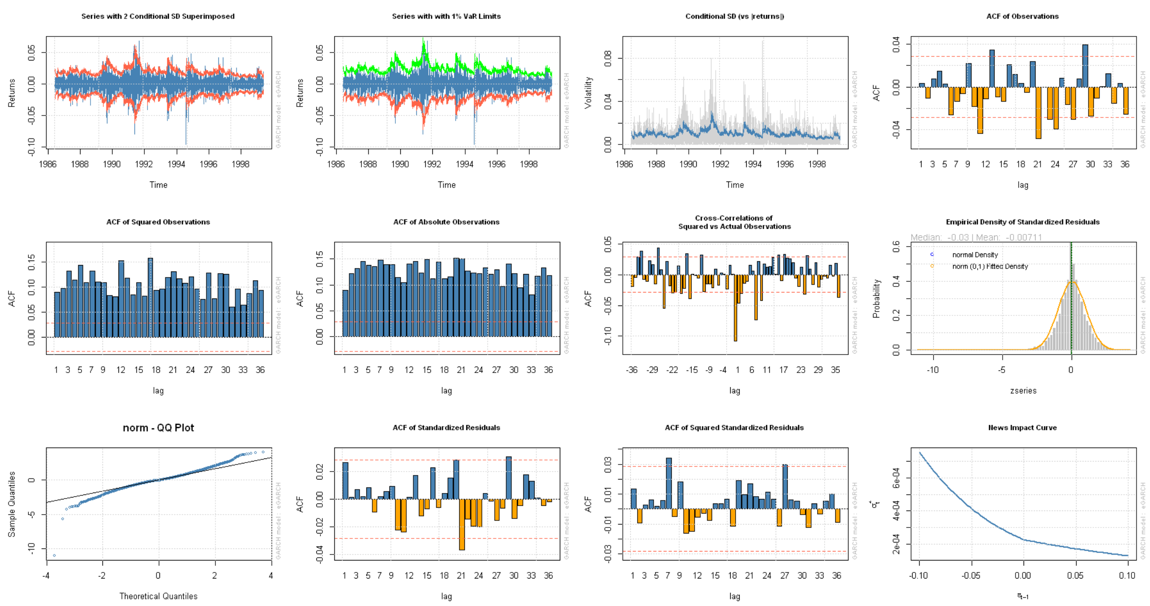
Si bien notamos que el EGARCH(2,1) tiene coeficientes gamma poco significativos, basados en los criterios de selección utilizando AIC y BIC, EGARCH(2,1) se presenta como una de las mejores opciones. Adicionalmente, esta definición está alineada con que estos modelos permiten capturar el comportamiento de la varianza positiva y negativa, lo que lo hacen modelos más flexibles para el análisis de series financieras.

Plots de nuestro mejor EGARCH

In [46]:

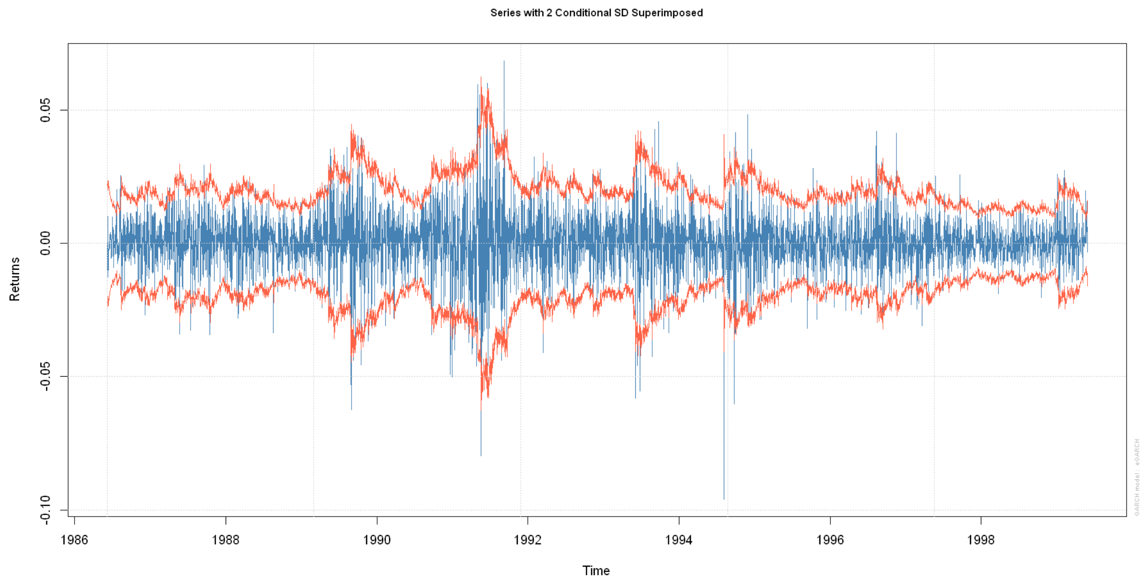
```
plot(gold_egarch21, which = "all")
```

```
please wait...calculating quantiles...
```



