

# Carteras de inversión.

Introducción al diseño y análisis.

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

04 de diciembre de 2025, 01:18 a.m.

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

## 1 Introducción.

- Conceptos fundamentales del análisis riesgo–rendimiento.
- Explicar la frontera media–varianza, destacando cómo se obtiene mediante métodos analíticos y numéricos.
- Construir y evaluar portafolios usando datos reales.
- Cómo los pesos óptimos cambian cuando se permiten o restringen las ventas en corto.

## 2 Diez empresas públicas de Estados Unidos.

Ticker	Nombre de la empresa	Industria
AMD	Advanced Micro Dev.	Computación de alto rendimiento
CNC	Centene Corp.	Servicios de salud
GIS	General Mills, Inc.	Productos de consumo envasados
LMT	Lockheed Martin Corp.	Aeroespacial y defensa
LRCX	Lam Research Corp.	Semiconductores
NEM	Newmont Corp.	Minería de metales preciosos
SNPS	Synopsys, Inc.	Software de diseño
SYF	Synchrony Financial	Servicios financieros
TRMB	Trimble Inc.	Tecnología geoespacial
TTD	The Trade Desk, Inc.	Tecnología publicitaria

### 3 Paquetes.

```
1 library(tidyquant)
2 library(tidyverse)
3 library(lubridate)
4 library(scales)
5 library(quadprog)
```

## 4 Inicialización, parte 1.

```
1 # Empresas y rango temporal
2
3 tickers <- c("AMD", "CNC", "GIS", "LMT", "LRCX", "NEM", "SNPS", "SYF", "TRMB", "TTD")
4 n_months <- 60L
5 price_start <- ymd("2020-09-01")
6 price_end <- price_start %m+% months(n_months + 1) - days(1)
7 returns_start <- price_start %m+% months(1)
8 returns_end <- returns_start %m+% months(n_months) - months(1)
9
10 # Descarga de precios históricos de las acciones
11
12 prices <- tq_get(tickers, from = price_start, to = price_end, get = "stock.prices")
13
14 # Cálculo de retornos mensuales por activo
15
16 monthly_returns <- prices |>
17   arrange(symbol, date) |>
18   group_by(symbol) |>
19   tq_transmute(select = adjusted, mutate_fun = periodReturn,
20               period = "monthly", type = "arithmetic",
21               col_rename = "monthly_return") |>
22   filter(between(date, returns_start, returns_end)) |>
23   slice_head(n = n_months) |>
24   ungroup()
25
26 stopifnot(n_distinct(monthly_returns$symbol) == length(tickers))
27
28 # Función: estadísticas básicas por activo
29
30 asset_stats <- function(data) {
31   data |>
32     group_by(symbol) |>
33     summarise(ER = mean(monthly_return, na.rm = TRUE),
34             SD = sd(monthly_return, na.rm = TRUE),
35             SR = ER / SD, .groups = "drop")}
36
37 # Función: construcción de rendimiento acumulado por activo
38
39 cum_index <- function(data, start_date) {
40   base <- tibble(date = start_date, symbol = unique(data$symbol), index_level = 1)
41   data |>
42     arrange(symbol, date) |>
43     group_by(symbol) |>
44     mutate(index_level = cumprod(1 + monthly_return)) |>
45     ungroup() |>
46     select(date, symbol, index_level) |>
47     bind_rows(base) |>
48     arrange(symbol, date)}
```

## 5 Inicialización, parte 2.

```
1  # Función: construcción de rendimiento acumulado por cartera
2
3  portfolio_series <- function(data, weights, label, start_date) {
4    data |>
5      filter(symbol %in% names(weights)) |>
6      group_by(date) |>
7      summarise(portfolio_return = sum(monthly_return * weights[symbol]), .groups = "drop") |>
8      arrange(date) |>
9      mutate(index_level = cumprod(1 + portfolio_return), symbol = label) |>
10     select(date, symbol, index_level) |>
11     bind_rows(tibble(date = start_date, symbol = label, index_level = 1))}
12
13 # Función: estadísticas de un portafolio
14
15 portfolio_stats <- function(data, weights, label) {
16   pr <- data |>
17     filter(symbol %in% names(weights)) |>
18     group_by(date) |>
19     summarise(portfolio_return = sum(monthly_return * weights[symbol]), .groups = "drop")
20
21   tibble(symbol = label, ER = mean(pr$portfolio_return, na.rm = TRUE),
22         SD = sd(pr$portfolio_return, na.rm = TRUE), SR = ER / SD)}
23
24 # Conversión de retornos a formato ancho y matriz
25
26 returns_wide <- monthly_returns |>
27   select(date, symbol, monthly_return) |>
28   pivot_wider(names_from = symbol, values_from = monthly_return) |>
29   arrange(date)
30
31 returns_mat_all <- returns_wide |>
32   select(-date) |>
33   as.matrix()
34
35 # Cálculo de medias, covarianzas y estadísticas
36
37 mu_vec_all <- colMeans(returns_mat_all, na.rm = TRUE)
38 cov_mat_all <- cov(returns_mat_all, use = "pairwise.complete.obs")
39 n_assets <- length(mu_vec_all)
40
41 stats_all <- tibble(symbol = names(mu_vec_all), ER = mu_vec_all,
42                   SD = apply(returns_mat_all, 2, sd, na.rm = TRUE))
43
44 color_values_all <- setNames(scales::hue_pal()(length(mu_vec_all)),
45                             names(mu_vec_all))
```

## 6 Matriz de correlaciones.

```
1 corr_mat <- returns_wide |>
2   select(-date) |>
3   cor(use = "pairwise.complete.obs")
4
5 print(round(corr_mat, 2), quote = FALSE, right = TRUE)
```

##		AMD	CNC	GIS	LMT	LRCX	NEM	SNPS	SYF	TRMB	TTD
##	AMD	1.00	-0.10	-0.16	-0.12	0.71	-0.06	0.70	0.35	0.50	0.51
##	CNC	-0.10	1.00	0.37	0.37	0.14	0.20	-0.16	0.12	0.01	-0.18
##	GIS	-0.16	0.37	1.00	0.46	-0.20	0.25	-0.27	-0.15	-0.20	-0.20
##	LMT	-0.12	0.37	0.46	1.00	0.00	0.32	-0.22	0.20	0.01	-0.10
##	LRCX	0.71	0.14	-0.20	0.00	1.00	0.04	0.56	0.59	0.61	0.38
##	NEM	-0.06	0.20	0.25	0.32	0.04	1.00	-0.04	-0.08	-0.05	-0.26
##	SNPS	0.70	-0.16	-0.27	-0.22	0.56	-0.04	1.00	0.33	0.48	0.31
##	SYF	0.35	0.12	-0.15	0.20	0.59	-0.08	0.33	1.00	0.67	0.29
##	TRMB	0.50	0.01	-0.20	0.01	0.61	-0.05	0.48	0.67	1.00	0.44
##	TTD	0.51	-0.18	-0.20	-0.10	0.38	-0.26	0.31	0.29	0.44	1.00

