

Modelo GARCH(1,1).

Introducción y aplicación.

Dr. Martín Lozano <https://mlozanoqf.github.io/>

09 de noviembre de 2025, 05:19 p.m.

	Fundamental	Intermedio	Especializado
Finanzas	×	✓	×
Estadística	×	✓	×
R	×	✓	×

1 Introducción.

- Estimación de la volatilidad condicional del índice S&P 500 mediante un modelo GARCH(1,1).
- Aplicar dos estrategias de estimación: un ajuste por máxima verosimilitud completa (Full MLE) y un método con *variance targeting*.
- Ejemplo práctico usando datos del S&P 500.

2 Paquetes.

```
1 library(tidyquant)
2 library(dplyr)
3 library(knitr)
4 library(ggplot2)
5 library(tidyr)
6 library(scales)
```

3 Rendimientos.

$$u_i = \frac{S_i - S_{i-1}}{S_{i-1}} \rightarrow u_i = \frac{S_i}{S_{i-1}} - 1, u_2 = \frac{2099.60}{2076.62} - 1 \rightarrow u_2 = 0.0110660492.$$

```
1 S <- tq_get("GSPC", from = "2015-07-10", to = "2020-07-10") |>
2   dplyr::select(date, S = close) |>
3   mutate(i = dplyr::row_number(), u_i = (S / lag(S)) - 1)
4
5 kable(rbind(head(S, 6), tail(S, 2)), digits = 10, format.args = list(scientific = FALSE))
```

date	S	i	u_i
2015-07-10	2076.62	1	NA
2015-07-13	2099.60	2	0.0110660492
2015-07-14	2108.95	3	0.0044531592
2015-07-15	2107.40	4	-0.0007349861
2015-07-16	2124.29	5	0.0080146804
2015-07-17	2126.64	6	0.0011061830
2020-07-08	3169.94	1258	0.0078274619
2020-07-09	3152.05	1259	-0.0056436062

4 Inicialización.

$$\sigma_n^2 = \omega + \alpha u_{n-1}^2 + \beta \sigma_{n-1}^2.$$

$$v_i = \sigma_i^2.$$

```
1 garch_variance_shifted <- function(u, omega, alpha, beta) {
2   n <- length(u)
3   v <- rep(NA_real_, n) # inicializa el vector de varianzas condicionales v.
4   if (n >= 2) v[2] <- u[1]^2 # valor inicial de v.
5   if (n >= 3) { # a partir de aquí se usa la ecuación de GARCH(1,1).
6     for (i in 3:n) {
7       v[i] <- omega + alpha * u[i - 1]^2 + beta * v[i - 1]
8     }
9   }
10  v
11 }
```

