

Ensayo sobre el cálculo de volatilidades

Julio César López Becerra

Table of contents

Introducción	1
Descarga de datos	3
Métodos de cálculo	4
Datos y gráfica preliminar	5
Volatilidad convencional o fija en el tiempo	6
Volatilidad con suavizamiento exponencial	6
Volatilidad GARCH	7
Resultados de volatilidades	7
Estimación de ventanas móviles	10
Backtest (prueba de Kupiec)	12
Conclusiones	16

Introducción

La volatilidad mide la incertidumbre o riesgo asociado a la evolución del precio de un activo. Por esta razón es un concepto fundamental en el análisis financiero, ya que puede usarse para cosas como: valorar activos, gestión de riesgos, construcción de portafolios y cálculo del VaR. Suele estimarse mediante la desviación estándar de los rendimientos.

La conglomeración de la volatilidad (volatility clustering) se refiere a un fenómeno señalado por primera vez en 1963. Fue descrito como: los grandes cambios son seguidos por grandes cambios, y los pequeños cambios por pequeños cambios. Frase que, adaptada más técnicamente,

podríamos decir que existen periodos de alta volatilidad y periodos de baja volatilidad, a lo que a veces llamamos la “memoria del mercado”. Esto ha motivado al desarrollo de modelos que van más allá de la desviación estándar.

En este documento se analizan tres enfoques para resolver este mismo problema: la volatilidad fija, volatilidad con suavizamiento exponencial y volatilidad GARCH.

```
# Librerías de yahoo finance de Github:
source("https://raw.githubusercontent.com/OscarVDelatorreTorres/yahooFinance/main/datosMulti")
```

```
[1] "All the quantitative finance functions loaded..."
```

```
# Librerías de cuantificación de riesgos en Github:
source("https://raw.githubusercontent.com/OscarVDelatorreTorres/riskManagementSuiteR/refs/heads/main/riskManagementSuiteR.R")
# Librerías de uso general:
library(plotly)
library(dplyr)
library(tseries)
library(quantmod)
library(zoo)
library(rugarch)
library(tidyverse)
library(tibble)
library(forecast)
library(doParallel)
library(foreach)
library(DT)
library(ggplot2)
```

```
#El método no convergía en algunas ventanas y tomaba demasiado tiempo, por lo que se usa una
rollGARCH_parallel <- function(x, model = "sGARCH", LLF = "norm",
                               ventana = 250, garchOrder = c(1,1),
                               arma = c(0,0), include.mean = TRUE,
                               cores = 15) {

  x <- as.numeric(x)
  n <- length(x)

  sig <- rep(NA_real_, n)
  idx <- seq(ventana, n)

  spec <- rugarch::ugarchspec(
```

```

    variance.model = list(model = model, garchOrder = garchOrder),
    mean.model      = list(armaOrder = arma, include.mean = include.mean),
    distribution.model = LLF
  )

  cl <- parallel::makeCluster(cores)
  doParallel::registerDoParallel(cl)

  parallel::clusterEvalQ(cl, {
    library(rugarch)
  })

  sig_idx <- foreach::foreach(t = idx, .combine = "c", .inorder = TRUE,
                              .export = c(),
                              .packages = c("rugarch")) %dopar% {

    win <- x[(t - ventana + 1):t]

    out <- tryCatch({
      fit <- rugarch::ugarchfit(
        spec = spec,
        data = win,
        solver = "hybrid",
        fit.control = list(eval.se = FALSE)
      )
      as.numeric(tail(rugarch::sigma(fit), 1))
    }, error = function(e) NA_real_)

    out
  }

  parallel::stopCluster(cl)
  sig[idx] <- sig_idx
  sig
}

```

Descarga de datos

```

tickerV=c("QQQ")
hastaD= as.Date("2024-11-20")

```

```
deD = as.Date("2019-11-20")
per="D"
paridadFX="USDMXN=X"
convertirFX=c(FALSE)
```

```
Datos=historico_multiples_precios(tickers=tickerV,de=deD,hasta=hastaD,periodicidad=per,fxRate=paridadFX)
```

```
[1] "Extrayendo RIC 1 de 1 (QQQ), periodicidad D"
```

```
[1] "Extrayendo QQQ..."
```

```
[1] "QQQ extraído de Yahoo Finance..."
```

```
[1] "Se terminó de extraer y procesar un total de 1 tickers desde las BD de Yahoo Finance..."
```

```
[1] "Tickers procesados: QQQ"
```

```
precios=Datos$tablaPrecios
rendimientos=Datos$tablaRendimientosCont
rendimientos_decimal <- rendimientos$QQQ/100
monto <- 500000
```

Métodos de cálculo

En esta sección se realizará un análisis de volatilidades para el EFT QQQ en pesos mexicanos, con una inversión de \$500,000 MXN para el periodo de 2023/11/20 a 2024/11/20.

