

Carteras.

xxxx.

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

21 de noviembre de 2025, 01:04 a.m.

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

1 Introducción.

- Estimar la volatilidad no condicional de los rendimientos del índice S&P 500, utilizando datos históricos de precios diarios.
- Rendimientos como cambios porcentuales y logaritmos naturales.
- Base para futuros modelos financieros: Black–Scholes, Markowitz, VaR, Monte Carlo, Random Walk, Geometric Brownian Motion, Vasicek, riesgo de crédito de Merton, CAPM, APT.

2 Paquetes.

Ticker	Nombre de la empresa	Industria
AMD	Advanced Micro Devices, Inc.	Semiconductores / Computación de alto rendimiento
CNC	Centene Corporation	Servicios de salud / Seguros de salud administrados
GIS	General Mills, Inc.	Alimentación / Productos de consumo envasados
LMT	Lockheed Martin Corporation	Aeroespacial y defensa
LRCX	Lam Research Corporation	Equipos para fabricación de semiconductores
NEM	Newmont Corporation	Minería de metales preciosos (oro, cobre, etc.)
SNPS	Synopsys, Inc.	Software de automatización de diseño electrónico (EDA)
SYF	Synchrony Financial	Servicios financieros al consumidor / Crédito al consumo
TRMB	Trimble Inc.	Tecnología geoespacial, construcción y transporte
TTD	The Trade Desk, Inc.	Tecnología publicitaria / Plataforma de compra de anuncios

```

1 library(tidyquant)
2 library(tidyverse)
3 library(lubridate)
4
5 tickers      <- c("AMD", "CNC", "GIS", "LMT", "LRCX", "NEM", "SNPS", "SYF", "TRMB", "TTD")
6 n_months    <- 60L
7 price_start  <- ymd("2020-09-01")                # un mes extra para generar los retornos
8 price_end    <- price_start %m+% months(n_months + 1) - days(1)
9 returns_start <- price_start %m+% months(1)
10 returns_end  <- returns_start %m+% months(n_months) - months(1)
11
12 prices <- tq_get(
13   tickers,
14   from = price_start,
15   to   = price_end,
16   get  = "stock.prices"
17 )
18

```

```

19 monthly_returns <- prices %>%
20   arrange(symbol, date) %>%
21   group_by(symbol) %>%
22   tq_transmute(
23     select      = adjusted,
24     mutate_fun  = periodReturn,
25     period      = "monthly",
26     type        = "arithmetic",
27     col_rename  = "monthly_return"
28   ) %>%
29   filter(between(date, returns_start, returns_end)) %>%
30   mutate(obs = row_number()) %>%
31   filter(obs <= n_months) %>%
32   select(-obs) %>%
33   ungroup()
34
35 stopifnot(n_distinct(monthly_returns$symbol) == length(tickers))
36
37 stats <- monthly_returns %>%
38   group_by(symbol) %>%
39   summarise(
40     expected_return = mean(monthly_return),
41     sd_return       = sd(monthly_return),
42     sharpe_ratio    = expected_return / sd_return,
43     .groups = "drop"
44   )
45
46 returns_wide <- monthly_returns %>%
47   select(date, symbol, monthly_return) %>%
48   pivot_wider(names_from = symbol, values_from = monthly_return) %>%
49   arrange(date)
50
51 corr_mat <- returns_wide %>%
52   select(-date) %>%
53   cor(use = "pairwise.complete.obs")
54
55 lowest_pairs <- corr_mat %>%
56   as_tibble(rownames = "symbol_i") %>%
57   pivot_longer(-symbol_i, names_to = "symbol_j", values_to = "corr") %>%
58   filter(symbol_i < symbol_j) %>%
59   arrange(corr) %>%
60   slice_head(n = 10)
61
62 print(stats)

```

A tibble: 10 x 4

	symbol	expected_return	sd_return	sharpe_ratio
	<chr>	<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
```

```
1 corr_mat_fmt <- corr_mat %>%
2   round(2) %>% # 2 decimales bastan para correlaciones
3   format(trim = TRUE, nsmall = 2) %>% # remueve espacios extra y alinea
4   as.matrix()
5
6 print(corr_mat_fmt, 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
```