5 Estimación Full MLE.

$$\sigma_n^2 = \omega + \alpha u_{n-1}^2 + \beta \sigma_{n-1}^2, v_i = \sigma_i^2, \text{log-verosimilitud condicional: } \sum_{i=1}^m \left[-\ln(v_i) - \frac{u_i^2}{v_i} \right].$$

```

1 u <- S$u_i |> stats::na.omit()
2
3 nll_garch_shifted <- function(par, u) {
4   omega <- par[1]; alpha <- par[2]; beta <- par[3]
5
6   if (omega <= 0 || alpha < 0 || beta < 0 || (alpha + beta) >= 1)
7     return(1e12)
8   v <- garch_variance_shifted(u, omega, alpha, beta)
9   mask <- !is.na(v) & v > 0 # identifica entradas válidas y positivas.
10  if (!any(mask)) return(1e12)
11  -sum( -log(v[mask]) - (u[mask]^2) / v[mask] )} # neg-log-likelihood Gaussiana.
12
13 start <- c(omega = 4e-6, alpha = 0.2, beta = 0.7)
14 # Broyden-Fletcher-Goldfarb-Shanno que usa memoria limitada y cotas de los parámetros.
15 fit <- optim(par = start, fn = nll_garch_shifted, u = u,
16               method = "L-BFGS-B", lower = c(1e-12, 0, 0),
17               upper = c(Inf, 1 - 1e-6, 1 - 1e-6)) # maximiza vía optim.
18
19 theta_hat <- fit$par
20 omega_hat <- theta_hat[1]; alpha_hat <- theta_hat[2]; beta_hat <- theta_hat[3]
21
22 v_hat <- garch_variance_shifted(u, omega_hat, alpha_hat, beta_hat)
23 mask <- !is.na(v_hat) & v_hat > 0
24 ell_i <- -log(v_hat[mask]) - (u[mask]^2) / v_hat[mask]
25 ll_tot <- sum(ell_i)
26
27 n_obs <- sum(mask) # 1257.
28 k <- 3 # estimamos 3 parámetros.
29 AIC <- -2 * ll_tot + 2 * k
30 BIC <- -2 * ll_tot + log(n_obs) * k
31
32 summary_tbl <-
33   tibble::tibble(metric = c("omega", "alpha", "beta", "logLik", "n_obs", "AIC", "BIC"),
34                 value = c(theta_hat, ll_tot, n_obs, AIC, BIC))
35
36 knitr::kable(summary_tbl, align = c("l", "r"), digits = c(0, 6), format.args = list(scientific
+   = FALSE), col.names = c("Estimador / métrica", "Valor"))

```

Estimador / métrica	Valor
omega	0.000003
alpha	0.229733
beta	0.758404
logLik	10835.375594
n_obs	1257.000000
AIC	-21664.751188
BIC	-21649.341739

6 Estimación *variance targeting*.

$$\sigma_n^2 = \omega + \alpha u_{n-1}^2 + \beta \sigma_{n-1}^2, \omega = V_L(1 - \alpha - \beta), v_i = \sigma_i^2.$$

La función de log-verosimilitud condicional Gaussiana: $\sum_{i=1}^m \left[-\ln(v_i) - \frac{u_i^2}{v_i} \right]$.

```

1 VL <- var(u, na.rm = TRUE)
2
3 nll_garch_VT_shifted <- function(par, u, VL) {
4   alpha <- par[1]; beta <- par[2]
5   if (alpha < 0 || beta < 0 || (alpha + beta) >= 1) return(1e12)
6   omega <- VL * (1 - alpha - beta)
7   v <- garch_variance_shifted(u, omega, alpha, beta)
8   mask <- !is.na(v) & v > 0
9   if (!any(mask)) return(1e12)
10  -sum( -log(v[mask]) - (u[mask]^2) / v[mask] )}
```

11

```

12 fit_vt <- optim(par = c(alpha = 0.2, beta = 0.7), fn = nll_garch_VT_shifted,
13   u = u, VL = VL, method = "L-BFGS-B", lower = c(0, 0),
14   upper = c(1 - 1e-6, 1 - 1e-6))
15
16 alpha_vt <- fit_vt$par[1]
17 beta_vt <- fit_vt$par[2]
18 omega_vt <- VL * (1 - alpha_vt - beta_vt)
19
20 v_vt <- garch_variance_shifted(u, omega_vt, alpha_vt, beta_vt)
21 mask_vt <- !is.na(v_vt) & v_vt > 0
22 ell_vt <- -log(v_vt[mask_vt]) - (u[mask_vt]^2) / v_vt[mask_vt]
23 ll_vt <- sum(ell_vt)
24
25 n_obs_vt <- sum(mask_vt) # 1257.
26 k_vt <- 2
27 AIC_vt <- -2 * ll_vt + 2 * k_vt
28 BIC_vt <- -2 * ll_vt + log(n_obs_vt) * k_vt
29
30 summary_vt <- tibble::tibble(
31   metric = c("omega (target)", "alpha", "beta", "logLik", "n_obs", "AIC", "BIC"), value =
32   c(omega_vt, alpha_vt, beta_vt, ll_vt, n_obs_vt, AIC_vt, BIC_vt))
33 knitr::kable(summary_vt, align = c("l", "r"), digits = c(0, 6), format.args = list(scientific
34   = FALSE), col.names = c("Estimador / métrica", "Valor"))
```