## 7 Pares de correlaciones mínimas.

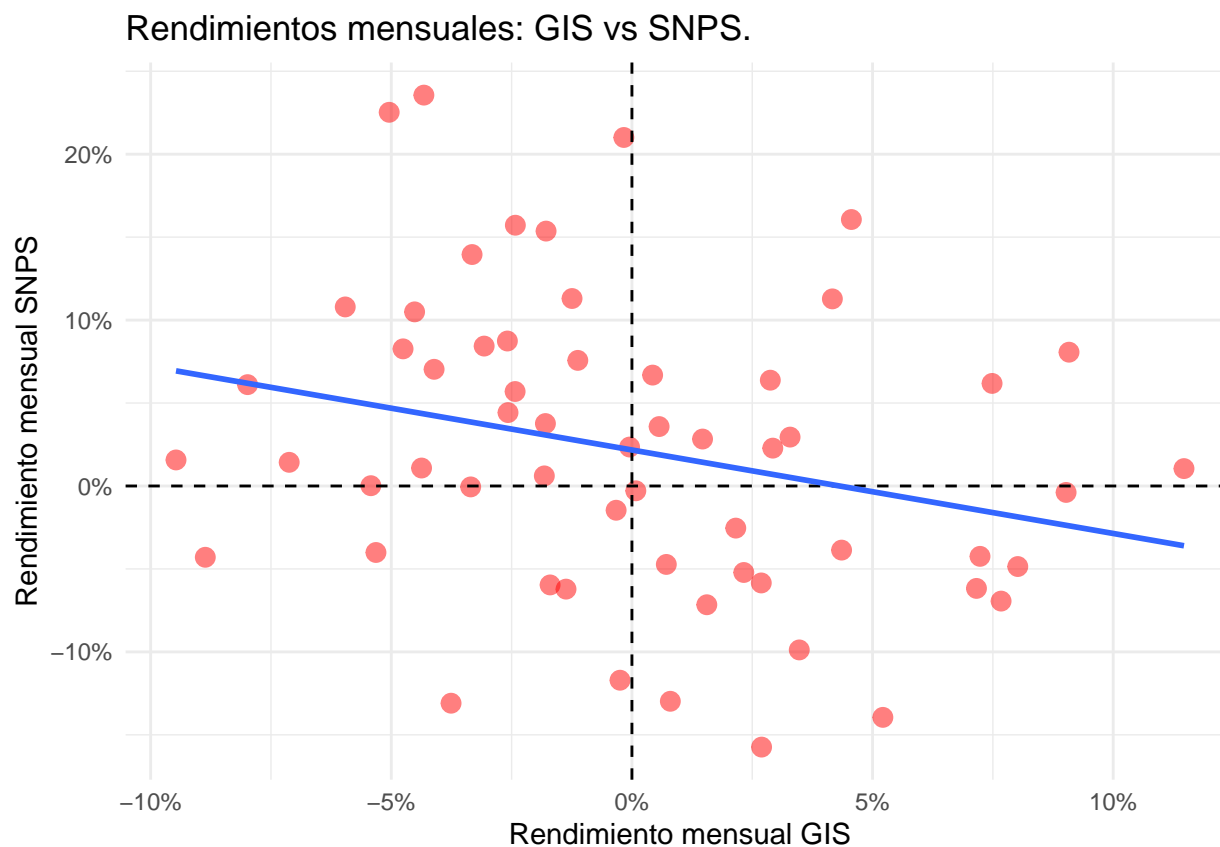
$$\sigma_{port}^2 = w_A^2 \sigma_A^2 + w_B^2 \sigma_B^2 + 2w_A w_B \sigma_A \sigma_B \rho_{A,B}.$$

```
1 lowest_pairs <- corr_mat |>
2   as_tibble(rownames = "symbol_i") |>
3   pivot_longer(-symbol_i, names_to = "symbol_j", values_to = "corr") |>
4   filter(symbol_i < symbol_j) |>
5   arrange(corr) |>
6   slice_head(n = 10)
7
8 print(lowest_pairs)
```

```
## # A tibble: 10 x 3
##   symbol_i symbol_j   corr
##   <chr>    <chr>    <dbl>
## 1 GIS      SNPS      -0.266
## 2 NEM      TTD       -0.263
## 3 LMT      SNPS      -0.217
## 4 GIS      TTD       -0.201
## 5 GIS      LRCX      -0.201
## 6 GIS      TRMB      -0.198
## 7 CNC      TTD       -0.180
## 8 AMD      GIS       -0.162
## 9 CNC      SNPS      -0.156
## 10 GIS     SYF       -0.148
```

## 8 Ilustración de la correlación.

```
1 # Pasar a formato ancho con una columna para GIS y otra para SNPS
2 gis_snps_wide <- monthly_returns |>
3   filter(symbol %in% c("GIS", "SNPS")) |>
4   select(date, symbol, monthly_return) |>
5   pivot_wider(names_from = symbol, values_from = monthly_return) |>
6   drop_na()
7
8 # Scatter GIS (eje X) vs SNPS (eje Y)
9 ggplot(gis_snps_wide, aes(x = GIS, y = SNPS)) +
10   geom_point(size = 3, alpha = 0.5, col = "red") +
11   geom_hline(yintercept = 0, linetype = "dashed") +
12   geom_vline(xintercept = 0, linetype = "dashed") +
13   scale_x_continuous(labels = percent_format(accuracy = 1)) +
14   scale_y_continuous(labels = percent_format(accuracy = 1)) +
15   labs(title = "Rendimientos mensuales: GIS vs SNPS.",
16        x = "Rendimiento mensual GIS", y = "Rendimiento mensual SNPS") +
17   geom_smooth(method = "lm", se = FALSE) +
18   theme_minimal()
```



## 9 Riesgo-rendimiento: 10 activos individuales.

```
1 stats <- asset_stats(monthly_returns) |>
2   arrange(-SR) |>
3   print()
```