```
1 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
```

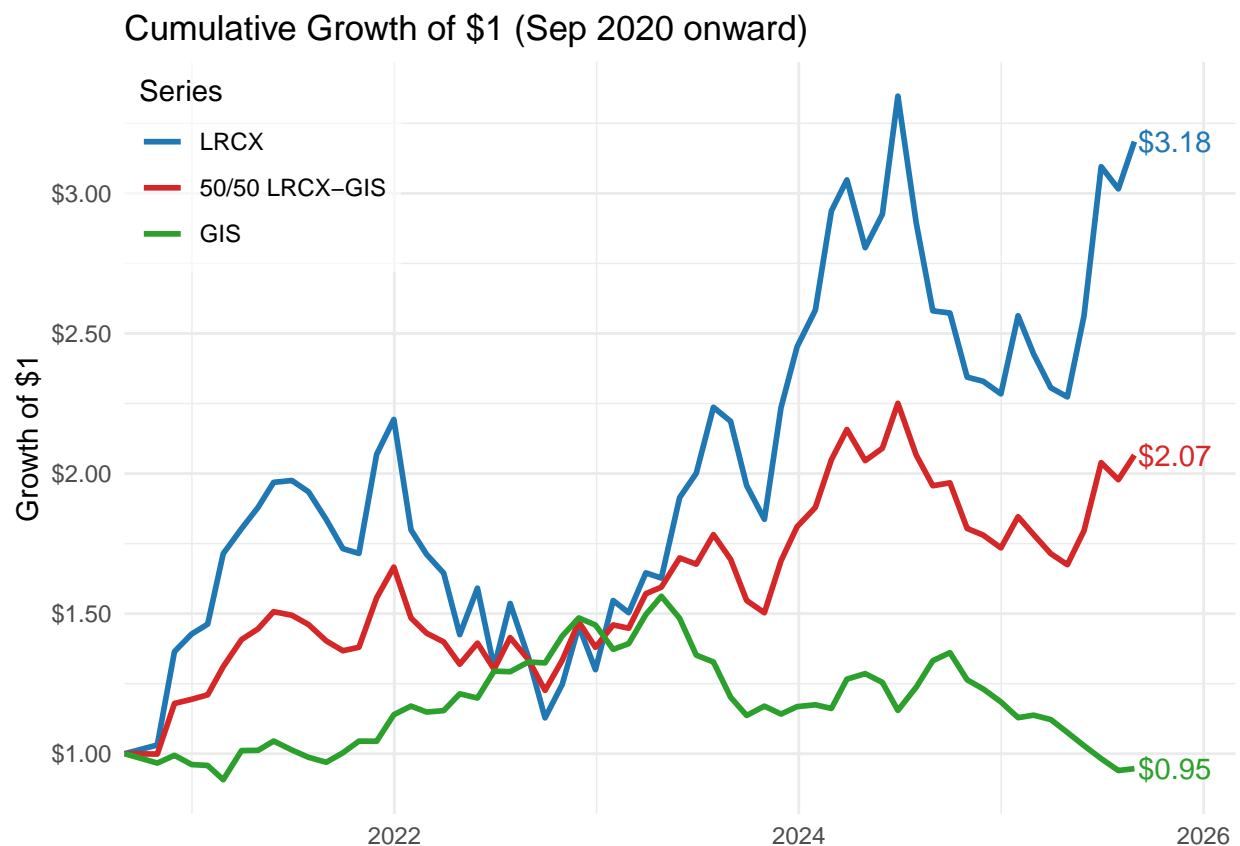
10 GIS SYF -0.148

```
1 library(scales)
2
3 focus_symbols <- c("LRCX", "GIS")
4 base_rows <- tibble(
5   date = price_start,
6   symbol = focus_symbols,
7   monthly_return = NA_real_,
8   index_level = 1
9 )
10
11 cum_returns <- monthly_returns %>%
12   filter(symbol %in% focus_symbols) %>%
13   arrange(symbol, date) %>%
14   group_by(symbol) %>%
15   mutate(index_level = cumprod(1 + monthly_return)) %>%
16   ungroup() %>%
17   bind_rows(base_rows) %>%
18   arrange(symbol, date) %>%
19   select(date, symbol, index_level)
20
21 portfolio <- cum_returns %>%
22   pivot_wider(names_from = symbol, values_from = index_level) %>%
23   mutate(`50/50 LRCX-GIS` = 0.5 * LRCX + 0.5 * GIS) %>%
24   select(date, `50/50 LRCX-GIS`) %>%
25   pivot_longer(-date, names_to = "symbol", values_to = "index_level")
26
27 plot_data <- bind_rows(cum_returns, portfolio)
28
29 last_values <- plot_data %>%
30   arrange(date) %>%
31   group_by(symbol) %>%
32   summarise(final_value = last(index_level), .groups = "drop") %>%
33   arrange(desc(final_value))
34
35 color_values <- c("#1f77b4", "#d62728", "#2ca02c")
36 names(color_values) <- last_values$symbol
37
38 plot_data <- plot_data %>%
39   mutate(symbol = factor(symbol, levels = last_values$symbol))
40
41 label_data <- plot_data %>%
42   group_by(symbol) %>%
43   filter(date == max(date)) %>%
44   ungroup() %>%
45   mutate(label = paste0("$", formatC(index_level, format = "f", digits = 2)))
46
47 ggplot(plot_data, aes(x = date, y = index_level, color = symbol)) +
48   geom_line(size = 1) +
49   geom_text(
50     data = label_data,
51     aes(label = label),
52     hjust = -0.05,
53     vjust = 0.5,
54     show.legend = FALSE
55   ) +
56   scale_color_manual(values = color_values, limits = last_values$symbol) +
57   scale_x_date(expand = expansion(mult = c(0, 0.1))) +
```

```

58   scale_y_continuous(labels = dollar_format(prefix = "$")) +
59   labs(
60     title = "Cumulative Growth of $1 (Sep 2020 onward)",
61     y = "Growth of $1",
62     x = NULL,
63     color = "Series"
64   ) +
65   theme_minimal() +
66   theme(
67     legend.position = c(0, 1),
68     legend.justification = c(0, 1),
69     legend.background = element_rect(fill = alpha("white", 0.6), color = NA)
70   )

```



```

1   weights <- c(LRCX = 0.5, GIS = 0.5)
2
3   stats_lrcx_gis <- monthly_returns %>%
4     filter(symbol %in% names(weights)) %>%
5     group_by(symbol) %>%
6     summarise(
7       expected_return = mean(monthly_return),
8       sd_return       = sd(monthly_return),
9       sharpe_ratio    = expected_return / sd_return,
10      .groups = "drop"
11    )
12
13  portfolio_stats <- monthly_returns %>%

```