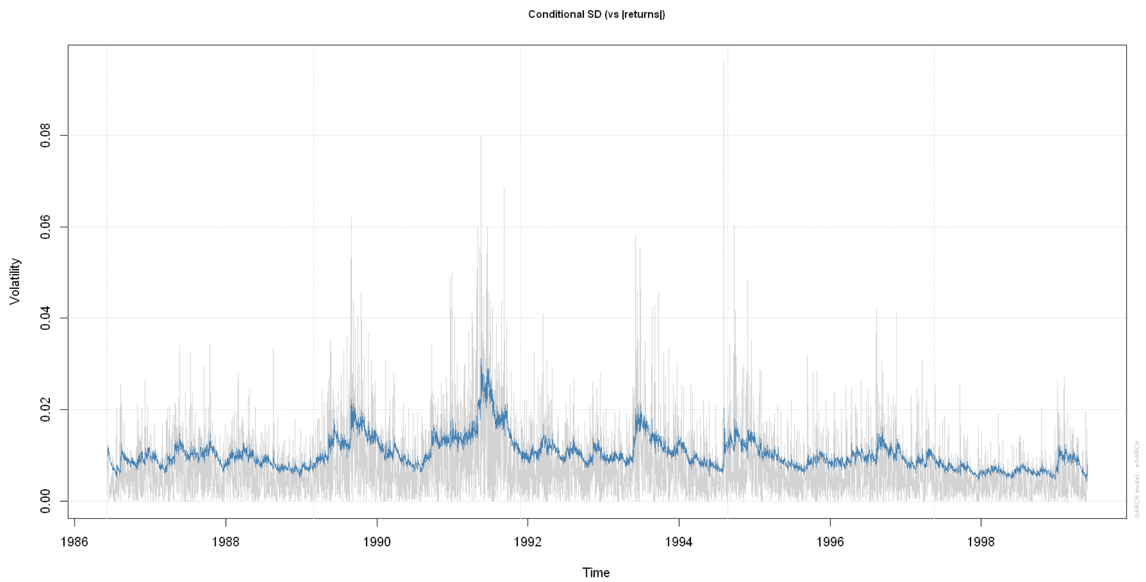
In [47]:

```
plot(gold_egarch21, which = 1)
```



