

Un robo-trader simple.

Construcción y evaluación de su desempeño.

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

21 de diciembre de 2025, 03:11 p.m.

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

1 Introducción.

- Este documento ilustra la construcción, implementación y evaluación de un robo-trader sencillo para AAPL.
- La señal de trading se basa en cruces del precio de cierre con un promedio móvil simple de 3 días (MA3), generando órdenes buy/sell.
- El desempeño se compara contra benchmarks y contra estrategias alternativas.
- Se reportan ganancia simple, número de operaciones y se documenta el proceso mediante gráficos y bitácoras de operaciones.

2 Paquetes.

```
1 library(tidyquant)
2 library(dplyr)
3 library(lubridate)
4 library(ggplot2)
5 library(tidyr)
6 library(purrr)
7 library(knitr)
```

3 Inicialización 1.

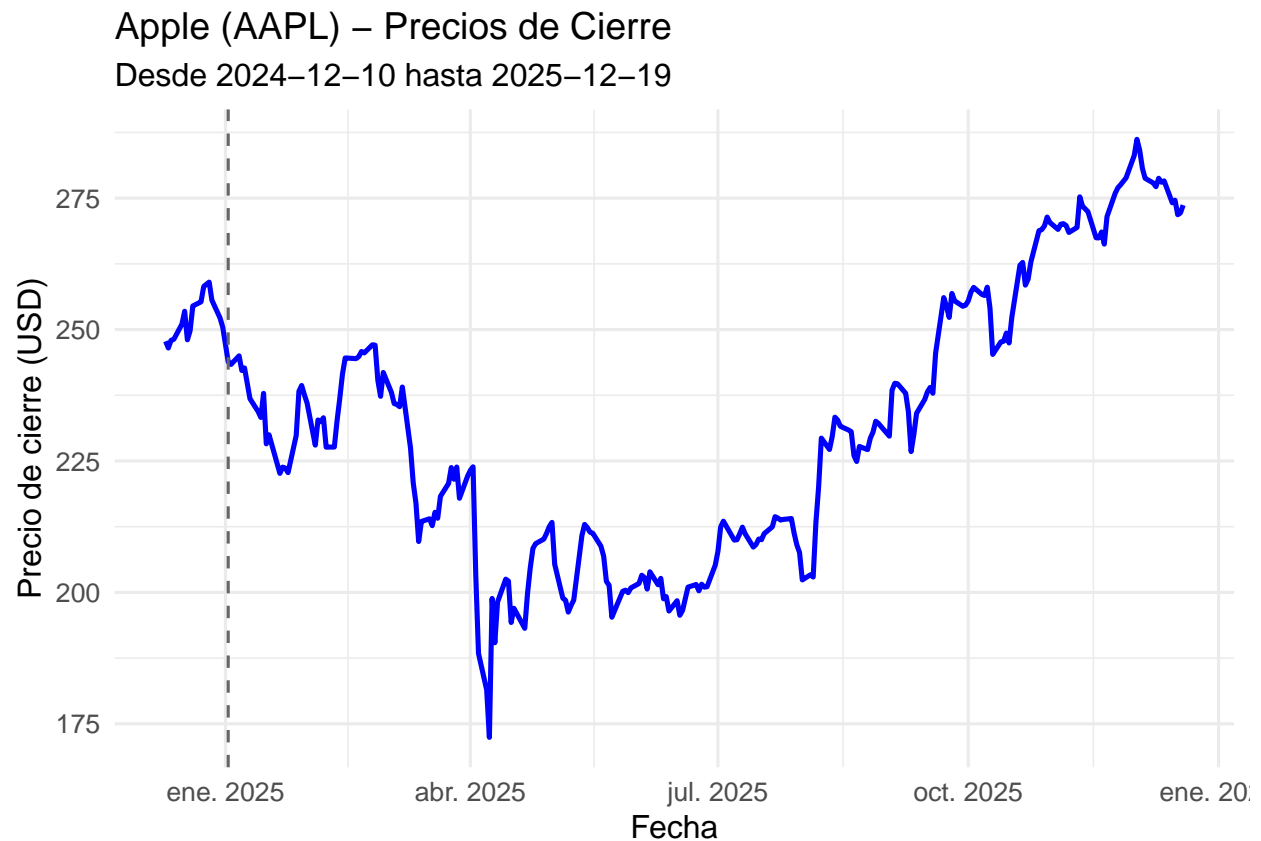
```
1 raw_prices <- tq_get("AAPL", from = "2024-11-01", to = "2025-12-31") |>
2   arrange(date)
3
4 prices_ma3 <- raw_prices |>
5   tq_mutate(select = close, mutate_fun = SMA, n = 3, col_rename = "ma3")
6
7 start_date <- prices_ma3 |>
8   filter(date >= as.Date("2025-01-02"), !is.na(ma3)) |>
9   slice_head(n = 1) |>
10  pull(date)
11
12 final_date <- prices_ma3 |>
13   filter(date <= as.Date("2025-12-19")) |>
14   slice_tail(n = 1) |>
15   pull(date)
16
17 rt_data <- prices_ma3 |>
18   filter(date >= start_date - days(1), date <= final_date) |>
19   mutate(sign_diff = sign(close - ma3),
20          lag_sign = lag(sign_diff),
21          cross = !is.na(sign_diff) & !is.na(lag_sign) & sign_diff != lag_sign,
22          signal = case_when(
23            cross & sign_diff > lag_sign ~ "buy",
24            cross & sign_diff < lag_sign ~ "sell",
25            TRUE ~ "hold"))
26
27 signals_rt <- rt_data |>
28   filter(signal %in% c("buy", "sell"), date >= start_date) |>
29   transmute(date, signal, price = close) |>
30   arrange(date)
31
32 signals_eval <- bind_rows(signals_rt,
33   tibble(date = final_date, signal = "final_close",
34     price = rt_data$close[rt_data$date == final_date][1])) |>
35   arrange(date)
36
37 signal_outcomes <- signals_eval |>
38   mutate(next_price = lead(price), next_signal = lead(signal)) |>
39   filter(signal %in% c("buy", "sell")) |>
40   mutate(trade_pnl = if_else(signal == "buy", next_price - price, price - next_price),
41          correct = trade_pnl > 0)
42
43 signals_omniscient <- signal_outcomes |> filter(correct) |>
44   select(date, signal, price)
45 signals_dumb <- signal_outcomes |> filter(!correct) |>
46   select(date, signal, price)
```