```

14   filter(symbol %in% names(weights)) %>%
15   mutate(weight = weights[symbol]) %>%
16   group_by(date) %>%
17   summarise(portfolio_return = sum(monthly_return * weight), .groups = "drop") %>%
18   summarise(
19     symbol      = "50/50 LRCX-GIS",
20     expected_return = mean(portfolio_return),
21     sd_return    = sd(portfolio_return),
22     sharpe_ratio  = expected_return / sd_return
23   )
24
25   final_table <- bind_rows(stats_lrcx_gis, portfolio_stats) %>%
26   mutate(symbol = factor(symbol, levels = last_values$symbol)) %>%
27   arrange(symbol)
28
29   print(final_table)

```

A tibble: 3 x 4

	symbol	expected_return	sd_return	sharpe_ratio
	<fct>	<dbl>	<dbl>	<dbl>
## 1	LRCX	0.0257	0.111	0.231
## 2	50/50 LRCX-GIS	0.0130	0.0560	0.231
## 3	GIS	0.000182	0.0477	0.00382

```

1   returns_matrix <- monthly_returns %>%
2   filter(symbol %in% names(weights)) %>%
3   select(date, symbol, monthly_return) %>%
4   pivot_wider(names_from = symbol, values_from = monthly_return) %>%
5   arrange(date) %>%
6   select(all_of(names(weights)))
7
8   cov_mat <- cov(returns_matrix, use = "pairwise.complete.obs")
9
10  mu_vec <- stats_lrcx_gis %>%
11  select(symbol, expected_return) %>%
12  deframe()
13  sd_vec <- stats_lrcx_gis %>%
14  select(symbol, sd_return) %>%
15  deframe()
16
17  frontier_data <- tibble(weight_lrcx = seq(0, 1, length.out = 200)) %>%
18  mutate(
19    weight_gis      = 1 - weight_lrcx,
20    expected_return = weight_lrcx * mu_vec["LRCX"] + weight_gis * mu_vec["GIS"],
21    variance = weight_lrcx^2 * sd_vec["LRCX"]^2 +
22              weight_gis^2 * sd_vec["GIS"]^2 +
23              2 * weight_lrcx * weight_gis * cov_mat["LRCX", "GIS"],
24    sd_return = sqrt(variance)
25  )
26  # sin arrange() para preservar el orden de los pesos
27
28  portfolio_label <- "19.90% LRCX / 80.10% GIS"
29  custom_weights <- c(LRCX = 0.1990055, GIS = 0.8009945)
30

```

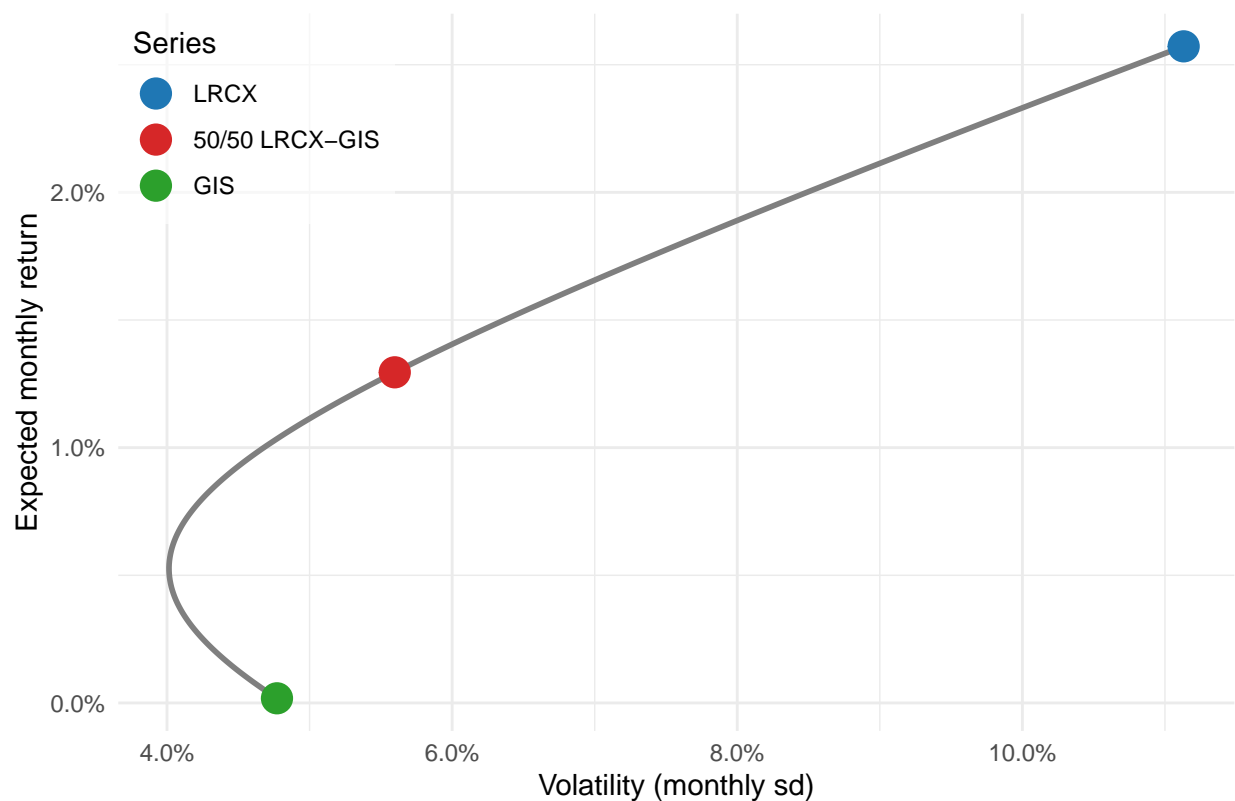