Estimador / métrica	Valor
omega (target)	0.000004
alpha	0.226349
beta	0.747038
logLik	10837.404644
n_obs	1257.000000
AIC	-21670.809288
BIC	-21660.536321

7 MLE versus *variance targeting*.

```
1 comparison_tbl <- dplyr::bind_rows(
2   summary_tbl |> dplyr::mutate(model = "Full MLE"),
3   summary_vt |> dplyr::mutate(
4     metric = dplyr::if_else(metric == "omega (target)", "omega", metric),
5     model = "Variance targeting")) |>
6   tidyr::pivot_wider(names_from = model, values_from = value) |>
7   dplyr::select(metric, `Full MLE`, `Variance targeting`)
8
9 knitr::kable(comparison_tbl,
10   col.names = c("Estimador / métrica", "Full MLE", "Variance targeting"),
11   align = c("l", "r", "r"), digits = c(0, 6, 6),
12   format.args = list(scientific = FALSE))
```

Estimador / métrica	Full MLE	Variance targeting
omega	0.000003	0.000004
alpha	0.229733	0.226349
beta	0.758404	0.747038
logLik	10835.375594	10837.404644
n_obs	1257.000000	1257.000000
AIC	-21664.751188	-21670.809288
BIC	-21649.341739	-21660.536321

8 Resultado.

$$\sigma_n^2 = \omega + \alpha u_{n-1}^2 + \beta \sigma_{n-1}^2, \omega = V_L(1 - \alpha - \beta), v_i = \sigma_i^2.$$

```

1 omega <- omega_vt
2 alpha <- alpha_vt
3 beta  <- beta_vt
4
5 S <- S |>
6   mutate(v_i = lag(u_i^2))
7
8 for (i in 2:nrow(S)) {
9   if (!is.na(S$u_i[i - 1]) && !is.na(S$v_i[i - 1])) {
10     S$v_i[i] <- omega + alpha * (S$u_i[i - 1]^2) + beta * S$v_i[i - 1]
11   }
12 }
13
14 kable(rbind(head(S, 6), tail(S, 2)), digits = 10, format.args = list(scientific = FALSE))

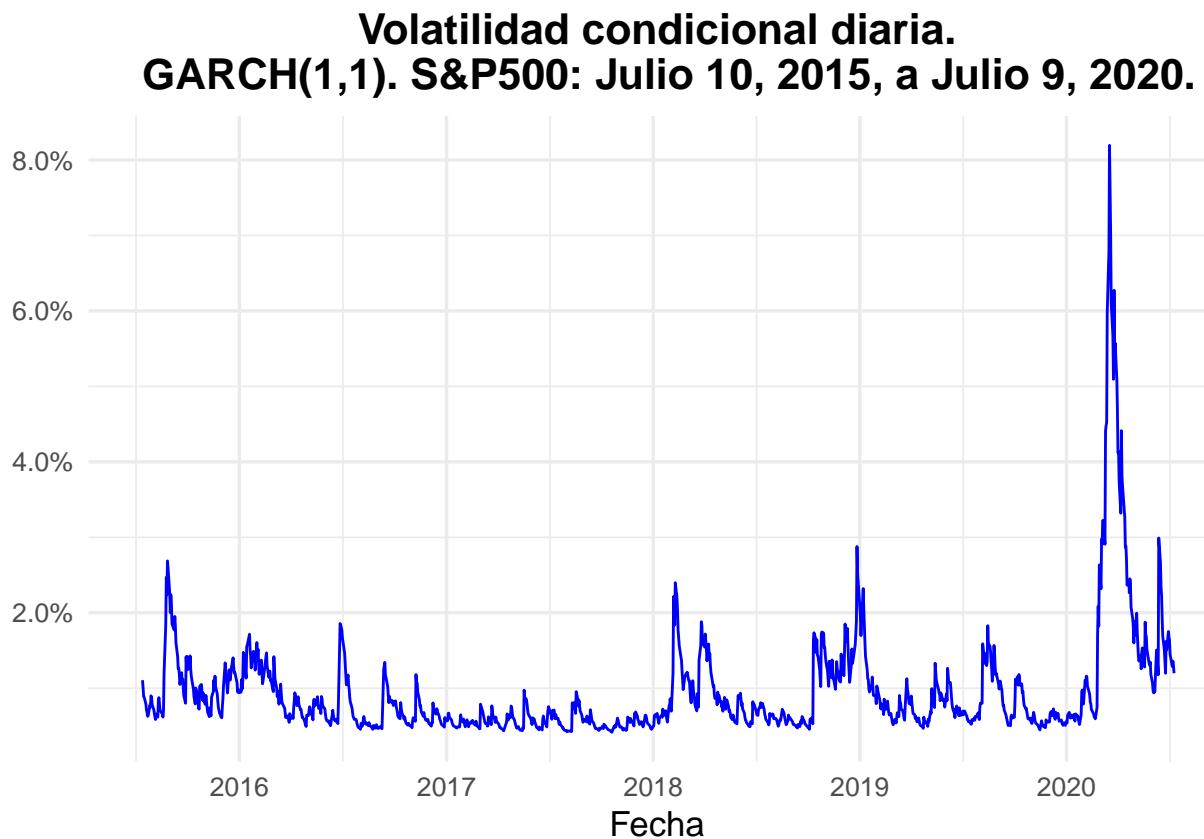
```