4 Inicialización 2.

```
1 # Helpers
2 simulate_pnl <- function(signals_df, final_price) {
3   pnl <- 0; position <- "flat"; entry <- NA_real_
4   for (i in seq_len(nrow(signals_df))) {
5     sig <- signals_df$signal[i]; px <- signals_df$price[i]
6     if (position == "flat") {
7       position <- if (sig == "buy") "long" else "short"; entry <- px
8     } else if (sig == "buy" && position == "short") {
9       pnl <- pnl + (entry - px); position <- "long"; entry <- px
10    } else if (sig == "sell" && position == "long") {
11      pnl <- pnl + (px - entry); position <- "short"; entry <- px
12    } else if (sig == position) {
13      pnl <- pnl + if (position == "long") (px - entry) else (entry - px)
14      entry <- px
15    }
16  }
17  if (!is.na(final_price) && position != "flat") {
18    pnl <- pnl + if (position == "long") (final_price - entry) else (entry - final_price)}
19  pnl}
20
21 make_bitacora <- function(signals_df, final_date, final_price) {
22   signals_eval <- bind_rows(signals_df |> arrange(date),
23                             tibble(date = final_date, signal = "final_close",
24                                     price = final_price)) |>
25   arrange(date)
26   position <- "flat"; entry <- NA_real_; pnl_acum <- 0; logs <- list()
27   for (i in seq_len(nrow(signals_eval))) {
28     sig <- signals_eval$signal[i]; px <- signals_eval$price[i]
29     if (position == "flat") {
30       if (sig %in% c("buy", "sell")) {
31         flujo <- if (sig == "buy") -px else px
32         logs[[length(logs) + 1]] <- tibble(date = signals_eval$date[i], 'acción' = sig,
33                                           price = px, flujo = flujo,
34                                           PnL = 0, PnL_acum = pnl_acum)
35       }
36       position <- if (sig == "buy") "long" else "short"; entry <- px
37     } else if (sig %in% c("buy", "sell", "final_close")) {
38       pnl <- if (position == "long") px - entry else entry - px
39       flujo <- if (sig == "buy") -px else if (sig == "sell") px else 0
40       pnl_acum <- pnl_acum + pnl
41       logs[[length(logs) + 1]] <- tibble(date = signals_eval$date[i], 'acción' = sig,
42                                         price = px, flujo = flujo,
43                                         PnL = pnl, PnL_acum = pnl_acum)
44       if (sig == "final_close") { position <- "flat"; entry <- NA_real_ }
45       else { position <- if (sig == "buy") "long" else "short"; entry <- px }
46     }
47   }
48   bind_rows(logs)
49 }
50
51 show_head_tail <- \(df, n = 5) bind_rows(slice_head(df, n = n), slice_tail(df, n = n))
```

5 Serie de tiempo.

```
1 last15_2024 <- prices_ma3 |>
2   filter(year(date) == 2024) |>
3   arrange(date) |>
4   tail(15)
5
6 start_date_plot <- min(last15_2024$date)
7 end_date <- as.Date("2025-12-19")
8
9 aapl_window <- prices_ma3 |>
10  filter(date >= start_date_plot, date <= end_date)
11
12 ggplot(aapl_window, aes(x = date, y = close)) +
13   geom_line(color = "blue", linewidth = 0.9) +
14   geom_vline(xintercept = as.Date("2025-01-02"), linetype = "dashed", color = "gray40") +
15   labs(title = "Apple (AAPL) – Precios de Cierre",
16        subtitle = paste("Desde", start_date_plot, "hasta", end_date),
17        x = "Fecha", y = "Precio de cierre (USD)") +
18   theme_minimal(base_size = 12)
```



6 Benchmarks.

```
1 first_2025 <- prices_ma3 |> filter(date >= as.Date("2025-01-01")) |>
2   slice_head(n = 1)
3 buy_hold_buy_date <- first_2025$date
4 buy_hold_buy_px <- first_2025$close
5
6 sell_12dec <- prices_ma3 |> filter(date <= as.Date("2025-12-12")) |>
7   slice_tail(n = 1)
8 buy_hold_sell_date <- sell_12dec$date
9 buy_hold_sell_px <- sell_12dec$close
10
11 min_row <- prices_ma3 |>
12   filter(date >= as.Date("2025-01-01"), date <= as.Date("2025-12-19")) |>
13   slice_min(close, n = 1, with_ties = FALSE)
14 min_date <- min_row$date
15 min_px <- min_row$close
16
17 final_row <- prices_ma3 |> filter(date <= as.Date("2025-12-19")) |>
18   slice_tail(n = 1)
19 final_date <- final_row$date
20 final_px <- final_row$close
21
22 buy_hold_gain <- buy_hold_sell_px - buy_hold_buy_px
23 trampa_gain <- (buy_hold_buy_px - min_px) + (final_px - min_px)
24
25 strategies <- tibble(estrategia = c("Buy & hold", "Trampa"),
26   ganancia_simple = c(buy_hold_gain, trampa_gain))
27
28 strategies
```

```
## # A tibble: 2 x 2
##   estrategia ganancia_simple
##   <chr>          <dbl>
## 1 Buy & hold      34.4
## 2 Trampa         173.
```

7 La trampa.

```
1 P0 <- buy_hold_buy_px
2 Pmin <- min_px
3 Pend <- final_px
4
5 tramo_corto <- P0 - Pmin
6 tramo_largo <- Pend - Pmin
7 gan_trampa <- tramo_corto + tramo_largo
8
9 tibble(P0, Pmin, Pend,
10        tramo_corto = tramo_corto,
11        tramo_largo = tramo_largo,
12        ganancia_trampa = gan_trampa) |>
13 mutate(across(everything(), ~ round(.x, 2))) |>
14 kable()
```

P0	Pmin	Pend	tramo_corto	tramo_largo	ganancia_trampa
243.85	172.42	273.67	71.43	101.25	172.68

Trampa: $P_0 = 243.85$ (cierres inicial), $P_{min} = 172.42$ (mínimo anual), $P_{end} = 273.67$ (cierres final). El tramo corto gana $243.85 - 172.42 = 71.43$; el tramo largo gana $273.67 - 172.42 = 101.25$. La ganancia total de Trampa es $71.43 + 101.25 = 172.68$.