```

31   series_order <- c("LRCX", "GIS", "50/50 LRCX-GIS", portfolio_label)
32 mean_var_points <- final_table %>%
33   filter(symbol %in% series_order) %>%           # deja solo los dos activos + dos carteras
34   mutate(symbol = factor(symbol, levels = series_order)) %>%
35   arrange(symbol)
36
37
38 ggplot() +
39   geom_path(
40     data = frontier_data,
41     aes(x = sd_return, y = expected_return),
42     color = "grey50",
43     linewidth = 1
44   ) +
45   geom_point(
46     data = mean_var_points,
47     aes(x = sd_return, y = expected_return, color = symbol),
48     size = 5
49   ) +
50   scale_color_manual(values = color_values, limits = last_values$symbol) +
51   scale_x_continuous(labels = percent_format(accuracy = 0.1)) +
52   scale_y_continuous(labels = percent_format(accuracy = 0.1)) +
53   labs(
54     title = "Mean-Variance View (Sep 2020 onward)",
55     x = "Volatility (monthly sd)",
56     y = "Expected monthly return",
57     color = "Series"
58   ) +
59   theme_minimal() +
60   theme(
61     legend.position = c(0, 1),
62     legend.justification = c(0, 1),
63     legend.background = element_rect(fill = alpha("white", 0.6), color = NA)
64   )

```

Mean–Variance View (Sep 2020 onward)



```

1 library(quadprog)
2
3 assets <- names(weights)
4 cov_two_assets <- cov_mat[assets, assets]
5
6 Dmat <- 2 * cov_two_assets
7 dvec <- rep(0, length(assets))
8 Amat <- matrix(1, nrow = length(assets), ncol = 1) # 1'w = 1
9 bvec <- 1
10
11 qp <- solve.QP(Dmat, dvec, Amat, bvec, meq = 1)
12 weights_opt <- setNames(qp$solution, assets)
13
14 print(weights_opt)

```

```

##          LRCX          GIS
## 0.1990059 0.8009941

```

```

1 library(scales)
2
3 focus_symbols <- c("LRCX", "GIS")
4 custom_weights <- c(LRCX = 0.1990055, GIS = 0.8009945)
5 portfolio_label <- "19.90% LRCX / 80.10% GIS"
6
7 base_rows <- tibble(

```

```

8     date = price_start,
9     symbol = focus_symbols,
10    monthly_return = NA_real_,
11    index_level = 1
12  )
13
14  cum_returns <- monthly_returns %>%
15    filter(symbol %in% focus_symbols) %>%
16    arrange(symbol, date) %>%
17    group_by(symbol) %>%
18    mutate(index_level = cumprod(1 + monthly_return)) %>%
19    ungroup() %>%
20    bind_rows(base_rows) %>%
21    arrange(symbol, date) %>%
22    select(date, symbol, index_level)
23
24  portfolio <- cum_returns %>%
25    pivot_wider(names_from = symbol, values_from = index_level) %>%
26    mutate("{portfolio_label}" := custom_weights["LRCX"] * LRCX +
27          custom_weights["GIS"] * GIS) %>%
28    select(date, all_of(portfolio_label)) %>%
29    pivot_longer(-date, names_to = "symbol", values_to = "index_level")
30
31  plot_data <- bind_rows(cum_returns, portfolio)
32
33  last_values <- plot_data %>%
34    arrange(date) %>%
35    group_by(symbol) %>%
36    summarise(final_value = last(index_level), .groups = "drop") %>%
37    arrange(desc(final_value))
38
39  color_values <- c("#1f77b4", "#d62728", "#2ca02c")
40  names(color_values) <- last_values$symbol
41
42  plot_data <- plot_data %>%
43    mutate(symbol = factor(symbol, levels = last_values$symbol))
44
45  label_data <- plot_data %>%
46    group_by(symbol) %>%
47    filter(date == max(date)) %>%
48    ungroup() %>%
49    mutate(label = paste0("$", formatC(index_level, format = "f", digits = 2)))
50
51  ggplot(plot_data, aes(x = date, y = index_level, color = symbol)) +
52    geom_line(size = 1) +
53    geom_text(
54      data = label_data,
55      aes(label = label),
56      hjust = -0.05,
57      vjust = 0.5,
58      show.legend = FALSE
59    ) +
60    scale_color_manual(values = color_values, limits = last_values$symbol) +
61    scale_x_date(expand = expansion(mult = c(0, 0.1))) +
62    scale_y_continuous(labels = dollar_format(prefix = "$")) +
63    labs(
64      title = "Cumulative Growth of $1 (Sep 2020 onward)",
65      y = "Growth of $1",
66      x = NULL,