Se estimarán las siguientes volatilidades:

- Fija
- EWMA con $\lambda = 0.95$
- EWMA con $\lambda = 0.98$
- GARCH(1,1) con distribución normal
- GARCH(1,1) con distribución t-student
- GARCH(1,1) con distribución GED
- GJR-GARCH(1,1) con distribución normal
- GJR-GARCH(1,1) con distribución t-student
- GJR-GARCH(1,1) con distribución GED

Se determinará el mejor ajuste por medio del AIC, y posteriormente se calcularán y graficarán estas volatilidades en una ventana móvil de 250 días desde el 2019/11/20 para observar el conglomerado de volatilidades.

Datos y gráfica preliminar

Tabla de precios diarios

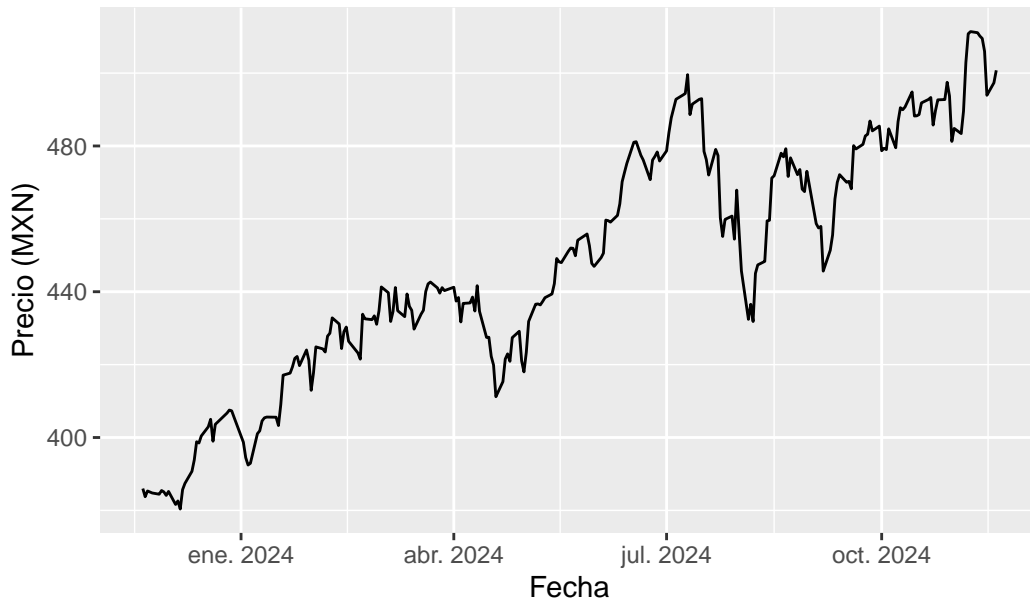
```
precios_250 <- tail(precios, 252)
rendimientos_250 <- tail(rendimientos_decimal, 252)
if (knitr::is_html_output()) {
  DT::datatable(precios_250)
} else {
  knitr::kable(head(precios_250, 10))
}
```

	Date	QQQ
1007	2023-11-20	385.9952
1008	2023-11-21	383.7528
1009	2023-11-22	385.3234
1010	2023-11-24	384.7801
1011	2023-11-27	384.4442
1012	2023-11-28	385.4518
1013	2023-11-29	385.0764
1014	2023-11-30	384.1084
1015	2023-12-01	385.2049
1016	2023-12-04	381.6288

Gráfica de precios diarios

```
if(knitr::is_html_output()){
  plot_ly(precios_250, x = ~Date, y = ~QQQ, type = 'scatter', mode = 'lines',
    line = list(color = 'blue')) %>%
    layout(title = 'Gráfica de precios diarios de QQQ',
      xaxis = list(title = 'Fecha'),
      yaxis = list(title = 'Precio (MXN)'))
} else {
  ggplot(precios_250, aes(x = Date, y = QQQ)) +
    geom_line() +
    labs(title = "Gráfica de precios diarios de QQQ", x = "Fecha", y = "Precio (MXN)")
}
```

Gráfica de precios diarios de QQQ



Volatilidad convencional o fija en el tiempo

Se calcula como la desviación estándar de los rendimientos en un periodo determinado, suponiendo que la varianza es constante. Se basa en el supuesto de homoscedasticidad.