8 Promedio móvil.

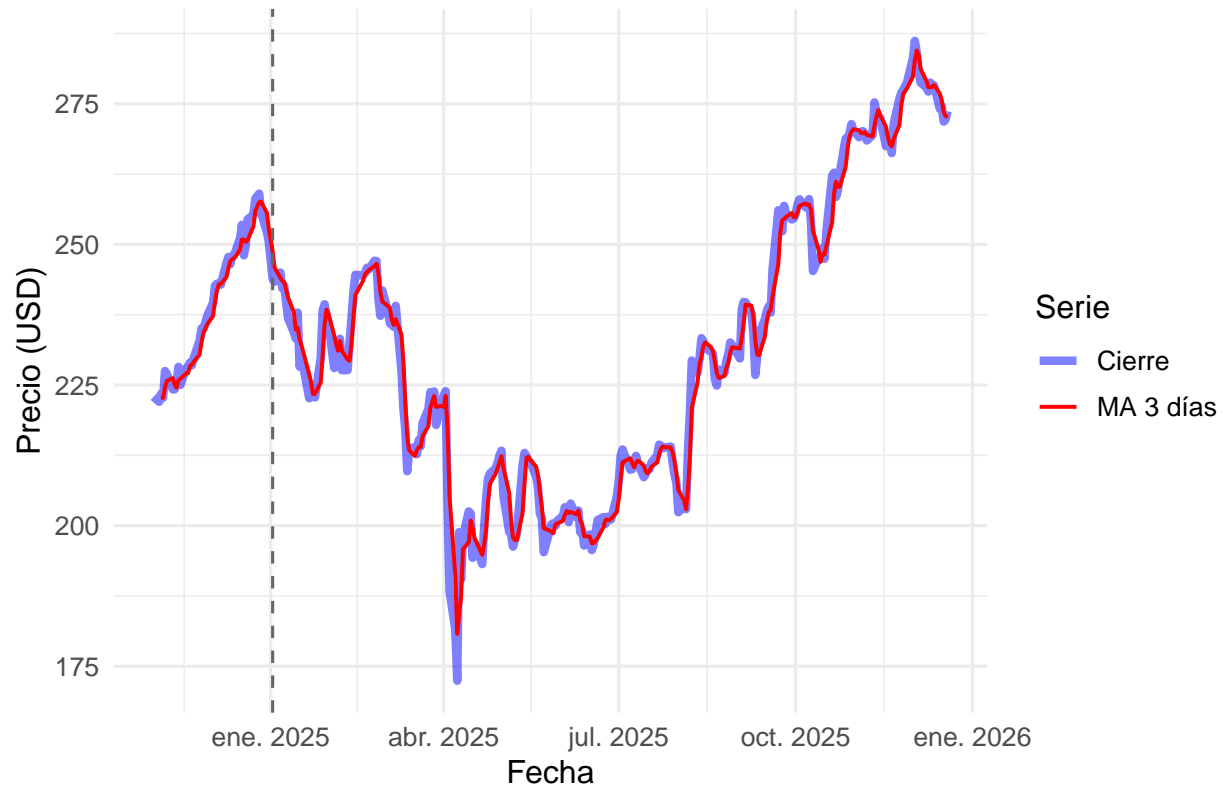
El robo-trader usa un promedio móvil simple (SMA) de 3 días sobre el precio de cierre. Sea P_t el precio de cierre en el día t . El promedio móvil simple de 3 días en t es $SMA3_t = \frac{1}{3}(P_t + P_{t-1} + P_{t-2})$, definido cuando existen los tres cierres previos, por eso los primeros 2 días resultan en NA.

La lógica de señal es: cruce de precio vs. SMA3.

- Señal de compra en t : cuando $\text{sign}(P_t - SMA3_t)$ cambia de negativo a positivo. El precio cruza desde abajo la SMA.
- Señal de venta en t : cuando $\text{sign}(P_t - SMA3_t)$ cambia de positivo a negativo. El precio cruza desde arriba la SMA.
- Si no cambia el signo, no hay señal, se mantiene “hold”.

```
1 ggplot(prices_ma3, aes(date)) +  
2   geom_line(aes(y = close, color = "Cierre"), linewidth = 1.5, alpha = 0.5) +  
3   geom_line(aes(y = ma3, color = "MA 3 días"), linewidth = 0.7, na.rm = TRUE) +  
4   geom_vline(xintercept = as.Date("2025-01-02"), linetype = "dashed", color = "gray40") +  
5   scale_color_manual(name = "Serie",  
6                       values = c("Cierre" = "blue", "MA 3 días" = "red")) +  
7   labs(title = "AAPL: Cierre vs. Promedio Móvil 3 días",  
8         x = "Fecha", y = "Precio (USD)") +  
9   theme_minimal(base_size = 12)
```