In [48]:

```
plot(gold_egarch21, which = 3)
```



Prediccion con nuestro mejor modelo EGARCH

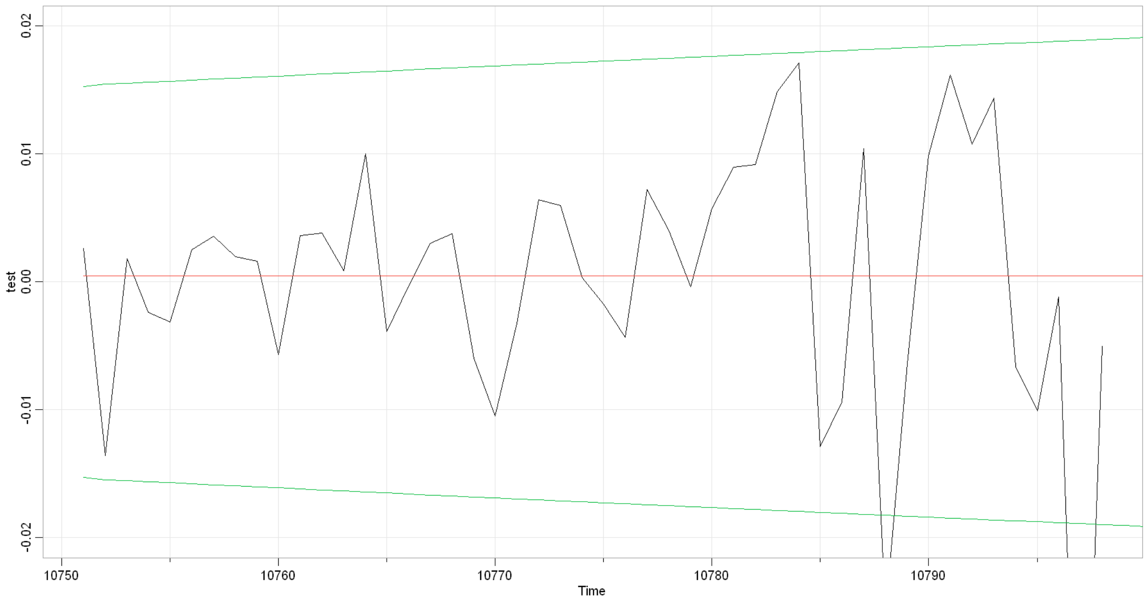
In [49]:

```
predict_gold = ugarchforecast(gold_egarch21,n.ahead = 50)
predict_gold
```

```
*------------------------------------*
*       GARCH Model Forecast         *
*------------------------------------*
Model: eGARCH
Horizon: 50
Roll Steps: 0
Out of Sample: 0

0-roll forecast [T0=1999-05-29]:
        Series    Sigma
T+1  0.0004629 0.007400
T+2  0.0004629 0.007497
T+3  0.0004629 0.007538
T+4  0.0004629 0.007578
T+5  0.0004629 0.007618
T+6  0.0004629 0.007659
T+7  0.0004629 0.007699
T+8  0.0004629 0.007739
T+9  0.0004629 0.007779
T+10 0.0004629 0.007818
T+11 0.0004629 0.007858
T+12 0.0004629 0.007898
T+13 0.0004629 0.007937
T+14 0.0004629 0.007977
T+15 0.0004629 0.008016
T+16 0.0004629 0.008055
T+17 0.0004629 0.008094
T+18 0.0004629 0.008133
T+19 0.0004629 0.008172
T+20 0.0004629 0.008211
T+21 0.0004629 0.008250
T+22 0.0004629 0.008288
T+23 0.0004629 0.008327
T+24 0.0004629 0.008365
T+25 0.0004629 0.008404
T+26 0.0004629 0.008442
T+27 0.0004629 0.008480
T+28 0.0004629 0.008518
T+29 0.0004629 0.008555
T+30 0.0004629 0.008593
T+31 0.0004629 0.008631
T+32 0.0004629 0.008668
T+33 0.0004629 0.008705
T+34 0.0004629 0.008742
T+35 0.0004629 0.008779
T+36 0.0004629 0.008816
T+37 0.0004629 0.008853
T+38 0.0004629 0.008890
T+39 0.0004629 0.008926
T+40 0.0004629 0.008963
T+41 0.0004629 0.008999
T+42 0.0004629 0.009035
T+43 0.0004629 0.009071
T+44 0.0004629 0.009107
T+45 0.0004629 0.009143
T+46 0.0004629 0.009179
T+47 0.0004629 0.009214
T+48 0.0004629 0.009250
T+49 0.0004629 0.009285
T+50 0.0004629 0.009320
```

In [50]:

```
tsplot(test,ylim = c(-0.02,0.02))
lines(ts(predict_gold@forecast$seriesFor,start = 10751),col=2)
lines(ts(predict_gold@forecast$seriesFor + 2*predict_gold@forecast$sigmaFor,start = 107
51),col=3)
lines(-ts(predict_gold@forecast$seriesFor + 2*predict_gold@forecast$sigmaFor,start = 10
751),col=3)
```



Sentido Económico de la Varianza Condicional Estimada

A continuación, le daremos un sentido económico a la varianza condicional estimada. Utilizando la medida de riesgo financiero CVar o Expected Shortfall y el VaR.

VaR (value at risk)

El VaR representa una pérdida máxima asociada con una probabilidad (α).