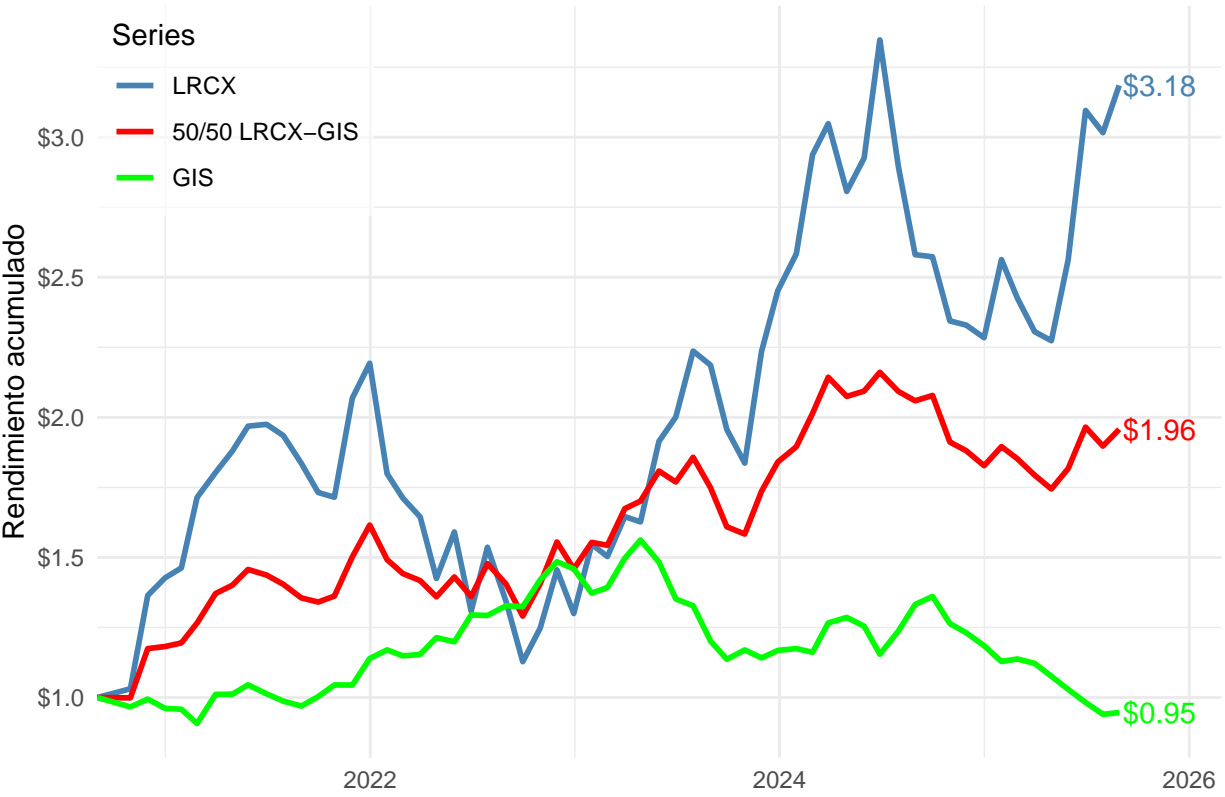
```
## # A tibble: 10 x 4
##   symbol      ER      SD      SR
##   <chr>    <dbl> <dbl>   <dbl>
## 1 SYF      0.0259  0.106  0.244
## 2 SNPS     0.0216  0.0902  0.240
## 3 LRCX     0.0257  0.111  0.231
## 4 AMD      0.0223  0.150  0.149
## 5 TRMB     0.0132  0.0970  0.136
## 6 LMT      0.00709 0.0647  0.110
## 7 NEM      0.0107  0.103  0.103
## 8 TTD      0.0170  0.184  0.0926
## 9 GIS      0.000182 0.0477  0.00382
## 10 CNC     -0.00542 0.102 -0.0532
```

## 10 Rendimiento acumulado, cartera ingenua.

$$R_{port} = 0.5\mu_{LRCX} + 0.5\mu_{GIS}.$$

```
1 focus <- c("LRCX", "GIS")
2 color_values <- c("LRCX"="steelblue","GIS"="green","50/50 LRCX-GIS"="red")
3
4 cum_focus <- monthly_returns |>
5   filter(symbol %in% focus) |>
6   cum_index(price_start)
7
8 portfolio_50_50 <- portfolio_series(monthly_returns, c(LRCX = 0.5, GIS = 0.5),
9                                     "50/50 LRCX-GIS", price_start)
10
11 plot_data <- bind_rows(cum_focus, portfolio_50_50)
12
13 last_vals <- plot_data |>
14   group_by(symbol) |>
15   summarise(final_value = last(index_level), .groups = "drop") |>
16   arrange(desc(final_value))
17
18 label_data <- plot_data |>
19   group_by(symbol) |>
20   filter(date == max(date)) |>
21   mutate(label = paste0("$", formatC(index_level, format="f", digits=2))) |>
22   ungroup()
23
24 ggplot(plot_data, aes(date, index_level,
25                       color = factor(symbol, levels = last_vals$symbol))) +
26   geom_line(linewidth = 1) +
27   geom_text(data = label_data, aes(label = label),
28           hjust = -0.05, vjust = 0.5, show.legend = FALSE) +
29   scale_color_manual(values = color_values, limits = last_vals$symbol) +
30   scale_x_date(expand = expansion(mult = c(0, 0.1))) +
31   scale_y_continuous(labels = dollar_format(prefix = "$", accuracy = 0.1)) +
32   labs(title = "Rendimiento acumulado, 2 activos y una cartera.",
33        y = "Rendimiento acumulado",
34        x = NULL, color = "Series") +
35   theme_minimal() +
36   theme(legend.position = c(0,1), legend.justification = c(0,1),
37         legend.background = element_rect(fill = alpha("white", 0.6), color = NA))
```

Rendimiento acumulado, 2 activos y una cartera.



## 11 Riesgo-rendimiento: cartera ingenua y componentes.

$$R_{port} = 0.5\mu_{LRCX} + 0.5\mu_{GIS}.$$

```
1 weights_50_50 <- c(LRCX = 0.5, GIS = 0.5)
2
3 stats_lrcx_gis <- stats |>
4   filter(symbol %in% names(weights_50_50))
5
6 portfolio_stats_50_50 <- portfolio_stats(monthly_returns, weights_50_50, "50/50 LRCX-GIS")
7
8 final_table <- bind_rows(stats_lrcx_gis, portfolio_stats_50_50) |>
9   arrange(-SR)
10
11 print(final_table)
```

```
## # A tibble: 3 x 4
##   symbol          ER      SD      SR
##   <chr>        <dbl>  <dbl>  <dbl>
## 1 50/50 LRCX-GIS 0.0130  0.0560 0.231
## 2 LRCX          0.0257  0.111  0.231
## 3 GIS           0.000182 0.0477 0.00382
```