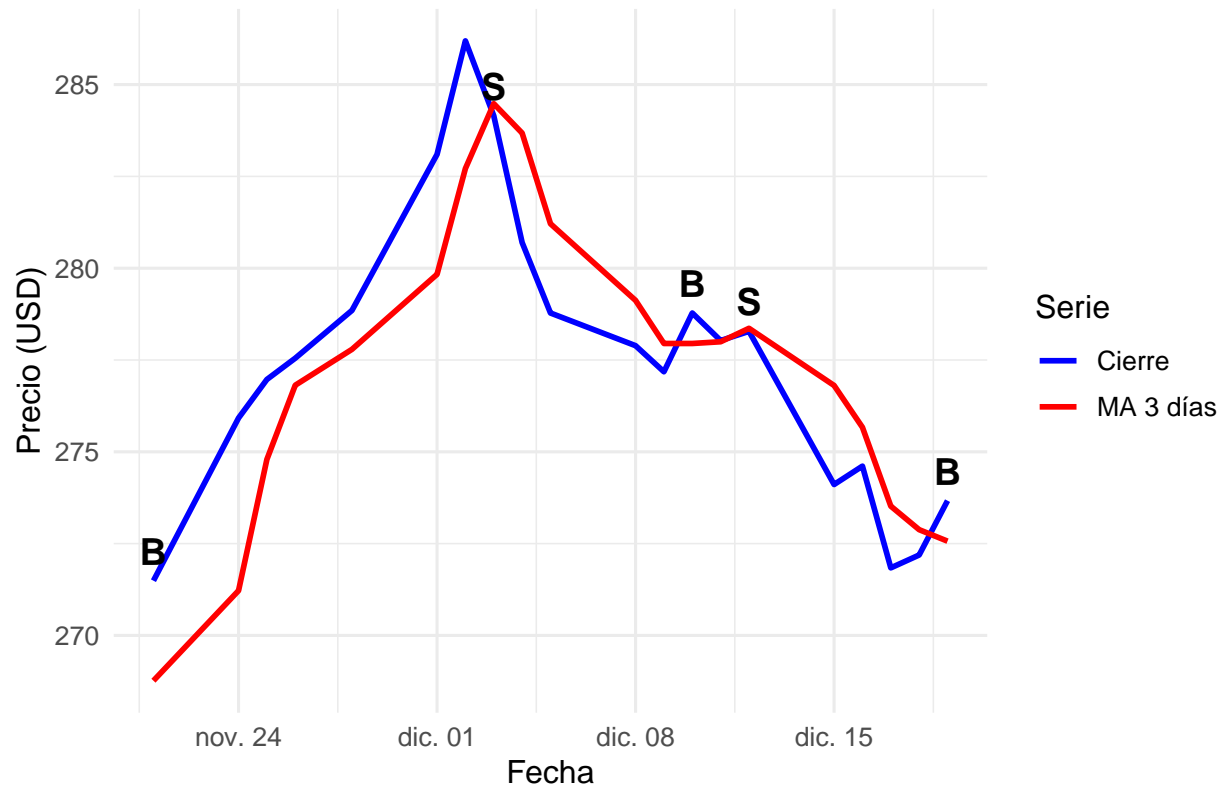
AAPL: Cierre vs. Promedio Móvil 3 días



9 Promedio móvil, zoom.

```
1 # Ventana del zoom
2 start_zoom <- as.Date("2025-11-21")
3 end_zoom   <- as.Date("2025-12-19")
4
5 # Señales buy/sell por cruce precio vs SMA3
6 signals_short <- prices_ma3 |>
7   mutate(sign_diff = sign(close - ma3), lag_sign = lag(sign_diff),
8     cross = !is.na(sign_diff) & !is.na(lag_sign) & sign_diff != lag_sign,
9     signal = case_when(
10       cross & sign_diff > lag_sign ~ "buy",
11       cross & sign_diff < lag_sign ~ "sell",
12       TRUE ~ NA_character_)) |>
13   filter(date >= start_zoom, date <= end_zoom, !is.na(signal)) |>
14   transmute(date, signal, price = close)
15
16 # Serie y etiquetas B/S
17 aapl_ma_short <- prices_ma3 |>
18   filter(date >= start_zoom, date <= end_zoom)
19
20 ggplot(aapl_ma_short, aes(date)) +
21   geom_line(aes(y = close, color = "Cierre"), linewidth = 1) +
22   geom_line(aes(y = ma3, color = "MA 3 días"), linewidth = 1, na.rm = TRUE) +
23   geom_text(data = signals_short,
24     aes(y = price, label = if_else(signal == "buy", "B", "S")),
25     vjust = -0.6, fontface = "bold", color = "black", size = 5) +
26   scale_color_manual(name = "Serie",
27     values = c("Cierre" = "blue", "MA 3 días" = "red")) +
28   labs(title = "AAPL: Cierre vs. MA 3 días (nov-dic 2025)",
29     x = "Fecha", y = "Precio (USD)") +
30   theme_minimal(base_size = 12)
```

AAPL: Cierre vs. MA 3 días (nov–dic 2025)



10 Descripción de 6 estrategias.

- Buy & hold: compras AAPL al primer cierre de 2025 y vendes el 12/dic/2025; PnL = diferencia entre esos cierres, sin trades intermedios.
- Trampa (short + long): abres corto al primer cierre de 2025, cubres y giras a largo en el mínimo anual (hasta 19/dic/2025) y cierras largo al final; capturas la caída y luego el rebote.
- Robo-trader (MA3): opera cada cruce cierre vs SMA 3d; cambia de lado en cada señal y liquida al cierre final; PnL suma cada tramo entre señales consecutivas.
- Aleatoria: 83 señales buy/sell alternadas en fechas al azar; PnL con la misma regla de cerrar y reabrir en cada señal, cierre forzado al final.
- Robo-omnisciente: aplica la regla de cruces pero solo ejecuta las señales que resultan ganadoras; si salen dos señales iguales seguidas, se cierra y reabre al nuevo precio, por eso el PnL refleja esos tramos y está bien calculado aun con sell-sell o buy-buy.
- Robo-dumb: igual lógica pero solo con señales perdedoras; también cierra y reabre en señales iguales consecutivas, de modo que el PnL recoge correctamente esos tramos negativos.