Tiene como ventajas el ser más simple, intuitivo y fácil de implementar, además de requerir una potencia de cómputo menor. Sin embargo, ignora el conglomerado de volatilidades, por lo que suele sobreestimar el riesgo en periodos tranquilos, y subestimarlos en periodos que no lo son.

```
volatilidad_fija <- function(rendimientos){  
  sd(rendimientos)  
}
```

Volatilidad con suavizamiento exponencial

Es una propuesta que mejora al método convencional permitiendo que la varianza cambie con el tiempo, asignando un peso mayor a observaciones más recientes.

Este enfoque reconoce el conglomerado de volatilidades, actualizando la misma con un promedio ponderado de volatilidades y rendimientos. Sigue siendo simple, fácil de implementar y

computar, por lo que presenta una gran ventaja respecto al método convencional. Sin embargo, como gran desventaja, no hay una manera objetiva de asignar el factor de suavizamiento.

```
volatilidad_ewma <- function(rendimientos, lambda){  
  n = length(rendimientos)  
  pesos=lambda^(0:(n-2))  
  pesos_inv = rev(pesos)  
  rend_cuadrado = rendimientos[-1]^2  
  varExp=(1-lambda)*sum(pesos_inv*rend_cuadrado)  
  sqrt(varExp)  
}
```

Volatilidad GARCH

Los modelos GARCH (Generalized Autoregressive Conditional Heteroskedasticity) son la formulación más sofisticada de volatilidad dinámica. La varianza depende tanto de choques pasados como de su historia.

Están diseñados específicamente para capturar el conglomerado de volatilidades, lo que les permite generar pronósticos más coherentes y robustos de la volatilidad futura. Y justo esta es su principal fortaleza, se ajusta a los datos y produce las estimaciones más realistas. Sin embargo, esto viene con una gran carga computacional y complejidad para su implementación, lo que hace que un modelo incorrecto de resultados que parecen correctos, pero no están bien fundamentados.

```
volatilidad_garch <- function(rendimientos, modelo = c("sGARCH", "gjrGARCH"), distribucion =  
  spec <- ugarchspec(variance.model = list(model = modelo,  
                                           garchOrder = c(1, 1)),  
                     mean.model = list(armaOrder = c(0, 0), include.mean = TRUE),  
                     distribution.model = distribucion)  
  garch_fit <- ugarchfit(spec = spec, data = rendimientos)  
  garch_fit  
}
```

```
aic <- function(garch_fit){  
  infocriterias(garch_fit)[1]  
}
```

Resultados de volatilidades

Estimación del IAC mediante los distintos métodos GARCH

```

sigma_fija <- volatilidad_fija(rendimientos_250)
sigma_ewma_95 <- volatilidad_ewma(rendimientos_250, lambda = 0.95)
sigma_ewma_98 <- volatilidad_ewma(rendimientos_250, lambda = 0.98)
modelos <- c(rep("sGARCH", 3), rep("gjrGARCH", 3))
distribuciones <- rep(c("norm", "std", "ged"), 2)
garch_fits <- list()
aics_valores <- numeric(6)

for (i in 1:6) {
  modelo_actual <- modelos[i]
  distribucion_actual <- distribuciones[i]

  fit <- volatilidad_garch(
    rendimientos = rendimientos_250,
    modelo = modelo_actual,
    distribucion = distribucion_actual
  )
  garch_fits[[i]] <- fit
  aics_valores[i] <- aic(fit)
}

tabla_aic <- data.frame(
  Modelo = modelos,
  Distribucion = distribuciones,
  AIC = aics_valores
) %>% arrange(AIC)

if (knitr::is_html_output()) {
  DT::datatable(tabla_aic)
} else {
  knitr::kable(tabla_aic)
}