## 12 Frontera media-varianza: 2 activos.

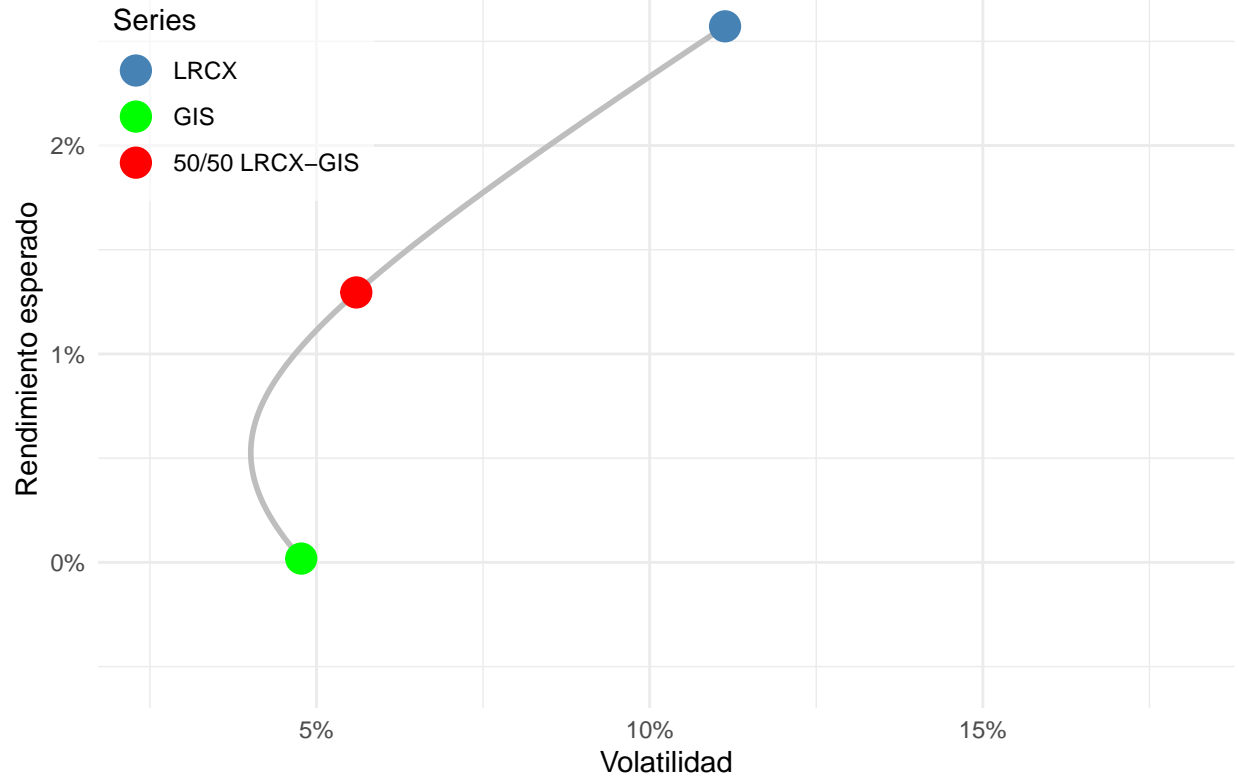
$$ER_{port} = w_A \mu_A + w_B \mu_B.$$

$$\sigma_{port}^2 = w_A^2 \sigma_A^2 + w_B^2 \sigma_B^2 + 2w_A w_B \sigma_{A,B}.$$

$$\sigma_{port}^2 = w_A^2 \sigma_A^2 + w_B^2 \sigma_B^2 + 2w_A w_B \sigma_A \sigma_B \rho_{A,B}.$$

```
1 returns_matrix_2 <- returns_wide |>
2   select(all_of(focus))
3
4 cov_mat_2 <- cov(returns_matrix_2, use = "pairwise.complete.obs")
5 mu_vec_2 <- colMeans(returns_matrix_2, na.rm = TRUE)
6
7 frontier_data <- tibble(weight_lrcx = seq(0, 1, length.out = 200)) |>
8   mutate(weight_gis = 1 - weight_lrcx,
9          ER = weight_lrcx * mu_vec_2["LRCX"] + weight_gis * mu_vec_2["GIS"],
10         variance = weight_lrcx^2 * var(returns_matrix_2$LRCX, na.rm = TRUE) +
11         weight_gis^2 * var(returns_matrix_2$GIS, na.rm = TRUE) +
12         2 * weight_lrcx * weight_gis * cov_mat_2["LRCX", "GIS"],
13         SD = sqrt(variance))
14
15 mean_var_points <- final_table |>
16   mutate(symbol = factor(symbol, levels = c("LRCX", "GIS", "50/50 LRCX-GIS"))) |>
17   arrange(symbol)
18
19 ggplot() +
20   geom_path(data = frontier_data, aes(SD, ER), color = "grey", linewidth = 1) +
21   geom_point(data = mean_var_points, aes(SD, ER, color = symbol), size = 5) +
22   scale_color_manual(values = color_values, limits = levels(mean_var_points$symbol)) +
23   scale_x_continuous(limits = c(0.025, 0.18), labels = percent_format(accuracy = 1)) +
24   scale_y_continuous(limits = c(min(mu_vec_all), max(mu_vec_all)),
25                     labels = percent_format(accuracy = 1)) +
26   labs(title = "Frontera media-varianza, 2 activos.",
27        x = "Volatilidad", y = "Rendimiento esperado", color = "Series") +
28   theme_minimal() +
29   theme(legend.position = c(0,1), legend.justification = c(0,1),
30         legend.background = element_rect(fill = alpha("white", 0.6), color = NA))
```

### Frontera media-varianza, 2 activos.



### 13 Cartera de mínima varianza: método numérico de optimización.

$\min_{\mathbf{w}} \mathbf{w}^\top \Sigma \mathbf{w}$  sujeto a:  $\mathbf{1}^\top \mathbf{w} = 1$ .

```
1 minvar2_label <- "Min-Var 2-Asset (LRCX-GIS)"
2
3 assets_2 <- focus
4 qp_2 <- solve.QP(Dmat = 2 * cov_mat_2,
5                 dvec = rep(0, length(assets_2)),
6                 Amat = matrix(1, nrow = length(assets_2), ncol = 1),
7                 bvec = 1, meq = 1)
8
9 weights_minvar2_qp <- setNames(qp_2$solution, assets_2)
10
11 print(weights_minvar2_qp)
```

```
##          LRCX          GIS
## 0.1990058 0.8009942
```

## 14 Resumen de componentes y carteras.

```
1 weights_50_50 <- c(LRCX = 0.5, GIS = 0.5)
2
3 stats_lrcx_gis <- stats |>
4   filter(symbol %in% names(weights_50_50))
5 stats_minvar2_qp <- stats |>
6   filter(symbol %in% names(weights_minvar2_qp))
7
8 portfolio_stats_50_50 <- portfolio_stats(monthly_returns, weights_50_50, "50/50 LRCX-GIS")
9 portfolio_stats_minvar2_qp <-
10   portfolio_stats(monthly_returns, weights_minvar2_qp, "MV LRCX-GIS")
11
12 final_table <- bind_rows(stats_lrcx_gis, portfolio_stats_50_50,
13                           portfolio_stats_minvar2_qp) |>
14   arrange(-SR)
15
16 print(final_table)
```

```
## # A tibble: 4 x 4
##   symbol          ER      SD      SR
##   <chr>        <dbl> <dbl> <dbl>
## 1 50/50 LRCX-GIS 0.0130  0.0560 0.231
## 2 LRCX          0.0257  0.111  0.231
## 3 MV LRCX-GIS  0.00526 0.0401 0.131
## 4 GIS           0.000182 0.0477 0.00382
```