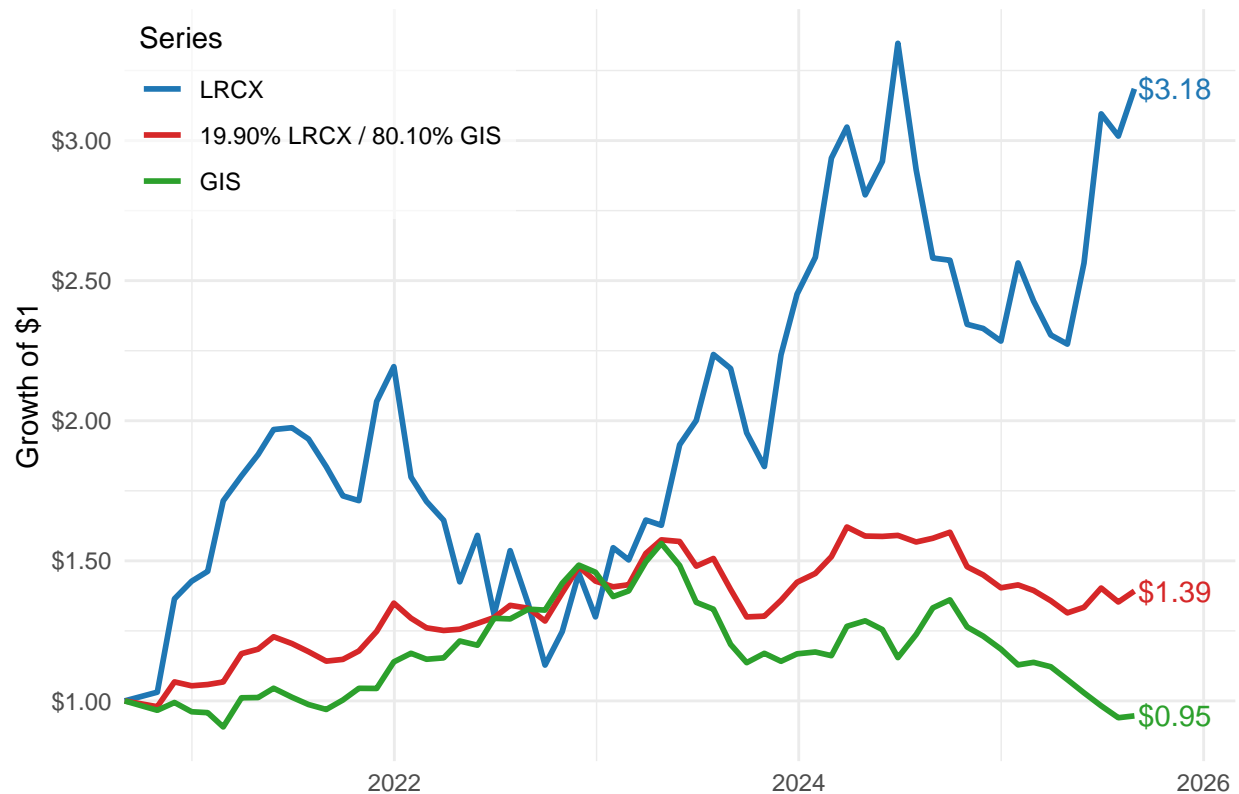
```

```

67   color = "Series"
68 ) +
69 theme_minimal() +
70 theme(
71   legend.position = c(0, 1),
72   legend.justification = c(0, 1),
73   legend.background = element_rect(fill = alpha("white", 0.6), color = NA)
74 )

```

Cumulative Growth of \$1 (Sep 2020 onward)



```

1  custom_portfolio_stats <- monthly_returns %>%
2    filter(symbol %in% names(custom_weights)) %>%
3    mutate(weight = custom_weights[symbol]) %>%
4    group_by(date) %>%
5    summarise(portfolio_return = sum(monthly_return * weight), .groups = "drop") %>%
6    summarise(
7      symbol      = portfolio_label,
8      expected_return = mean(portfolio_return),
9      sd_return    = sd(portfolio_return),
10     sharpe_ratio  = expected_return / sd_return
11   )
12
13 table_levels <- c("LRCX", "GIS", "50/50 LRCX-GIS", portfolio_label)
14
15 final_table <- final_table %>%
16   mutate(symbol = as.character(symbol)) %>%
17   bind_rows(custom_portfolio_stats) %>%
18   mutate(symbol = factor(symbol, levels = table_levels)) %>%

```

```

19   arrange(symbol)
20
21   print(final_table)

```

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

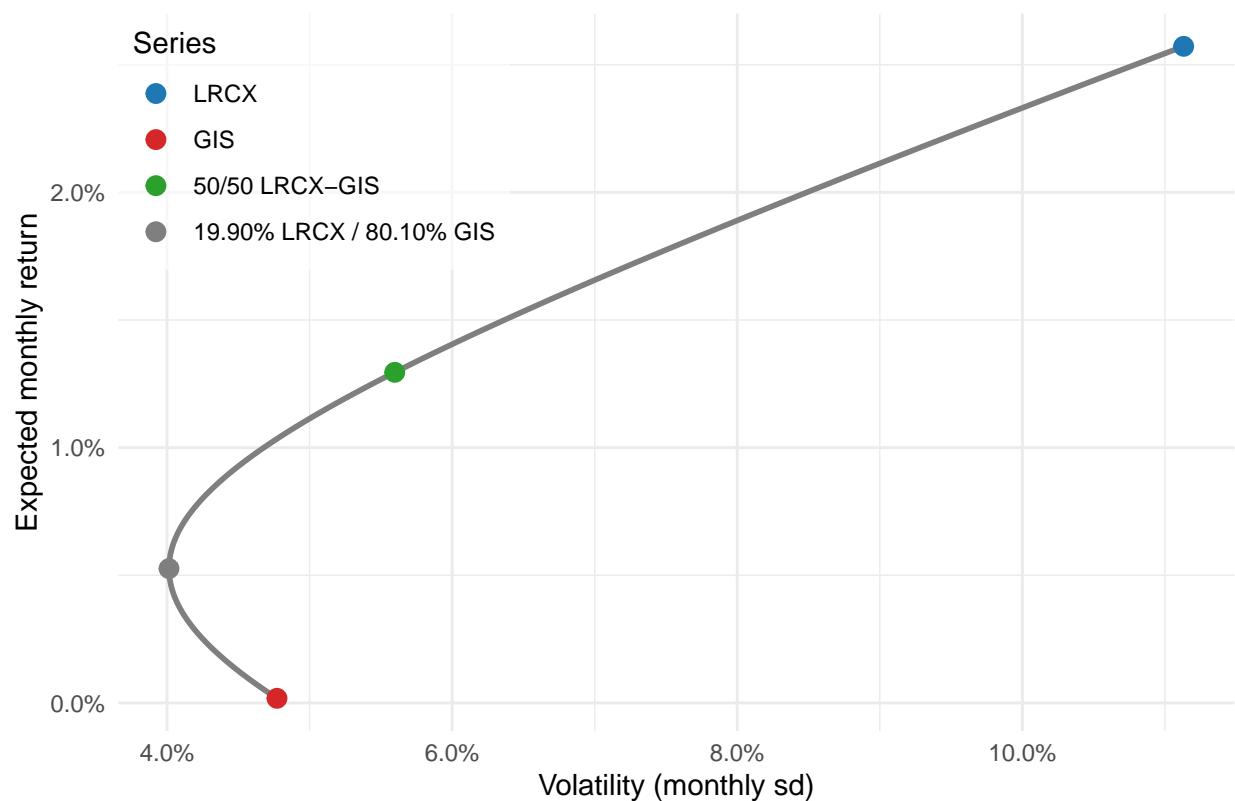
	symbol	expected_return	sd_return	sharpe_ratio
	<fct>	<dbl>	<dbl>	<dbl>
## 1	LRCX	0.0257	0.111	0.231
## 2	GIS	0.000182	0.0477	0.00382
## 3	50/50 LRCX-GIS	0.0130	0.0560	0.231
## 4	19.90% LRCX / 80.10% GIS	0.00526	0.0401	0.131