```

Modelo	Distribucion	AIC
gjrGARCH	ged	-6.272284
sGARCH	ged	-6.251630
gjrGARCH	std	-6.250962
gjrGARCH	norm	-6.240850
sGARCH	std	-6.227694
sGARCH	norm	-6.209379

Volatilidades estimadas por tipo de método

```

tabla_volatilidades_tipo = data.frame(
  Metodo = c("Fija", "EWMA ( =0.95)", "EWMA ( =0.98)",
             paste0("GARCH (", modelos, ", ", " ", distribuciones, ")")),
  Volatilidad = c(sigma_fija, sigma_ewma_95, sigma_ewma_98,
                  sapply(garch_fits, function(fit) {
                    sigma(fit)[length(sigma(fit))]
                  })))
)
if(knitr::is_html_output()) {
  DT::datatable(tabla_volatilidades_tipo)
} else {
  knitr::kable(tabla_volatilidades_tipo)
}

```

Metodo	Volatilidad
Fija	0.0109800
EWMA (=0.95)	0.0115399
EWMA (=0.98)	0.0118336
GARCH (sGARCH, norm)	0.0119095
GARCH (sGARCH, std)	0.0122580
GARCH (sGARCH, ged)	0.0120901
GARCH (gjrGARCH, norm)	0.0129537
GARCH (gjrGARCH, std)	0.0133228
GARCH (gjrGARCH, ged)	0.0133007

Se presenta una gráfica de las volatilidades.

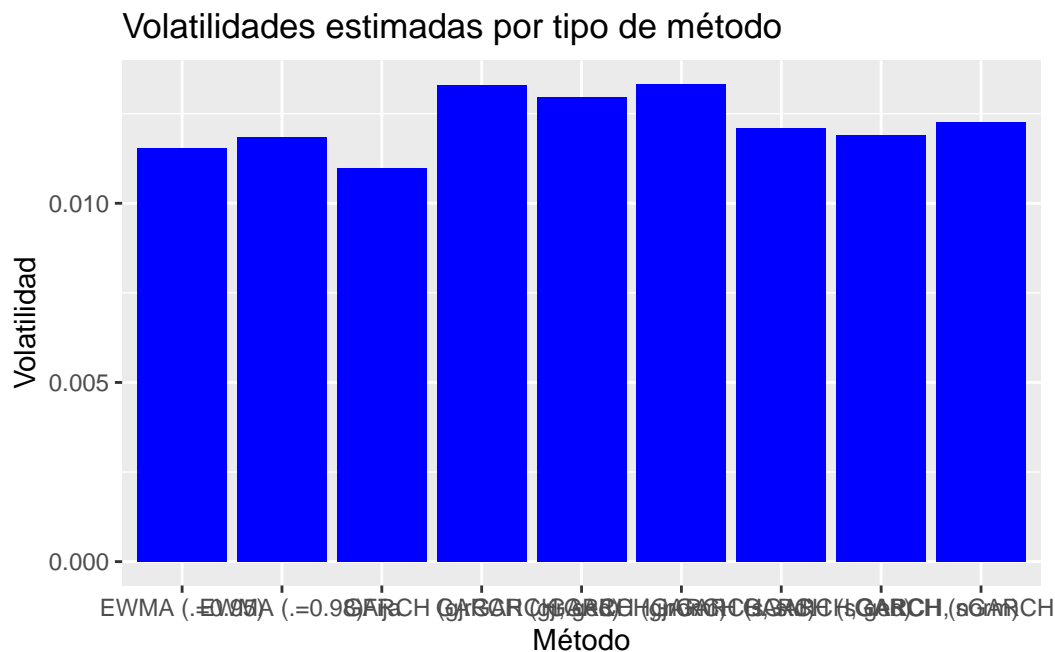
```

if(knitr::is_html_output()){
  plot_ly()%>%
    add_trace(x=~Metodo,y=~Volatilidad,data=tabla_volatilidades_tipo,
              type="bar",name="Volatilidades estimadas") %>%
    layout(title="Volatilidades estimadas por tipo de método",
           xaxis=list(title="Método"),
           yaxis=list(title="Volatilidad"))
} else {
  #| fig-width: 7
  #| fig-height: 4

  ggplot(tabla_volatilidades_tipo, aes(x = Metodo, y = Volatilidad)) +
    geom_bar(stat = "identity", fill = "blue") +