El VaR al nivel $\alpha \in (0,1)$ es el número más pequeño y tal que la probabilidad de que $Y := X$ no exceda y es al menos $1-\alpha$.

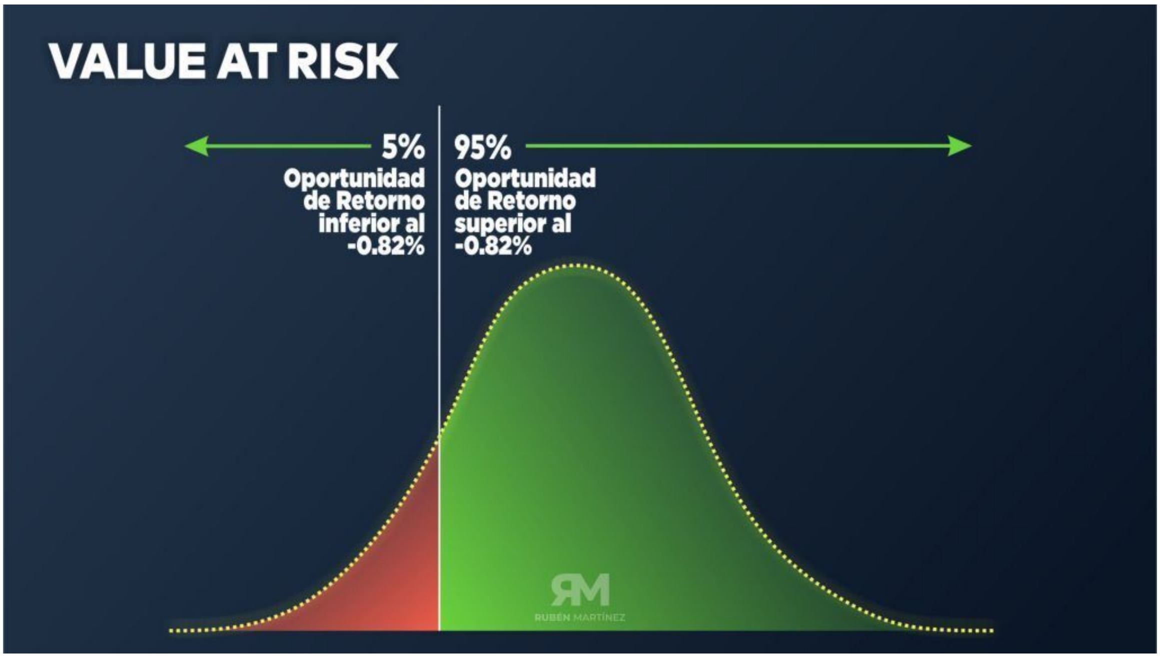
Matemáticamente, $\mathrm{VaR}_\alpha(X)$ es el $(1-\alpha)$-cuantil de Y, es decir:

$$\mathrm{VaR}_\alpha(X) = -\inf\{x \in \mathbb{R} : F_X(x) > \alpha\} = F_Y^{-1}(1-\alpha)$$

Sin embargo esta fórmula no puede ser usada de manera directa para hacer cálculos a menos que se asuma que X tiene una distribución paramétrica.

In [51]:

```
image_read("var.png")
```




```
# A tibble: 1 x 7
  format width height colorspace matte filesize density
  <chr>  <int>  <int> <chr>      <lgl>    <int> <chr>
1 JPEG    1024    576 sRGB       FALSE    34690 72x72
```

CVAR o Expected Shortfall

Mientras que el VaR representa una pérdida máxima asociada con una probabilidad (α) y un horizonte de tiempo definidos, el ES o CVAR es la pérdida esperada si se cruza ese umbral del peor de los casos (perdida máxima). En otras palabras, el ES cuantifica las pérdidas esperadas que ocurren más allá del punto de ruptura del VaR.