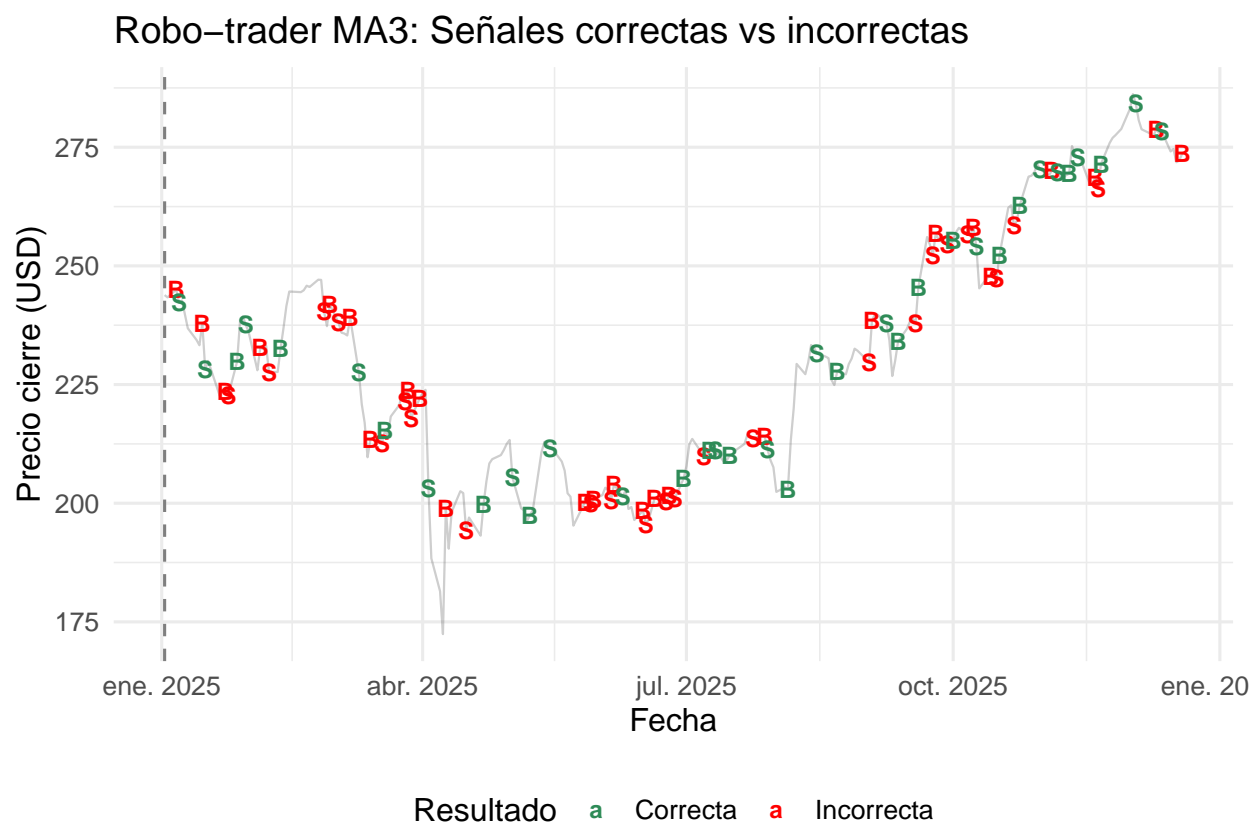
11 Desempeño de las 6 estrategias.

```
1 set.seed(123)
2 prices <- prices_ma3
3
4 P0 <- prices |> filter(date >= as.Date("2025-01-01")) |> slice_head(n = 1) |>
5   pull(close)
6 bh_sell_px <- prices |> filter(date <= as.Date("2025-12-12")) |> slice_tail(n = 1) |>
7   pull(close)
8 Pmin <- prices |> filter(date >= as.Date("2025-01-01"), date <= final_date) |>
9   slice_min(close, n = 1, with_ties = FALSE) |> pull(close)
10 Pend <- prices |> filter(date == final_date) |> pull(close)
11
12 buy_hold_gain <- bh_sell_px - P0
13 trampa_gain <- (P0 - Pmin) + (Pend - Pmin)
14
15 trading_days <- prices |>
16   filter(date >= start_date, date <= final_date) |>
17   pull(date)
18
19 random_dates <- sort(sample(trading_days, size = 83, replace = FALSE))
20 start_side <- sample(c("buy", "sell"), 1)
21 random_signals_vec <- if (start_side == "buy") rep(c("buy", "sell"), length.out = 83) else
22   rep(c("sell", "buy"), length.out = 83)
23
24 signals_rand <- tibble(date = random_dates,
25   signal = random_signals_vec) |>
26   left_join(prices |> select(date, close), by = "date") |>
27   rename(price = close) |>
28   arrange(date)
29
30 # Reuso de signal_outcomes ya calculado arriba
31 robo_trader_gain <- simulate_pnl(signals_rt, final_price = Pend)
32 random_gain <- simulate_pnl(signals_rand, final_price = Pend)
33 robo_omnisciente_gain <- simulate_pnl(signals_omniscient, final_price = Pend)
34 robo_dumb_gain <- simulate_pnl(signals_dumb, final_price = Pend)
35
36 rt_trades <- nrow(signals_rt)
37 rand_trades <- nrow(signals_rand)
38 omn_trades <- nrow(signals_omniscient)
39 dumb_trades <- nrow(signals_dumb)
40
41 strategies <- tibble(estrategia = c("Buy & hold", "Trampa", "Robo-trader",
42   "Aleatoria", "Robo-omnisciente",
43   "Robo-dumb"),
44   trades = c(NA_integer_, NA_integer_, rt_trades, rand_trades, omn_trades,
45     dumb_trades),
46   ganancia_simple = c(buy_hold_gain, trampa_gain, robo_trader_gain, random_gain,
47     robo_omnisciente_gain, robo_dumb_gain),
48   ganancia_por_trade = ganancia_simple / trades)
49
50 strategies_tbl <- strategies |>
51   mutate(ganancia_simple = round(ganancia_simple, 2),
52     ganancia_por_trade = round(ganancia_por_trade, 2))
53
54 strategies_tbl |>
55   kable(digits = c(NA, 0, 2, 2))
```

estrategia	trades	ganancia_simple	ganancia_por_trade
Buy & hold	NA	34.43	NA
Trampa	NA	172.68	NA
Robo-trader	83	47.03	0.57
Aleatoria	83	22.30	0.27
Robo-omnisciente	35	189.00	5.40
Robo-dumb	48	-245.07	-5.11