```

```
labs(title = "Volatilidades estimadas por tipo de método", x = "Método", y = "Volatilidad")
}
```



Estimación de ventanas móviles

Ahora, se estimará la volatilidad en una ventana móvil de 250 días para cada uno de los métodos mencionados anteriormente.

```
ventana <- 250
n <- length(rendimientos_decimal)
volatilidades_movil <- data.frame(Date = rendimientos$Date,
  fija = rollapply(rendimientos_decimal, FUN=sd, ventana, fill = NA, align = "right"),
  ews95 = rolleWSigma(rendimientos_decimal, lambda = 0.95, ventana = ventana),
  ews98 = rolleWSigma(rendimientos_decimal, lambda = 0.98, ventana = ventana),
  garch_norm = rollGARCH_parallel(rendimientos_decimal, model = "sGARCH", LLF = "norm", garchO
  garch_std = rollGARCH_parallel(rendimientos_decimal, model = "sGARCH", LLF = "std", garchO
  garch_ged = rollGARCH_parallel(rendimientos_decimal, model = "sGARCH", LLF = "ged", garchO
  gjr_norm = rollGARCH_parallel(rendimientos_decimal, model = "gjrGARCH", LLF = "norm", garcl
  gjr_std = rollGARCH_parallel(rendimientos_decimal, model = "gjrGARCH", LLF = "std", garchO
  gjr_ged = rollGARCH_parallel(rendimientos_decimal, model = "gjrGARCH", LLF = "ged", garchO
```