```

1   table_levels <- c("LRCX", "GIS", "50/50 LRCX-GIS", portfolio_label)
2
3   mean_var_points <- final_table %>%
4     mutate(symbol = factor(symbol, levels = table_levels))
5
6   color_values_mv <- c(
7     "LRCX"           = "#1f77b4",
8     "GIS"            = "#d62728",
9     "50/50 LRCX-GIS" = "#2ca02c",
10    portfolio_label   = "#ff7f0e"
11  )
12
13  ggplot() +
14    geom_path(
15      data = frontier_data,
16      aes(x = sd_return, y = expected_return),
17      color = "grey50",
18      linewidth = 1
19    ) +
20    geom_point(
21      data = mean_var_points,
22      aes(x = sd_return, y = expected_return, color = symbol),
23      size = 3
24    ) +
25    scale_color_manual(values = color_values_mv, limits = table_levels) +
26    scale_x_continuous(labels = percent_format(accuracy = 0.1)) +
27    scale_y_continuous(labels = percent_format(accuracy = 0.1)) +
28    labs(
29      title = "Mean-Variance View (Sep 2020 onward)",
30      x = "Volatility (monthly sd)",
31      y = "Expected monthly return",
32      color = "Series"
33    ) +
34    theme_minimal() +
35    theme(
36      legend.position = c(0, 1),
37      legend.justification = c(0, 1),
38      legend.background = element_rect(fill = alpha("white", 0.6), color = NA)
39    )

```

Mean–Variance View (Sep 2020 onward)



```

1  library(scales)
2
3  focus_symbols <- c("LRCX", "GIS")
4  ci_mult <- 1.96
5
6  base_rows <- tibble(
7    date = price_start,
8    symbol = focus_symbols,
9    monthly_return = NA_real_,
10   index_level = 1
11 )
12
13 cum_returns <- monthly_returns %>%
14   filter(symbol %in% focus_symbols) %>%
15   arrange(symbol, date) %>%
16   group_by(symbol) %>%
17   mutate(index_level = cumprod(1 + monthly_return)) %>%
18   ungroup() %>%
19   bind_rows(base_rows) %>%
20   arrange(symbol, date) %>%
21   select(date, symbol, index_level)
22
23 final_values <- cum_returns %>%
24   group_by(symbol) %>%
25   summarise(final_value = last(index_level), .groups = "drop")
26
27 stats_assets <- monthly_returns %>%
28   filter(symbol %in% focus_symbols) %>%

```

```

29   group_by(symbol) %>%
30   summarise(sd_return = sd(monthly_return), .groups = "drop")
31
32 portfolios <- list(
33   "50/50 LRCX-GIS" = c(LRCX = 0.5, GIS = 0.5)
34 )
35 portfolios[[portfolio_label]] <- custom_weights
36
37 stats_portfolios <- bind_rows(lapply(names(portfolios), function(label) {
38   w <- portfolios[[label]]
39
40   pr <- monthly_returns %>%
41     filter(symbol %in% names(w)) %>%
42     mutate(weight = w[symbol]) %>%
43     group_by(date) %>%
44     summarise(portfolio_return = sum(monthly_return * weight), .groups = "drop") %>%
45     arrange(date)
46
47   final_value <- pr %>%
48     mutate(index_level = cumprod(1 + portfolio_return)) %>%
49     pull(index_level) %>%
50     last()
51
52   tibble(
53     symbol      = label,
54     final_value = final_value,
55     sd_return   = sd(pr$portfolio_return)
56   )
57 })))
58
59 result_table <- bind_rows(
60   left_join(final_values, stats_assets, by = "symbol"),
61   stats_portfolios
62 ) %>%
63   mutate(
64     ci_low = final_value - ci_mult * sd_return,
65     ci_high = final_value + ci_mult * sd_return
66   ) %>%
67   select(symbol, final_value, ci_low, ci_high)
68
69 print(result_table)

```

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

	symbol	final_value	ci_low	ci_high
	<chr>	<dbl>	<dbl>	<dbl>
## 1	GIS	0.947	0.853	1.04
## 2	LRCX	3.18	2.97	3.40
## 3	50/50 LRCX-GIS	1.96	1.85	2.07
## 4	19.90% LRCX / 80.10% GIS	1.30	1.22	1.38