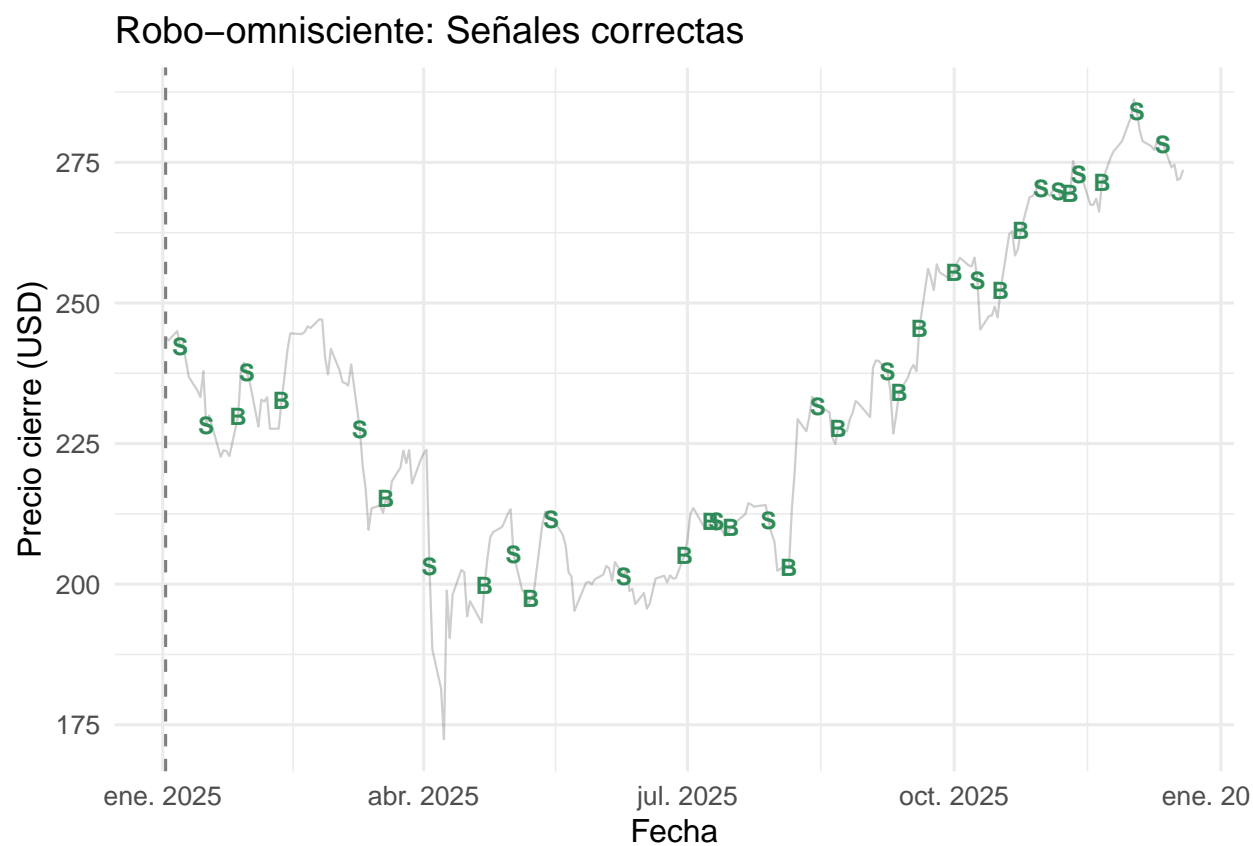
12 Visualización del Robo-trader.

```
1 signals_plot <- signals_rt |>
2   left_join(signal_outcomes |> select(date, signal, correct),
3     by = c("date", "signal")) |>
4   mutate(correcto = if_else(correct, "Correcta", "Incorrecta"))
5
6 ggplot(rt_data, aes(date, close)) +
7   geom_line(color = "gray35", linewidth = 0.4, alpha = 0.3) +
8   geom_text(data = signals_plot,
9     aes(y = price, label = if_else(signal == "buy", "B", "S"),
10       color = correcto), fontface = "bold", size = 3) +
11   scale_color_manual(values = c(Correcta = "#2E8B57", Incorrecta = "red"),
12     name = "Resultado") +
13   geom_vline(xintercept = as.Date("2025-01-02"), linetype = "dashed", color = "gray50") +
14   labs(title = "Robo-trader MA3: Señales correctas vs incorrectas",
15     x = "Fecha", y = "Precio cierre (USD)") +
16   theme_minimal(base_size = 12) +
17   theme(legend.position = "bottom")
```



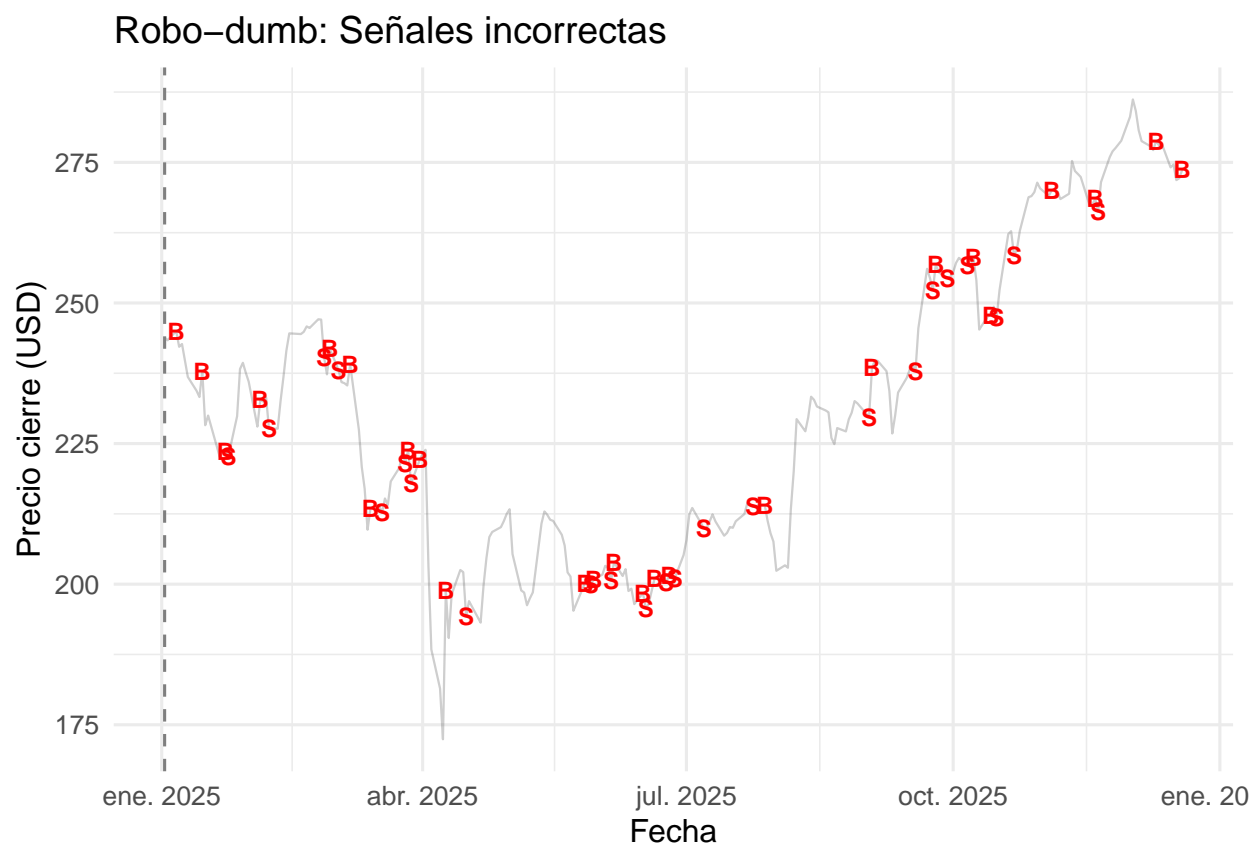
13 Visualización del Robo-omnisciente.

```
1 signals_plot_omn <- signals_omniscient |>
2   mutate(correcto = "Correcta")
3
4 ggplot(rt_data, aes(date, close)) +
5   geom_line(color = "gray35", linewidth = 0.4, alpha = 0.3) +
6   geom_text(data = signals_plot_omn,
7     aes(y = price, label = if_else(signal == "buy", "B", "S"), color = correcto),
8     fontface = "bold", size = 3) +
9   scale_color_manual(values = c(Correcta = "#2E8B57"), name = "Resultado") +
10  geom_vline(xintercept = as.Date("2025-01-02"), linetype = "dashed", color = "gray50") +
11  labs(title = "Robo-omnisciente: Señales correctas",
12    x = "Fecha", y = "Precio cierre (USD)") +
13  theme_minimal(base_size = 12) +
14  theme(legend.position = "none")
```