date	S	i	u_i	v_i
2015-07-10	2076.62	1	NA	NA
2015-07-13	2099.60	2	0.0110660492	NA
2015-07-14	2108.95	3	0.0044531592	0.0001224574
2015-07-15	2107.40	4	-0.0007349861	0.0000999353
2015-07-16	2124.29	5	0.0080146804	0.0000787441
2015-07-17	2126.64	6	0.0011061830	0.0000773307
2020-07-08	3169.94	1258	0.0078274619	0.0001684868
2020-07-09	3152.05	1259	-0.0056436062	0.0001437005

9 Gráfico.

$$\sigma_n^2 = \omega + \alpha u_{n-1}^2 + \beta \sigma_{n-1}^2.$$

```
1 ggplot(S, aes(x = date, y = 100*v_i^.5)) +
2   geom_line(color = "blue") +
3   labs(title = "Volatilidad condicional diaria.
4   GARCH(1,1). S&P500: Julio 10, 2015, a Julio 9, 2020.",
5   y = NULL, x = "Fecha") +
6   scale_y_continuous(labels = scales::label_percent(accuracy = 0.1, scale = 1)) +
7   theme_minimal(base_size = 13) +
8   theme(plot.title = element_text(face = "bold", hjust = 0.5),
9         axis.title.y = element_text(angle = 0, vjust = 0.5))
```