El ES o CVAR es el resultado de tomar el promedio ponderado de las observaciones de las cuales la pérdida excede el VaR. Por lo tanto, el ES o CVaR supera la estimación del VaR, ya que puede cuantificar situaciones más arriesgadas, complementando así la información que brinda el VaR.

$$\mathrm{ES}_\alpha(X) = E[-X \mid X \le -\mathrm{VaR}_\alpha(X)] = -\frac{1}{\alpha}\int_0^\alpha \mathrm{VaR}_\alpha(X)\,d\alpha = -\frac{1}{\alpha}\int_{-\infty}^{-\mathrm{VaR}_\alpha(X)} x f(x)\,dx.$$

Expected Shortfall (distribucion normal):

$$\mathrm{ES}_\alpha(X) = -\mu + \sigma\frac{\varphi(\Phi^{-1}(\alpha))}{\alpha}$$

Donde:

$\varphi(x) = \frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}$ es la función de densidad de probabilidad normal estándar.

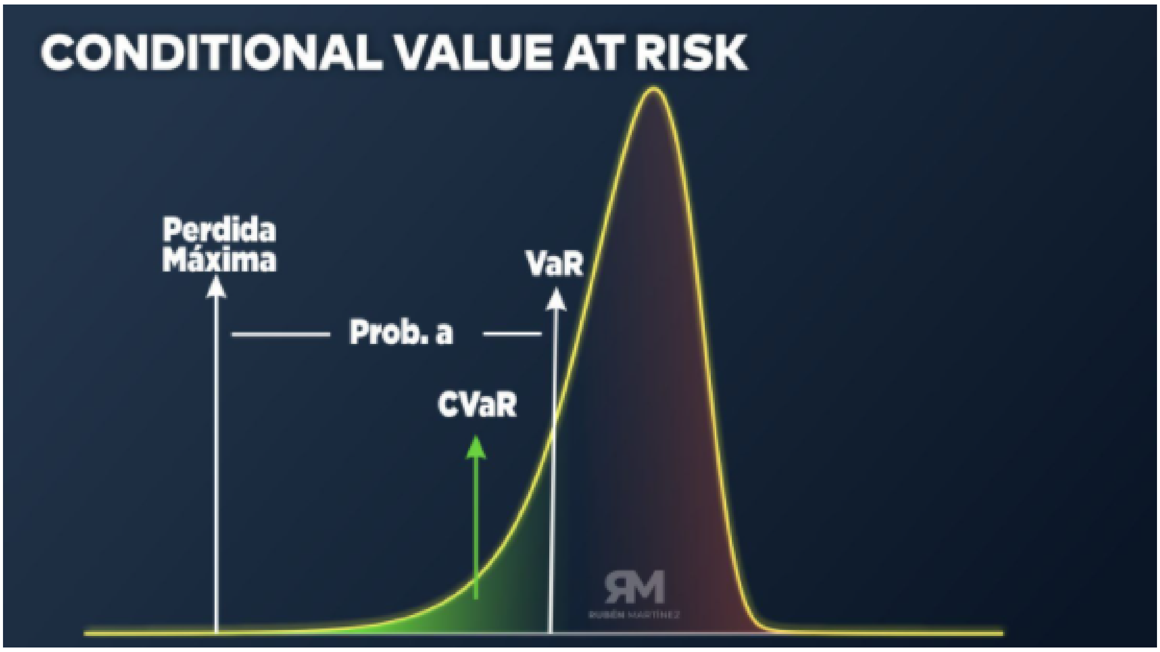
$\Phi(x)$ es la función de distribución acumulada normal estándar. Por lo tanto $\Phi^{-1}(\alpha)$ es el cuantil normal estándar.

μ es el valor medio de la serie

σ es la varianza condicional

In [52]:

```
image_read("cvar vs var.png")
```




```
# A tibble: 1 x 7
  format width height colorspace matte filesize density
  <chr>  <int>  <int> <chr>      <lgl>    <int> <chr>
1 PNG      600    335 sRGB       FALSE   115619 38x38
```

In [53]:

```
sigma = predict_gold@forecast$sigmaFor
length(sigma)
```

50

In [54]:

```
#Fijamos nuestro alfa para el VAR con probabilidad del 5%
alpha = 0.05
VaR_0.05 = ''
for (i in 1:length(sigma)){
    VaR_0.05[i]    = qnorm(alpha) * sigma[i]
    }
```

In [55]:

```
alpha = 0.05
```

In [56]:

```
#ES con probabilidad del 5%
ES_0.05 = ''
#VaR_0.05 = ''

for (i in 1:length(sigma)){
    ES_0.05[i] = -dnorm(qnorm(alpha)) / alpha * sigma[i]
    #VaR_0.05[i]    = qnorm(alpha) * sigma[i]
    }
```