14 Visualización del Robo-dumb.

```
1 signals_plot_dumb <- signals_dumb |>
2   mutate(correcto = "Incorrecta")
3
4 ggplot(rt_data, aes(date, close)) +
5   geom_line(color = "gray35", linewidth = 0.4, alpha = 0.3) +
6   geom_text(data = signals_plot_dumb,
7             aes(y = price, label = if_else(signal == "buy", "B", "S"), color = correcto),
8             fontface = "bold", size = 3) +
9   scale_color_manual(values = c(Incorrecta = "red"), name = "Resultado") +
10  geom_vline(xintercept = as.Date("2025-01-02"), linetype = "dashed", color = "gray50") +
11  labs(title = "Robo-dumb: Señales incorrectas",
12        x = "Fecha", y = "Precio cierre (USD)") +
13  theme_minimal(base_size = 12) +
14  theme(legend.position = "none")
```



15 Bitácora Robo-trader.

```
1 bitacora_num <- make_bitacora(signals_rt, final_date, Pend) |>
2   mutate(price = round(price, 2),
3           flujo = round(flujo, 2),
4           PnL = round(PnL, 2),
5           PnL_acum = round(PnL_acum, 2))
6
7 bitacora_num |>
8   show_head_tail(n = 5) |>
9   kable(digits = c(NA, NA, 2, 2, 2, 2))
```

date	acción	price	flujo	PnL	PnL_acum
2025-01-06	buy	245.00	-245.00	0.00	0.00
2025-01-07	sell	242.21	242.21	-2.79	-2.79
2025-01-15	buy	237.87	-237.87	4.34	1.55
2025-01-16	sell	228.26	228.26	-9.61	-8.06
2025-01-23	buy	223.66	-223.66	4.60	-3.46
2025-12-03	sell	284.15	284.15	12.66	37.55
2025-12-10	buy	278.78	-278.78	5.37	42.92
2025-12-12	sell	278.28	278.28	-0.50	42.42
2025-12-19	buy	273.67	-273.67	4.61	47.03
2025-12-19	final_close	273.67	0.00	0.00	47.03

- 03-dic. Señal sell abre un short en 284.15; ganancia 0 en la apertura, PnL acum 37.55 arrastrado de antes.
- 10-dic. Señal buy cierra el short 284.15→278.78 con +5.37 y abre un long a 278.78; PnL acum 42.92.
- 12-dic. Señal sell cierra el long 278.78→278.28 con -0.50 y abre un short a 278.28; PnL acum 42.42.
- 19-dic. Señal buy cierra el short 278.28→273.67 con +4.61 y abre un long a 273.67; PnL acum 47.03.
- 19-dic (final_close). Cierra el long 273.67→273.67 con 0; PnL final 47.03.

16 Resumen Robo-trader.

```
1 confusion <- signal_outcomes |>
2   mutate(outcome = if_else(correct, "correcta", "incorrecta")) |>
3   count(signal, outcome) |>
4   pivot_wider(names_from = outcome, values_from = n, values_fill = 0)
5
6 confusion
```

```
## # A tibble: 2 x 3
##   signal correcta incorrecta
##   <chr>      <int>      <int>
## 1 buy         17         25
## 2 sell         18         23
```

17 Bitácora estrategia aleatoria.

```
1 bitacora_aleatoria_num <- make_bitacora(signals_rand, final_date, Pend) |>
2   mutate(price = round(price, 2),
3          flujo = round(flujo, 2),
4          PnL = round(PnL, 2),
5          PnL_acum = round(PnL_acum, 2))
6
7 bitacora_aleatoria_num |>
8   show_head_tail(n = 5) |>
9   kable(digits = c(NA, NA, 2, 2, 2, 2))
```

date	acción	price	flujo	PnL	PnL_acum
2025-01-07	buy	242.21	-242.21	0.00	0.00
2025-01-10	sell	236.85	236.85	-5.36	-5.36
2025-01-13	buy	234.40	-234.40	2.45	-2.91
2025-01-22	sell	223.83	223.83	-10.57	-13.48
2025-01-23	buy	223.66	-223.66	0.17	-13.31
2025-12-11	sell	278.03	278.03	-0.75	25.28
2025-12-12	buy	278.28	-278.28	-0.25	25.03
2025-12-16	sell	274.61	274.61	-3.67	21.36
2025-12-19	buy	273.67	-273.67	0.94	22.30
2025-12-19	final_close	273.67	0.00	0.00	22.30

18 Bitácora Robo-omnisciente.

```
1 bitacora_omnisciente_num <- make_bitacora(signals_omniscient, final_date, Pend) |>
2   mutate(price = round(price, 2),
3           flujo = round(flujo, 2),
4           PnL = round(PnL, 2),
5           PnL_acum = round(PnL_acum, 2))
6
7 bitacora_omnisciente_num |>
8   show_head_tail(n = 5) |>
9   kable(digits = c(NA, NA, 2, 2, 2, 2))
```

date	acción	price	flujo	PnL	PnL_acum
2025-01-07	sell	242.21	242.21	0.00	0.00
2025-01-16	sell	228.26	228.26	13.95	13.95
2025-01-27	buy	229.86	-229.86	-1.60	12.35
2025-01-30	sell	237.59	237.59	7.73	20.08
2025-02-11	buy	232.62	-232.62	4.97	25.05
2025-11-13	sell	272.95	272.95	3.52	164.40
2025-11-21	buy	271.49	-271.49	1.46	165.86
2025-12-03	sell	284.15	284.15	12.66	178.52
2025-12-12	sell	278.28	278.28	5.87	184.39
2025-12-19	final_close	273.67	0.00	4.61	189.00

- 13-nov. Señal sell abre un short a 272.95; ganancia 0 (solo apertura), PnL acum 164.40.
- 21-nov. Señal buy cierra el short 272.95→271.49 con +1.46 y abre un long a 271.49; PnL acum 165.86.
- 03-dic. Señal sell cierra el long 271.49→284.15 con +12.66 y abre un short a 284.15; PnL acum 178.52.
- 12-dic. Señal sell repetida cierra el short 284.15→278.28 con +5.87 y reabre short a 278.28; PnL acum 184.39.
- 19-dic (final_close). Cierra el short 278.28→273.67 con +4.61; PnL final 189.00.