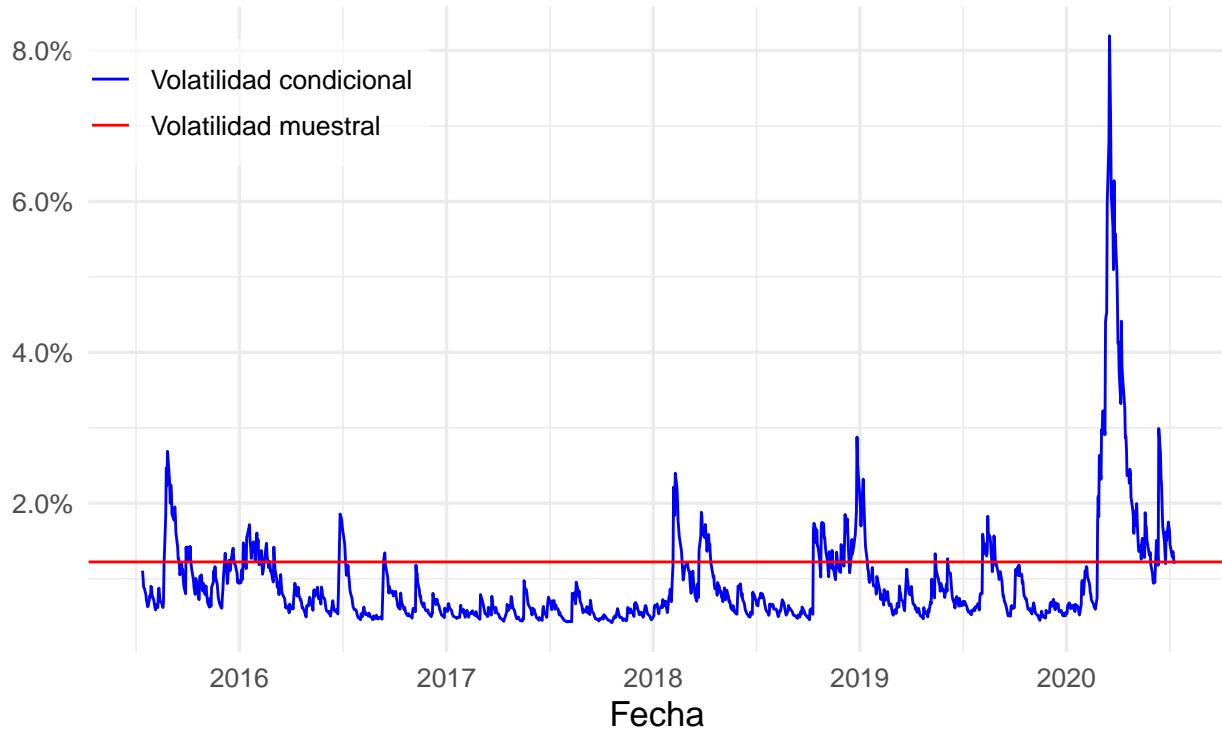
10 Comparación.

Azul: $\sigma_n^2 = \omega + \alpha u_{n-1}^2 + \beta \sigma_{n-1}^2$.

Rojo: $s = \frac{1}{n-1} \sum_{i=1}^n (u_i - \bar{u})^2$.

```
1 ggplot(S, aes(x = date, y = 100 * v_i^.5, colour = "Volatilidad condicional")) +
2   geom_line() +
3   geom_hline(aes(yintercept = 1.221865, colour = "Volatilidad muestral")) +
4   scale_colour_manual(
5     values = c("Volatilidad condicional" = "blue", "Volatilidad muestral" = "red"),
6     name = NULL) +
7   labs(
8     title = "Volatilidad condicional y muestral diaria.\nGARCH(1,1). S&P500: Julio 10, 2015,
9       a Julio 9, 2020.",
9     y = NULL, x = "Fecha") +
10  scale_y_continuous(labels = scales::label_percent(accuracy = 0.1, scale = 1)) +
11  theme_minimal(base_size = 13) +
12  theme(
13    plot.title = element_text(face = "bold", hjust = 0.5),
14    axis.title.y = element_text(angle = 0, vjust = 0.5),
15    legend.position = c(0.3, 0.95),
16    legend.justification = c("right", "top"),
17    legend.background = element_rect(fill = scales::alpha("white", 0.6), colour = NA))
```

**Volatilidad condicional y muestral diaria.
GARCH(1,1). S&P500: Julio 10, 2015, a Julio 9, 2020.**



11 Conclusión.

- El modelo *variance targeting* presenta mayor log-verosimilitud y menores valores de AIC y BIC.
- Por tanto, se selecciona el GARCH(1,1) con *variance targeting*.