## 15 Cartera de mínima varianza: método analítico algebraico para 2 activos.

$$w_1^* = \frac{\sigma_2^2 - \sigma_{12}}{\sigma_1^2 + \sigma_2^2 - 2\sigma_{12}}, \quad w_2^* = 1 - w_1^*.$$

```
1 Sigma_2 <- cov_mat_2[assets_2, assets_2]
2 var1 <- Sigma_2[1, 1]; var2 <- Sigma_2[2, 2]; cov12 <- Sigma_2[1, 2]
3 w1 <- (var2 - cov12) / (var1 + var2 - 2 * cov12)
4 weights_minvar2_closed <- setNames(c(w1, 1 - w1), assets_2)
5 print(weights_minvar2_closed)
```

```
##          LRCX          GIS
## 0.1990058 0.8009942
```

## 16 Cartera de mínima varianza: método analítico matricial para 2 activos.

$$w^* = \frac{\Sigma^{-1}\mathbf{1}}{\mathbf{1}^\top \Sigma^{-1}\mathbf{1}}.$$

```
1 Sigma_2 <- cov_mat_2[assets_2, assets_2, drop = FALSE]
2 ones <- rep(1, length(assets_2))
3 Sigma_inv <- solve(Sigma_2)
4 num <- Sigma_inv %*% ones
5 den <- as.numeric(t(ones) %*% Sigma_inv %*% ones)
6 w_star <- as.numeric(num / den)
7 names(w_star) <- assets_2
8 print(w_star)
```

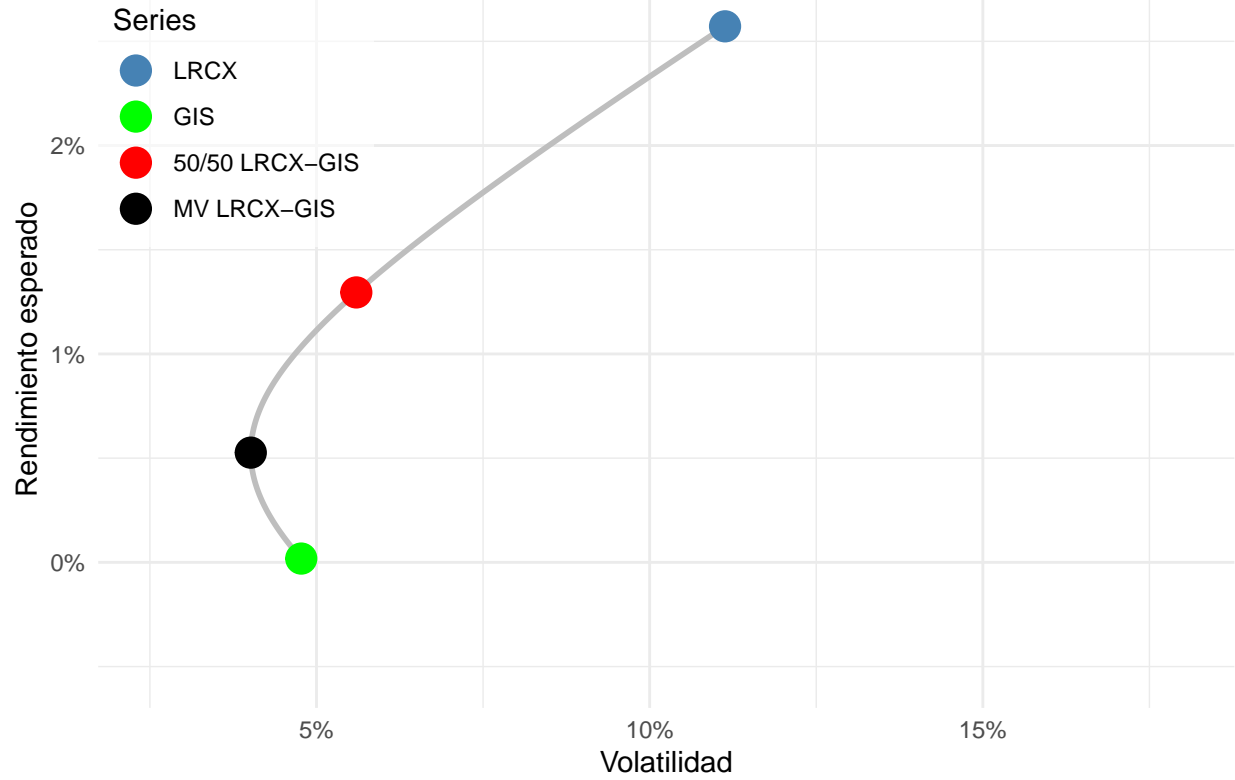
```
##          LRCX          GIS
## 0.1990058 0.8009942
```

## 17 Frontera media-varianza: 2 activos, cartera MV.

$$ER_{port} = \sum_{i=1}^n w_i \mu_i, \sigma_{port} = \sqrt{\mathbf{w}^\top \Sigma \mathbf{w}}.$$

```
1 minvar2_label <- "MV LRCX-GIS"
2 minvar_point_2 <- tibble(symbol = minvar2_label,
3   ER = sum(weights_minvar2_qp * mu_vec_2),
4   SD = sqrt(as.numeric(t(weights_minvar2_qp) %*% cov_mat_2 %*% weights_minvar2_qp)))
5
6 mean_var_points_ext <- bind_rows(stats_lrcx_gis, portfolio_stats_50_50, minvar_point_2) |>
7   mutate(symbol = factor(symbol,
8     levels = c("LRCX", "GIS", "50/50 LRCX-GIS", minvar2_label))) |>
9   arrange(symbol)
10
11 color_values_ext <- c(color_values, setNames("black", minvar2_label))
12
13 ggplot() +
14   geom_path(data = frontier_data, aes(SD, ER), color = "grey", linewidth = 1) +
15   geom_point(data = mean_var_points_ext, aes(SD, ER, color = symbol), size = 5) +
16   scale_color_manual(values = color_values_ext, limits = levels(mean_var_points_ext$symbol)) +
17   scale_x_continuous(limits = c(0.025, 0.18), labels = percent_format(accuracy = 1)) +
18   scale_y_continuous(limits = c(min(mu_vec_all), max(mu_vec_all)),
19     labels = percent_format(accuracy = 1)) +
20   labs(title = "Frontera media-varianza, 2 activos y 2 carteras.",
21     x = "Volatilidad", y = "Rendimiento esperado", color = "Series") +
22   theme_minimal() +
23   theme(legend.position = c(0, 1), legend.justification = c(0, 1),
24     legend.background = element_rect(fill = alpha("white", 0.6), color = NA))
```

Frontera media-varianza, 2 activos y 2 carteras.