```

1  library(scales)
2
3  base_rows_all <- tibble(
4    date = price_start,
5    symbol = tickers,
6    monthly_return = NA_real_,
7    index_level = 1
8  )
9
10 cum_returns_all <- monthly_returns %>%
11   filter(symbol %in% tickers) %>%
12   arrange(symbol, date) %>%
13   group_by(symbol) %>%
14   mutate(index_level = cumprod(1 + monthly_return)) %>%
15   ungroup() %>%
16   bind_rows(base_rows_all) %>%
17   arrange(symbol, date) %>%
18   select(date, symbol, index_level)
19
20 last_values_all <- cum_returns_all %>%
21   arrange(date) %>%
22   group_by(symbol) %>%
23   summarise(final_value = last(index_level), .groups = "drop") %>%
24   arrange(desc(final_value))
25
26 plot_data_all <- cum_returns_all %>%
27   mutate(symbol = factor(symbol, levels = last_values_all$symbol))
28
29 label_data_all <- plot_data_all %>%
30   group_by(symbol) %>%
31   filter(date == max(date)) %>%
32   ungroup() %>%
33   mutate(label = paste0("$", formatC(index_level, format = "f", digits = 2)))
34
35 color_values_all <- setNames(scales::hue_pal()(length(tickers)), last_values_all$symbol)
36
37 ggplot(plot_data_all, aes(x = date, y = index_level, color = symbol)) +
38   geom_line(linewidth = 1) +
39   geom_text(
40     data = label_data_all,
41     aes(label = label),
42     hjust = -0.05,
43     vjust = 0.5,
44     size = 3,
45     show.legend = FALSE,
46     check_overlap = TRUE
47   ) +
48   scale_color_manual(values = color_values_all, limits = levels(plot_data_all$symbol)) +
49   scale_x_date(expand = expansion(mult = c(0, 0.12))) +
50   scale_y_continuous(labels = dollar_format(prefix = "$")) +
51   labs(
52     title = "Cumulative Growth of $1 - All 10 Tickers (Sep 2020 onward)",
53     y = "Growth of $1",
54     x = NULL,
55     color = "Ticker"
56   ) +
57   theme_minimal() +
58   theme(
59     legend.position = c(0, 1),

```

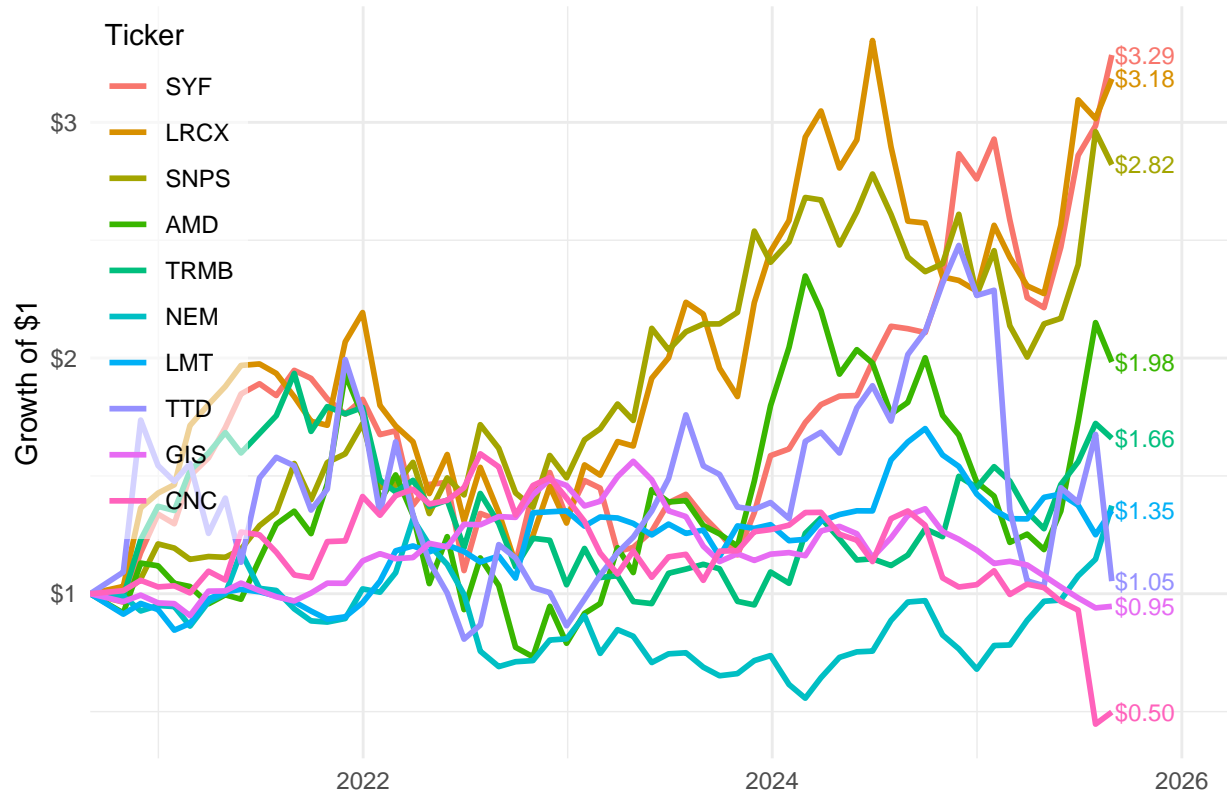