In [57]:

```
alpha = 0.03
```

In [58]:

```
#ES con probabilidad del 3%
ES_0.03 = ''
#VaR_0.03 = ''

for (i in 1:length(sigma)){
    ES_0.03[i] = -dnorm(qnorm(alpha)) / alpha * sigma[i]
    #VaR_0.03[i]    = qnorm(alpha) * sigma[i]
    }
```

In [59]:

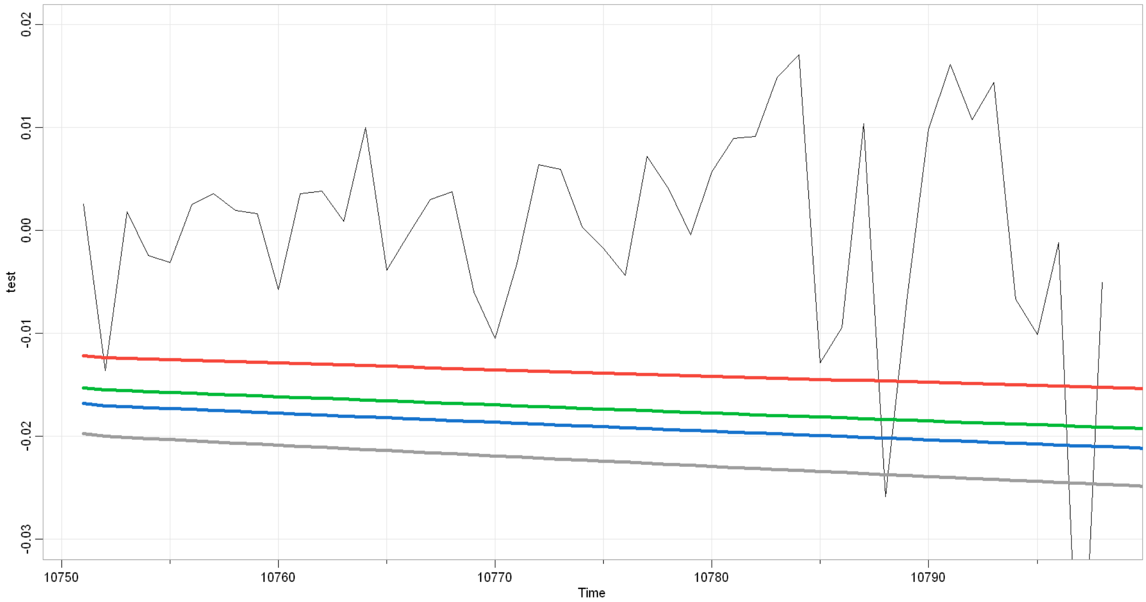
```
alpha = 0.01
```

In [60]:

```
#ES con probabilidad del 1%
ES_0.01 = ''
#VaR_0.01 = ''

for (i in 1:length(sigma)){
    ES_0.01[i] = -dnorm(qnorm(alpha)) / alpha * sigma[i]
    #VaR_0.01[i]   = qnorm(alpha) * sigma[i]
    }
```

In [61]:

```
#Graficamos distintos ES con distintos alphas
tsplot(test,ylim = c(-0.03,0.02))
lines(ts(VaR_0.05,start = 10751),col=2,lwd = 4) #Rojo
lines(ts(ES_0.05,start = 10751),col=3,lwd = 4) #Verde
lines(ts(ES_0.03,start = 10751),col=4,lwd = 4) #Azul
lines(ts(ES_0.01,start = 10751),col=8,lwd = 4) #Gris
```