## 18 Cartera de mínima varianza: método numérico para 10 activos.

$\min_{\mathbf{w}} \mathbf{w}^\top \Sigma \mathbf{w}$  sujeto a:  $\mathbf{1}^\top \mathbf{w} = 1, \mathbf{w} \geq \mathbf{0}$ .

```
1 minvar10_ns_label <- "MV 10-Asset (No Shorts)"
2
3 weights_minvar10_ns <- solve.QP(2 * cov_mat_all, rep(0, length(mu_vec_all)),
4   cbind(rep(1, length(mu_vec_all)), diag(length(mu_vec_all))),
5   c(1, rep(0, length(mu_vec_all))), meq = 1)$solution
6
7 names(weights_minvar10_ns) <- names(mu_vec_all)
8
9 minvar_point_10_ns <- tibble(symbol = minvar10_ns_label,
10   ER = sum(weights_minvar10_ns * mu_vec_all),
11   SD = sqrt(as.numeric(t(weights_minvar10_ns) %*% cov_mat_all %*% weights_minvar10_ns)),
12   SR = ER/SD)
13
14 minvar_point_10_ns
```

```
## # A tibble: 1 x 4
```

```
##   symbol          ER      SD      SR
##   <chr>        <dbl>  <dbl> <dbl>
## 1 MV 10-Asset (No Shorts) 0.00717 0.0337 0.213
```

## 19 Cartera de mínima varianza: método numérico para 10 activos, pesos.

$\min_{\mathbf{w}} \mathbf{w}^\top \Sigma \mathbf{w}$  sujeto a:  $\mathbf{1}^\top \mathbf{w} = 1, \mathbf{w} \geq 0$ .

```
1 minvar_weights_table_ns <- tibble(symbol = names(mu_vec_all),
2   weight = weights_minvar10_ns,
3   weight_pct = scales::percent(weights_minvar10_ns, accuracy = 0.01)) |>
4   arrange(desc(weight)) |>
5   bind_rows(tibble(symbol = "Total",
6   weight = sum(weights_minvar10_ns),
7   weight_pct = scales::percent(sum(weights_minvar10_ns),
8   accuracy = 0.01)))
9
10 print(minvar_weights_table_ns)
```

```
## # A tibble: 11 x 3
##   symbol    weight weight_pct
##   <chr>      <dbl> <chr>
## 1 GIS      5.40e- 1 54.01%
## 2 SNPS     1.83e- 1 18.34%
## 3 LMT      1.12e- 1 11.17%
## 4 TRMB     4.62e- 2 4.62%
## 5 NEM      4.52e- 2 4.52%
## 6 SYF      3.05e- 2 3.05%
## 7 TTD      3.00e- 2 3.00%
## 8 CNC      1.30e- 2 1.30%
## 9 AMD      1.21e-17 0.00%
## 10 LRCX     0      0.00%
## 11 Total   1      e+ 0 100.00%
```

## 20 Cartera de mínima varianza: método analítico matricial para 10 activos, pesos.

$$w^* = \frac{\Sigma^{-1}\mathbf{1}}{\mathbf{1}^\top \Sigma^{-1}\mathbf{1}}. \text{ Markowitz, ventas en corto permitidas.}$$

```

1 minvar10_short_label <- "MV (shorts)"
2
3 Sigma_all <- cov_mat_all
4 ones_all <- rep(1, length(mu_vec_all))
5 inv_Sigma_all <- solve(Sigma_all)
6 A_all <- as.numeric(t(ones_all) %*% inv_Sigma_all %*% ones_all)
7
8 weights_minvar10_short <- as.numeric(inv_Sigma_all %*% ones_all / A_all)
9 names(weights_minvar10_short) <- names(mu_vec_all)
10
11 minvar_point_10_short <-
12   tibble(symbol = minvar10_short_label,
13     ER = sum(weights_minvar10_short * mu_vec_all),
14     SD = sqrt(as.numeric(t(weights_minvar10_short) %*% Sigma_all %*% weights_minvar10_short)))
15
16 weights_gmv_table_short <- tibble(symbol = names(mu_vec_all),
17   weight = weights_minvar10_short,
18   weight_pct = scales::percent(weights_minvar10_short, accuracy = 0.01)) |>
19   arrange(desc(weight)) |>
20   bind_rows(tibble(symbol = "Total",
21     weight = sum(weights_minvar10_short),
22     weight_pct = scales::percent(sum(weights_minvar10_short),
23       accuracy = 0.01)))
24
25 print(weights_gmv_table_short)

```

```

## # A tibble: 11 x 3
##   symbol    weight weight_pct
##   <chr>      <dbl> <chr>
## 1 GIS        0.539   53.93%
## 2 SNPS        0.240   24.05%
## 3 LMT         0.106   10.62%
## 4 TRMB        0.0490   4.90%
## 5 TTD         0.0469   4.69%
## 6 NEM         0.0419   4.19%
## 7 LRCX        0.0416   4.16%
## 8 SYF         0.0187   1.87%
## 9 CNC         0.00510  0.51%
## 10 AMD       -0.0893  -8.93%
## 11 Total      1      100.00%

```

## 21 Frontera media-varianza: 10 activos, sin ventas en corto.

```
1 target_returns <- seq(min(mu_vec_all), max(mu_vec_all), length.out = 150)
2 Amat_ns <- rbind(rep(1, n_assets), mu_vec_all, diag(n_assets))
3
4 frontier_points_ns <- map_dfr(target_returns, function(tr) {
5   bvec <- c(1, tr, rep(0, n_assets))
6   sol <- tryCatch(solve.QP(2 * cov_mat_all, rep(0, n_assets),
7     t(Amat_ns), bvec, meq = 2), error = function(e) NULL)
8   if (is.null(sol)) return(NULL)
9   tibble(ER = tr,
10     SD = sqrt(as.numeric(t(sol$solution) %*% cov_mat_all %*% sol$solution)))) |>
11   drop_na()
12
13 stats_all_ext_ns <- bind_rows(stats_all, minvar_point_10_ns)
14 color_values_ext_ns <- c(color_values_all, setNames("black", minvar10_ns_label))
15
16 ggplot() +
17   geom_path(data = frontier_points_ns, aes(SD, ER), color = "grey", linewidth = 1) +
18   geom_point(data = stats_all_ext_ns, aes(SD, ER, color = symbol), size = 5) +
19   geom_text(data = stats_all_ext_ns, aes(SD, ER, label = symbol, color = symbol),
20     vjust = 2, size = 3, show.legend = FALSE) +
21   scale_color_manual(values = color_values_ext_ns, limits = stats_all_ext_ns$symbol) +
22   scale_x_continuous(limits = c(0.025, 0.19), labels = percent_format(accuracy = 1)) +
23   scale_y_continuous(limits = c(-0.01, 0.03), labels = percent_format(accuracy = 1)) +
24   labs(title = "Frontera media-varianza, 10 activos sin ventas en corto.",
25     x = "Volatilidad", y = "Rendimiento esperado") +
26   guides(color = "none") +
27   theme_minimal()
```