Gráfica de volatilidades en ventana móvil

```
if(knitr::is_html_output()){
  fig1 <- plot_ly(volatilidades_movil, x = ~Date) %>%
    add_trace(y = ~fija, name = 'Fija', type = 'scatter', mode = 'lines', line = list(color = 'red'))
    add_trace(y = ~ews95, name = 'EWMA (=0.95)', type = 'scatter', mode = 'lines', line = list(color = 'blue'))
    add_trace(y = ~ews98, name = 'EWMA (=0.98)', type = 'scatter', mode = 'lines', line = list(color = 'green'))
    layout(title = '1. Volatilidades: Fija y EWMA',
           xaxis = list(title = 'Fecha'),
           yaxis = list(title = 'Volatilidad'))

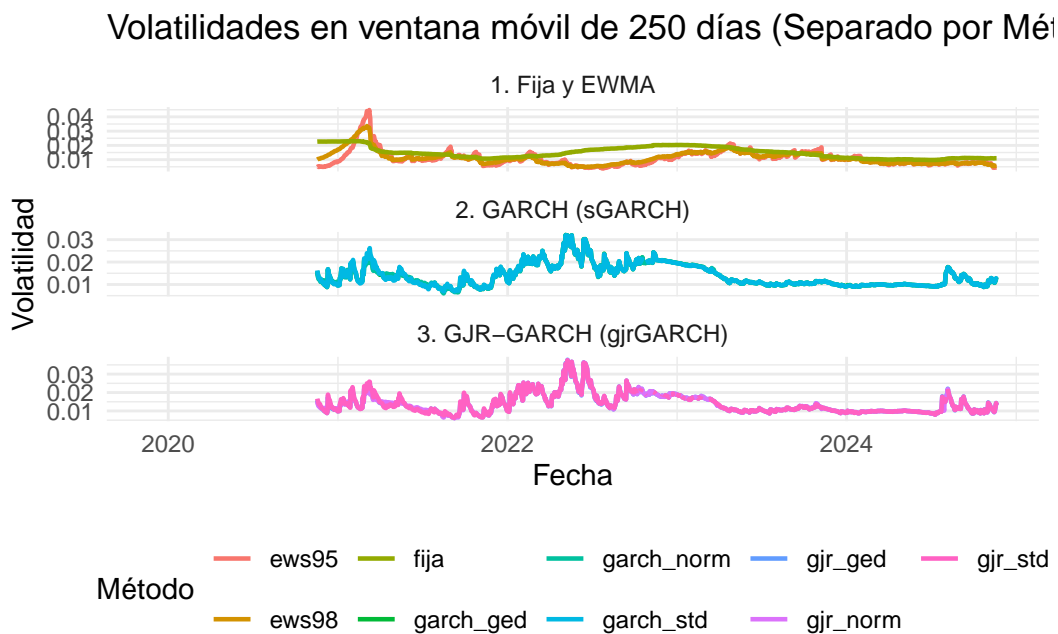
  fig2 <- plot_ly(volatilidades_movil, x = ~Date) %>%
    add_trace(y = ~garch_norm, name = 'GARCH (norm)', type = 'scatter', mode = 'lines', line = list(color = 'red'))
    add_trace(y = ~garch_std, name = 'GARCH (std)', type = 'scatter', mode = 'lines', line = list(color = 'blue'))
    add_trace(y = ~garch_ged, name = 'GARCH (ged)', type = 'scatter', mode = 'lines', line = list(color = 'green'))
    layout(title = '2. Volatilidades: GARCH (sGARCH)',
           xaxis = list(title = 'Fecha'),
           yaxis = list(title = 'Volatilidad'))

  fig3 <- plot_ly(volatilidades_movil, x = ~Date) %>%
    add_trace(y = ~gjr_norm, name = 'GJR-GARCH (norm)', type = 'scatter', mode = 'lines', line = list(color = 'red'))
    add_trace(y = ~gjr_std, name = 'GJR-GARCH (std)', type = 'scatter', mode = 'lines', line = list(color = 'blue'))
    add_trace(y = ~gjr_ged, name = 'GJR-GARCH (ged)', type = 'scatter', mode = 'lines', line = list(color = 'green'))
    layout(title = '3. Volatilidades: GJR-GARCH (gjrGARCH)',
           xaxis = list(title = 'Fecha'),
           yaxis = list(title = 'Volatilidad'))

  print(fig1)
  print(fig2)
  print(fig3)
}else{
  volatilidades_movil_long <- volatilidades_movil %>%
    pivot_longer(~Date, names_to = "Metodo", values_to = "Volatilidad") %>%
    mutate(
      Grupo = case_when(
        Metodo %in% c("fija", "ews95", "ews98") ~ "1. Fija y EWMA",
        Metodo %in% c("garch_norm", "garch_std", "garch_ged") ~ "2. GARCH (sGARCH)",
        Metodo %in% c("gjr_norm", "gjr_std", "gjr_ged") ~ "3. GJR-GARCH (gjrGARCH)",
        TRUE ~ "Otro"
      )
    )
}
```

```
ggplot(volatilidades_movil_long, aes(x = Date, y = Volatilidad, color = Metodo)) +
  geom_line(linewidth = 0.8) +
  labs(title = "Volatilidades en ventana móvil de 250 días (Separado por Método)",
       x = "Fecha",
       y = "Volatilidad",
       color = "Método") +
  facet_wrap(~ Grupo, scales = "free_y", ncol = 1) +
  theme_minimal() +
  theme(legend.position = "bottom")
}
```

Warning: Removed 2241 rows containing missing values or values outside the scale range (`geom_line()`).



Backtest (prueba de Kupiec)

Para finalizar, se realizará un backtest para evaluar el desempeño de las estimaciones de volatilidad en la predicción del VaR, esto mediante la prueba de Kupiec.

```

PL <- monto * rendimientos_decimal
columnas_sigma <- colnames(volatilidades_movil)[-1]
confianzas <- c(0.95, 0.98)
resultados_backtest <- data.frame()
for (conf in confianzas){
  alpha <- 1 - conf
  for(col in columnas_sigma){
    sigma_col <- volatilidades_movil[[col]]
    VaR_estimado <- VaR(monto, sigma_col, conf, "norm", 1, length(na.omit(rendimientos_decimal)))
    resultados_kupiec <- KupiecBackTest(PL, VaR_estimado, alpha)
    resultados_backtest <- rbind(resultados_backtest, data.frame(
      Metodo = col,
      Confianza = conf,
      excepciones = resultados_kupiec$Statistic,
      valor_crítico = resultados_kupiec$criticalValue,
      rango_min = resultados_kupiec$twoSidedCriticalValue[1],
      rango_max = resultados_kupiec$twoSidedCriticalValue[2],
      valor_esperado = resultados_kupiec$expectedExceeds,
      p_value = resultados_kupiec$pValue
    ))
  }
}

```

[1] "Estimating VaR..."

[1] "Estimating with normpdf VaR at 95% of confidence..."