Ejemplo practico

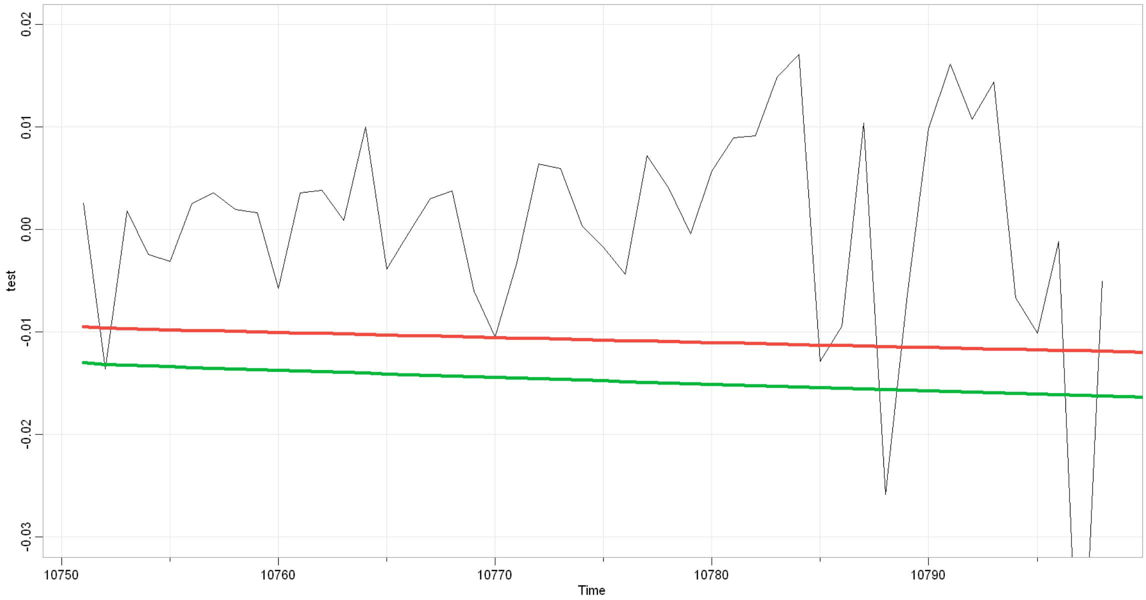
Con alpha = 10%

In [62]:

```
alpha = 0.1
ES_0.10 = ''
VaR_0.10 = ''

for (i in 1:length(sigma)){
    ES_0.10[i] = -dnorm(qnorm(alpha)) / alpha * sigma[i]
    VaR_0.10[i]   = qnorm(alpha) * sigma[i]
    }
```

In [63]:

```
tsplot(test,ylim = c(-0.03,0.02))
lines(ts(VaR_0.10,start = 10751),col=2,lwd = 4) #Rojo
lines(ts(ES_0.10,start = 10751),col=3,lwd = 4) #Verde
```



In [64]:

```
print(paste0("VaR estimado para el segundo día respecto al día anterior: ",round(-(as.n
umeric(VaR_0.10[2])*100),4), " %"))
print(paste0("ES estimado para el segundo día respecto al día anterior: ",round(-(as.nu
meric(ES_0.10[2])*100),4), " %"))
print(paste0("Perdida real para en el segundo día respecto al día anterior: ",round(-te
st[2]*100,4)," %"))
```

```
[1] "VaR estimado para el segundo día respecto al día anterior: 0.9608 %"
[1] "ES estimado para el segundo día respecto al día anterior: 1.3158 %"
[1] "Perdida real para en el segundo día respecto al día anterior: 1.3578
%"
```

En este caso observamos que para el VaR predicho en el día dos, un evento de probabilidad 10% genera una perdida mínima diaria (VaR) de un 0.96% del valor. Sin embargo, utilizando el ES nos permite cuantificar una situación más arriesgada donde en promedio nos estima una pérdida de 1.32% diaria frente al mismo evento de probabilidad del 10%.

Como podemos apreciar en el grafico donde la traza negra es el retorno diario real, el ES es una estimación más certera de lo que ocurrió.

Suponiendo que realizamos una inversión de 100 dólares en oro el día 1. El modelo con ES estima que el día dos tengo un 10% de probabilidad de perder en promedio 1.32 dólares, con el diario del lunes (test) el cálculo del ES fue una correcta estimación de lo que podría suceder. Si hubiéramos tomado el VaR como indicador, nuestro cliente hubiera pensado que la probabilidad de ocurrencia de este evento (perdida de 1.32 dólares) sería del 5%, ya que no considera lo que sucede luego de cruzar ese umbral.