19 Bitácora Robo-dumb.

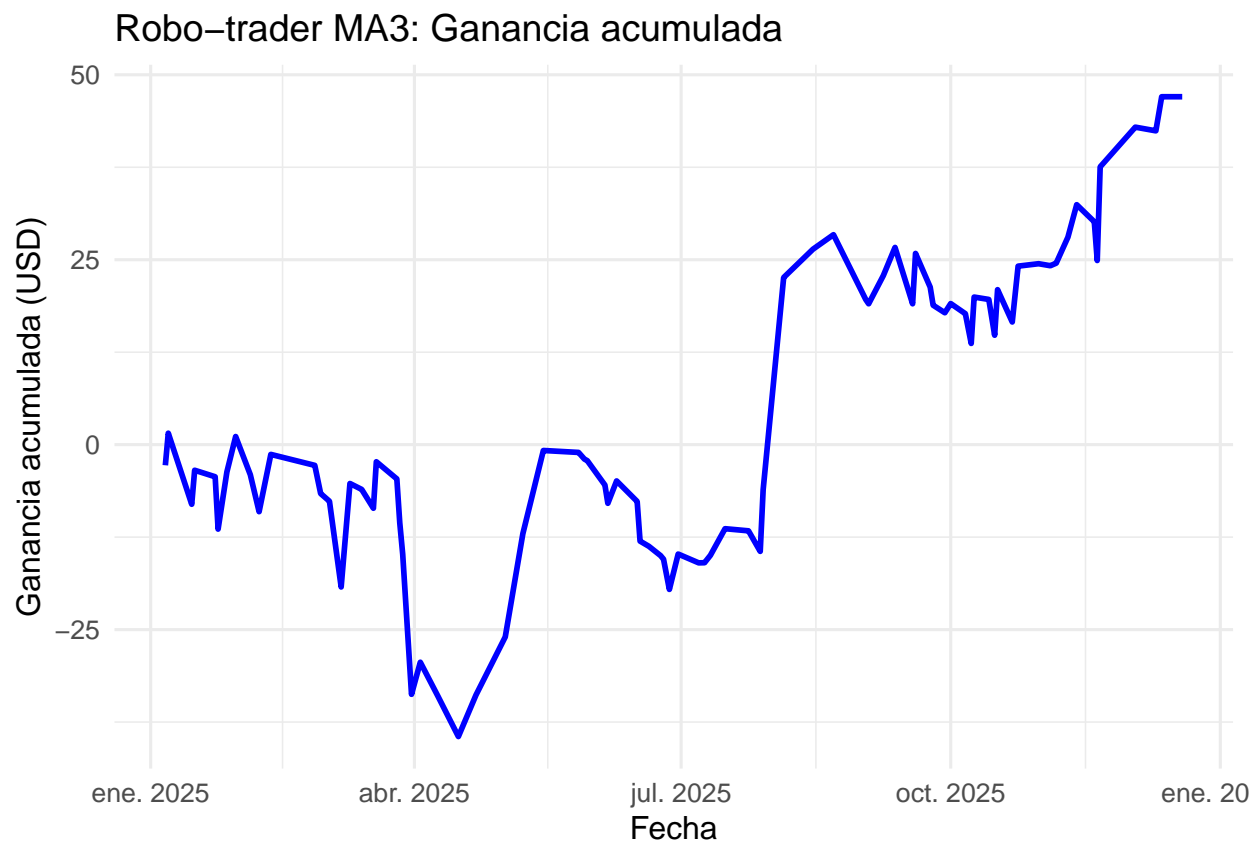
```
1 bitacora_dumb_num <- make_bitacora(signals_dumb, final_date, Pend) |>
2   mutate(price = round(price, 2),
3           flujo = round(flujo, 2),
4           PnL = round(PnL, 2),
5           PnL_acum = round(PnL_acum, 2))
6
7 bitacora_dumb_num |>
8   show_head_tail(n = 5) |>
9   kable(digits = c(NA, NA, 2, 2, 2, 2))
```

date	acción	price	flujo	PnL	PnL_acum
2025-01-06	buy	245.00	-245.00	0.00	0.00
2025-01-15	buy	237.87	-237.87	-7.13	-7.13
2025-01-23	buy	223.66	-223.66	-14.21	-21.34
2025-01-24	sell	222.78	222.78	-0.88	-22.22
2025-02-04	buy	232.80	-232.80	-10.02	-32.24
2025-11-19	buy	268.56	-268.56	-1.48	-225.12
2025-11-20	sell	266.25	266.25	-2.31	-227.43
2025-12-10	buy	278.78	-278.78	-12.53	-239.96
2025-12-19	buy	273.67	-273.67	-5.11	-245.07
2025-12-19	final_close	273.67	0.00	0.00	-245.07

- 19-nov. Señal buy abre un long en 268.56; ganancia 0 en la apertura, PnL acum -225.12 arrastrado de antes.
- 20-nov. Señal sell cierra el long 268.56→266.25 con -2.31 y abre un short a 266.25; PnL acum -227.43.
- 10-dic. Señal buy cierra el short 266.25→278.78 con -12.53 y abre un long a 278.78; PnL acum -239.96.
- 19-dic (buy repetida): cierra el long 278.78→273.67 con -5.11 y reabre long a 273.67; PnL acum -245.07.
- 19-dic (final_close). Cierra el long 273.67→273.67 con 0; PnL final -245.07.

20 Ganancias Robo-trader.

```
1 ggplot(signal_outcomes |> mutate(PnL_acum = cumsum(replace_na(trade_pnl, 0))),  
2   aes(x = date, y = PnL_acum)) +  
3   geom_line(color = "blue", linewidth = 1) +  
4   labs(title = "Robo-trader MA3: Ganancia acumulada",  
5         x = "Fecha", y = "Ganancia acumulada (USD)") +  
6   theme_minimal(base_size = 12)
```



21 Conclusión.

- La comparación evidencia cómo una regla mecánica (MA3) se posiciona frente a estrategias pasivas, un benchmark idealizado y controles aleatorios.
- La bitácora y los gráficos permiten validar la lógica del algoritmo y la trazabilidad del PnL por operación.
- En conjunto, el ejercicio sirve como referencia metodológica para evaluar reglas de trading en términos de desempeño y frecuencia de operación.