[1] "Estimating VaR..."

[1] "Estimating with normpdf VaR at 95% of confidence..."

[1] "Estimating VaR..."

[1] "Estimating with normpdf VaR at 95% of confidence..."

[1] "Estimating VaR..."

[1] "Estimating with normpdf VaR at 95% of confidence..."

[1] "Estimating VaR..."

[1] "Estimating with normpdf VaR at 95% of confidence..."

[1] "Estimating VaR..."

[1] "Estimating with normpdf VaR at 95% of confidence..."

[1] "Estimating VaR..."

[1] "Estimating with normpdf VaR at 95% of confidence..."

[1] "Estimating VaR..."

[1] "Estimating with normpdf VaR at 95% of confidence..."

[1] "Estimating VaR..."

[1] "Estimating with normpdf VaR at 95% of confidence..."

[1] "Estimating VaR..."

[1] "Estimating with normpdf VaR at 98% of confidence..."

[1] "Estimating VaR..."

[1] "Estimating with normpdf VaR at 98% of confidence..."

[1] "Estimating VaR..."

[1] "Estimating with normpdf VaR at 98% of confidence..."

[1] "Estimating VaR..."

```
[1] "Estimating with normpdf VaR at 98% of confidence..."

[1] "Estimating VaR..."

[1] "Estimating with normpdf VaR at 98% of confidence..."

[1] "Estimating VaR..."

[1] "Estimating with normpdf VaR at 98% of confidence..."

[1] "Estimating VaR..."

[1] "Estimating with normpdf VaR at 98% of confidence..."

[1] "Estimating VaR..."

[1] "Estimating with normpdf VaR at 98% of confidence..."

[1] "Estimating VaR..."

[1] "Estimating with normpdf VaR at 98% of confidence..."
```

Resultados del backtest

```
if(knitr::is_html_output()){
  DT::datatable(resultados_backtest)
} else {
  knitr::kable(resultados_backtest)
}
```

Metodo	Confianza	excepciones	valor_crítico	rango_min	rango_max	valor_esperado	p_value
fija	0.95	57	62	37	64	50	0.1525153
ews95	0.95	112	62	37	64	50	0.0000000

Metodo	Confianza	excepciones	valor_crítico	rango_min	rango_max	valor_esperado	op_value
ews98	0.95	110	62	37	64	50	0.0000000
garch_norm	0.95	59	62	37	64	50	0.0965826
garch_std	0.95	58	62	37	64	50	0.1163048
garch_ged	0.95	58	62	37	64	50	0.1222398
gjr_norm	0.95	52	62	37	64	50	0.3733138
gjr_std	0.95	48	60	36	62	48	0.4852778
gjr_ged	0.95	51	62	37	64	50	0.4285490
fija	0.98	32	30	11	31	20	0.0048623
ews95	0.98	79	30	11	31	20	0.0000000
ews98	0.98	75	30	11	31	20	0.0000000
garch_norm	0.98	24	30	11	31	20	0.1635955
garch_std	0.98	24	30	11	31	20	0.1590192
garch_ged	0.98	24	30	11	31	20	0.1635955
gjr_norm	0.98	21	30	11	31	20	0.3691606
gjr_std	0.98	21	29	10	30	19	0.3018526
gjr_ged	0.98	21	30	11	31	20	0.3691606

Como podemos observar en la tabla anterior, los métodos de suavizamiento exponencial fallan en la prueba de Kupiec, y la volatilidad fija funciona al 95%, pero no al 98%. Los modelos GARCH pasan todas las pruebas con todas las distribuciones, lo que podría justificar su uso incluso con su mayor complejidad computacional.

Conclusiones

En este ensayo se analizó, usando distintos métodos, la estimación de volatilidades para el EFT QQQ en pesos mexicanos. Se observaron las ventajas y desventajas de cada método, así como su desempeño en una prueba de backtest mediante la prueba de Kupiec. Como era de esperarse, los modelos GARCH demostraron ser los mejores para capturar la volatilidad dinámica; sin embargo, si bien falló una de las pruebas, es sorprendente que la volatilidad fija, siendo el método más simple, funciona correctamente en una y se aproxima bastante en la otra, lo que a pesar de sus limitaciones lo hace una opción para análisis rápidos o cuando se sospecha poca variabilidad.