Analisis a un paso del ejemplo practico

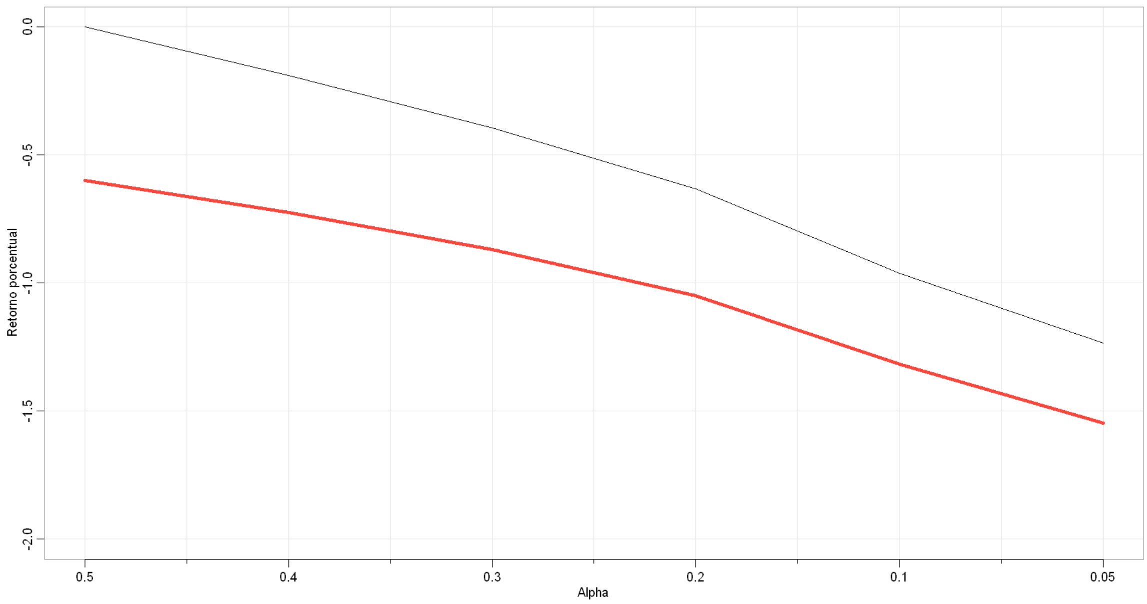
Visto lo anterior, parece interesante plantear diferentes escenarios a un paso más que hacer una predicción a muchos pasos. A continuación, presentaremos la construcción de alternativas a un paso en función de variaciones en el alpha tanto para VaR como para ES.

In [65]:

```
alphab = c(0.5,0.4,0.3,0.2,0.1,0.05)
ES_a_un_paso = ''
VaR_a_un_paso = ''

for (i in 1:length(alphab)){
    ES_a_un_paso[i] = -dnorm(qnorm(alphab[i])) / alphab[i] * sigma[2]
    VaR_a_un_paso[i]   = qnorm(alphab[i]) * sigma[2]
    }
```

In [66]:

```
tsplot(as.numeric(VaR_a_un_paso)*100, ylim = c(-2,0), xlab = "Alpha", ylab = "Retorno p
orcentual",  xaxt='n')
lines(as.numeric(ES_a_un_paso)*100,col=2,lwd = 4) #Rojo
axis(1,at = 1:6,labels = alphab)
```



In [67]:

```
#Verificacion
print(as.numeric(ES_a_un_paso[5])*100)
print(as.numeric(ES_0.10[2])*100)
```

```
[1] -1.315755
[1] -1.315755
```

Bibliografia

-Time Series Analysis and Its Applications With R - Robert H. Shumway y David S. Stoffer - Libro de clase

-MODELOS ARCH, GARCH Y EGARCH: APLICACIONES A SERIES FINANCIERAS - http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0121-47722008000100011 (http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0121-47722008000100011)

-https://en.wikipedia.org/wiki/Autoregressive_conditional_heteroskedasticity (https://en.wikipedia.org/wiki/Autoregressive_conditional_heteroskedasticity)

-https://en.wikipedia.org/wiki/Expected_shortfall (https://en.wikipedia.org/wiki/Expected_shortfall)

-https://www.quantstart.com/articles/Generalised-Autoregressive-Conditional-Heteroskedasticity-GARCH-p-q-Models-for-Time-Series-Analysis/ (https://www.quantstart.com/articles/Generalised-Autoregressive-Conditional-Heteroskedasticity-GARCH-p-q-Models-for-Time-Series-Analysis/)