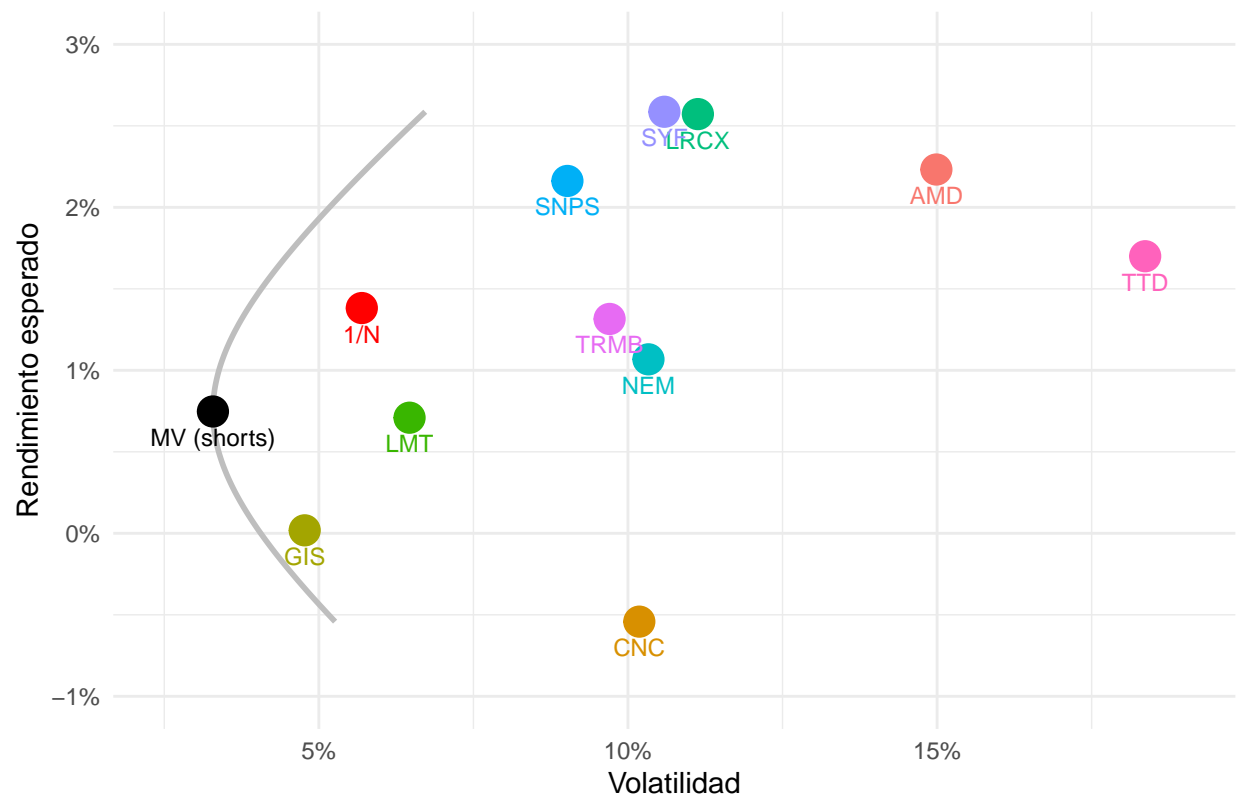
Frontera media-varianza, 10 activos sin ventas en corto.



## 22 Frontera media-varianza: 10 activos, con ventas en corto.

```
1 Amat_short <- rbind(rep(1, n_assets), mu_vec_all)
2
3 frontier_points_short <- map_dfr(target_returns, function(tr) {
4   bvec <- c(1, tr)
5   sol <- tryCatch(solve.QP(2 * cov_mat_all, rep(0, n_assets),
6     t(Amat_short), bvec, meq = 2), error = function(e) NULL)
7   if (is.null(sol)) return(NULL)
8   tibble(ER = tr,
9     SD = sqrt(as.numeric(t(sol$solution) %*% cov_mat_all %*% sol$solution)))) |>
10  drop_na()
11
12 # Punto 1/N (igual ponderación)
13 weights_1N <- setNames(rep(1 / n_assets, n_assets), names(mu_vec_all))
14 point_1N <- tibble(symbol = "1/N", ER = sum(weights_1N * mu_vec_all),
15   SD = sqrt(as.numeric(t(weights_1N) %*% cov_mat_all %*% weights_1N)))
16
17 stats_all_ext_short <- bind_rows(stats_all, minvar_point_10_short, point_1N)
18 color_values_ext_short <- c(color_values_all,
19   setNames("black", minvar10_short_label), "1/N" = "red")
20
21 ggplot() +
22   geom_path(data = frontier_points_short, aes(SD, ER), color = "grey", linewidth = 1) +
23   geom_point(data = stats_all_ext_short, aes(SD, ER, color = symbol), size = 5) +
24   geom_text(data = stats_all_ext_short, aes(SD, ER, label = symbol, color = symbol),
25     vjust = 2, size = 3, show.legend = FALSE) +
26   scale_color_manual(values = color_values_ext_short, limits = stats_all_ext_short$symbol) +
27   scale_x_continuous(limits = c(0.025, 0.19), labels = percent_format(accuracy = 1)) +
28   scale_y_continuous(limits = c(-0.01, 0.03), labels = percent_format(accuracy = 1)) +
29   labs(title = "Frontera media-varianza, 10 activos con ventas en corto y 2 carteras.",
30     x = "Volatilidad", y = "Rendimiento esperado") +
31   guides(color = "none") +
32   theme_minimal()
```

Frontera media–varianza, 10 activos con ventas en corto y 2 carteras.



## 23 Riesgo-rendimiento: 10 activos individuales y carteras.

```

1 weights_50_50 <- c(LRCX = 0.5, GIS = 0.5)
2 custom_weights <- c(LRCX = 0.1990055, GIS = 0.8009945)
3 portfolio_label <- "19.90% LRCX / 80.10% GIS"
4
5 weights_1N <- setNames(rep(1 / n_assets, n_assets), names(mu_vec_all))
6
7 weights_specs <- list("50/50 LRCX-GIS" = weights_50_50,
8                       "19.90% LRCX / 80.10% GIS" = custom_weights,
9                       "MV 10-Asset (No Shorts)" = weights_minvar10_ns,
10                      "MV 10-Asset (Shorts Allowed)" = weights_minvar10_short,
11                      "1/N" = weights_1N)
12
13 portfolio_stats_ext <- imap_dfr(weights_specs, ~ portfolio_stats(monthly_returns, .x, .y))
14
15 final_table <- bind_rows(asset_stats(monthly_returns), portfolio_stats_ext) |>
16   mutate(symbol = factor(symbol, levels = c(tickers, names(weights_specs)))) |>
17   arrange(symbol)
18
19 print(final_table)

```

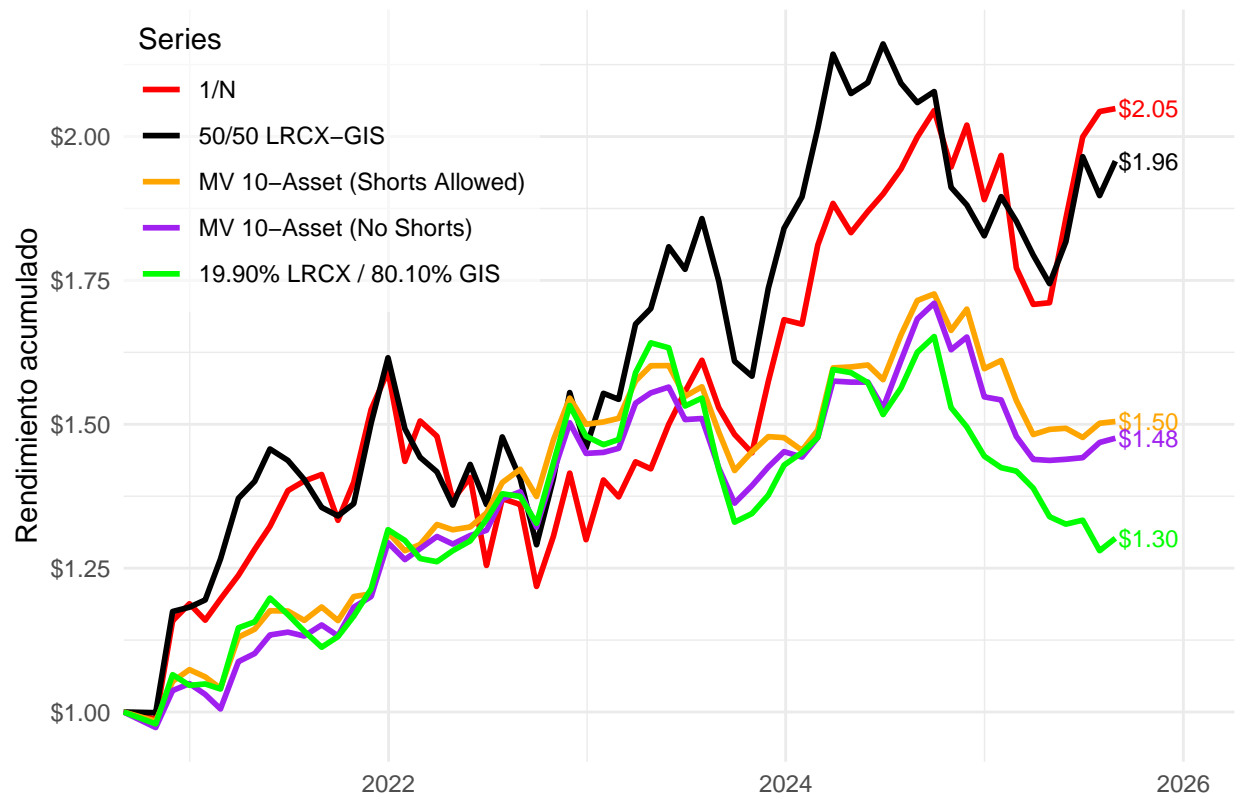
## # A tibble: 15 x 4

##	symbol	ER	SD	SR
##	<fct>	<dbl>	<dbl>	<dbl>
##	1 AMD	0.0223	0.150	0.149
##	2 CNC	-0.00542	0.102	-0.0532
##	3 GIS	0.000182	0.0477	0.00382
##	4 LMT	0.00709	0.0647	0.110
##	5 LRCX	0.0257	0.111	0.231
##	6 NEM	0.0107	0.103	0.103
##	7 SNPS	0.0216	0.0902	0.240
##	8 SYF	0.0259	0.106	0.244
##	9 TRMB	0.0132	0.0970	0.136
##	10 TTD	0.0170	0.184	0.0926
##	11 50/50 LRCX-GIS	0.0130	0.0560	0.231
##	12 19.90% LRCX / 80.10% GIS	0.00526	0.0401	0.131
##	13 MV 10-Asset (No Shorts)	0.00717	0.0337	0.213
##	14 MV 10-Asset (Shorts Allowed)	0.00747	0.0329	0.227
##	15 1/N	0.0138	0.0569	0.243

## 24 Rendimiento acumulado, todas las carteras.

```
1 portfolio_cum <- imap_dfr(weights_specs,
2                             ~ portfolio_series(monthly_returns, .x, .y, price_start))
3
4 # Último valor real de cada serie (usa la fecha máxima, no la última fila)
5 last_values_port <- portfolio_cum |>
6   group_by(symbol) |>
7   slice_max(order_by = date, n = 1, with_ties = FALSE) |>
8   select(symbol, final_value = index_level) |>
9   arrange(desc(final_value))
10
11 # Orden de series según rendimiento final (mayor a menor)
12 series_levels <- last_values_port$symbol
13
14 palette_base <- c("50/50 LRCX-GIS" = "black",
15                  "19.90% LRCX / 80.10% GIS" = "green",
16                  "MV 10-Asset (No Shorts)" = "purple",
17                  "MV 10-Asset (Shorts Allowed)" = "orange",
18                  "1/N" = "red")
19
20 portfolio_colors <- palette_base[series_levels]
21
22 portfolio_cum <- portfolio_cum |>
23   mutate(symbol = factor(symbol, levels = series_levels))
24
25 label_data_port <- portfolio_cum |>
26   group_by(symbol) |>
27   slice_max(order_by = date, n = 1, with_ties = FALSE) |>
28   mutate(label = paste0("$", formatC(index_level, format = "f", digits = 2))) |>
29   ungroup()
30
31 ggplot(portfolio_cum, aes(date, index_level, color = symbol)) +
32   geom_line(linewidth = 1) +
33   geom_text(data = label_data_port, aes(label = label),
34             hjust = -0.05, vjust = 0.5, size = 3.2, show.legend = FALSE) +
35   scale_color_manual(values = portfolio_colors, breaks = series_levels,
36                     limits = series_levels, drop = FALSE) +
37   scale_x_date(expand = expansion(mult = c(0, 0.12))) +
38   scale_y_continuous(labels = dollar_format(prefix = "$")) +
39   labs(title = "Rendimiento acumulado.",
40        y = "Rendimiento acumulado", x = NULL, color = "Series") +
41   theme_minimal() +
42   theme(legend.position = c(0, 1), legend.justification = c(0, 1),
43         legend.background = element_rect(fill = alpha("white", 0.6), color = NA))
```

## Rendimiento acumulado.



## 25 Conclusión.

- La diversificación reduce riesgo y permite construir portafolios con mejor relación riesgo–rendimiento que los activos individuales.
- La frontera media–varianza es una herramienta poderosa para visualizar las decisiones de inversión y comprender el intercambio entre riesgo y retorno.