```

60     legend.justification = c(0, 1),
61     legend.background = element_rect(fill = alpha("white", 0.6), color = NA)
62 )

```

Cumulative Growth of \$1 — All 10 Tickers (Sep 2020 onward)



```

1  end_date <- max(cum_returns_all$date)
2
3  segment_data <- last_values_all %>%
4    mutate(
5      start_date = price_start,
6      start_value = 1,
7      end_date = end_date
8    )
9
10 segment_plot <- ggplot() +
11   geom_segment(
12     data = segment_data,
13     aes(
14       x = start_date,
15       xend = end_date,
16       y = start_value,
17       yend = final_value,
18       color = symbol
19     ),
20     linewidth = 1
21   ) +
22   geom_point(
23     data = segment_data,

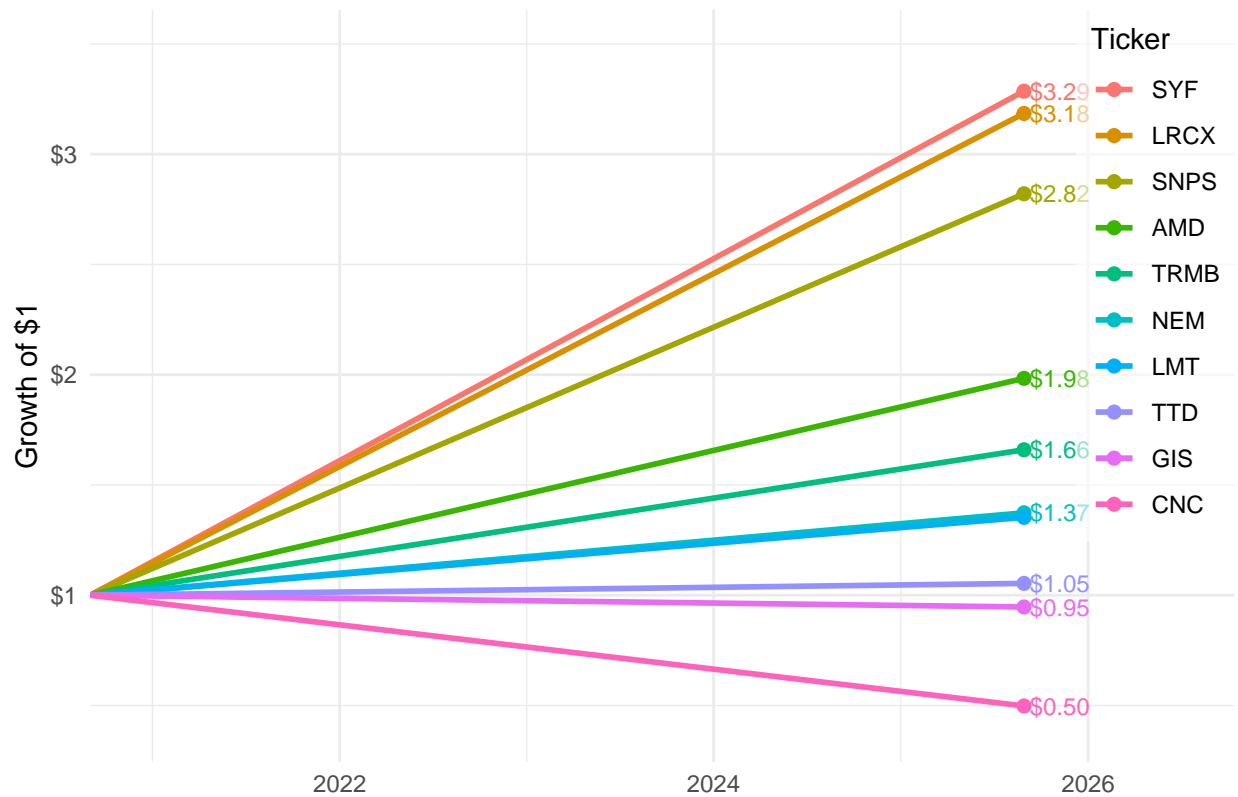
```

```

24     aes(x = end_date, y = final_value, color = symbol),
25     size = 2
26 ) +
27 geom_text(
28   data = segment_data,
29   aes(
30     x = end_date,
31     y = final_value,
32     label = paste0("$", formatC(final_value, format = "f", digits = 2)),
33     color = symbol
34   ),
35   nudge_x = 70,                # desplaza texto ~1 mes a la derecha
36   vjust = 0.5,
37   size = 3,
38   show.legend = FALSE,
39   check_overlap = TRUE
40 ) +
41 scale_color_manual(values = color_values_all, limits = last_values_all$symbol) +
42 scale_x_date(expand = expansion(mult = c(0, 0.18))) +
43 scale_y_continuous(
44   limits = c(0.4, 3.5),
45   labels = dollar_format(prefix = "$")
46 ) +
47 labs(
48   title = "Start vs Final Growth of $1 - All 10 Tickers",
49   x = NULL,
50   y = "Growth of $1",
51   color = "Ticker"
52 ) +
53 theme_minimal() +
54 theme(
55   legend.position = c(1, 1),
56   legend.justification = c(1, 1),
57   legend.background = element_rect(fill = alpha("white", 0.6), color = NA)
58 )
59
60 segment_plot

```

Start vs Final Growth of \$1 — All 10 Tickers



```

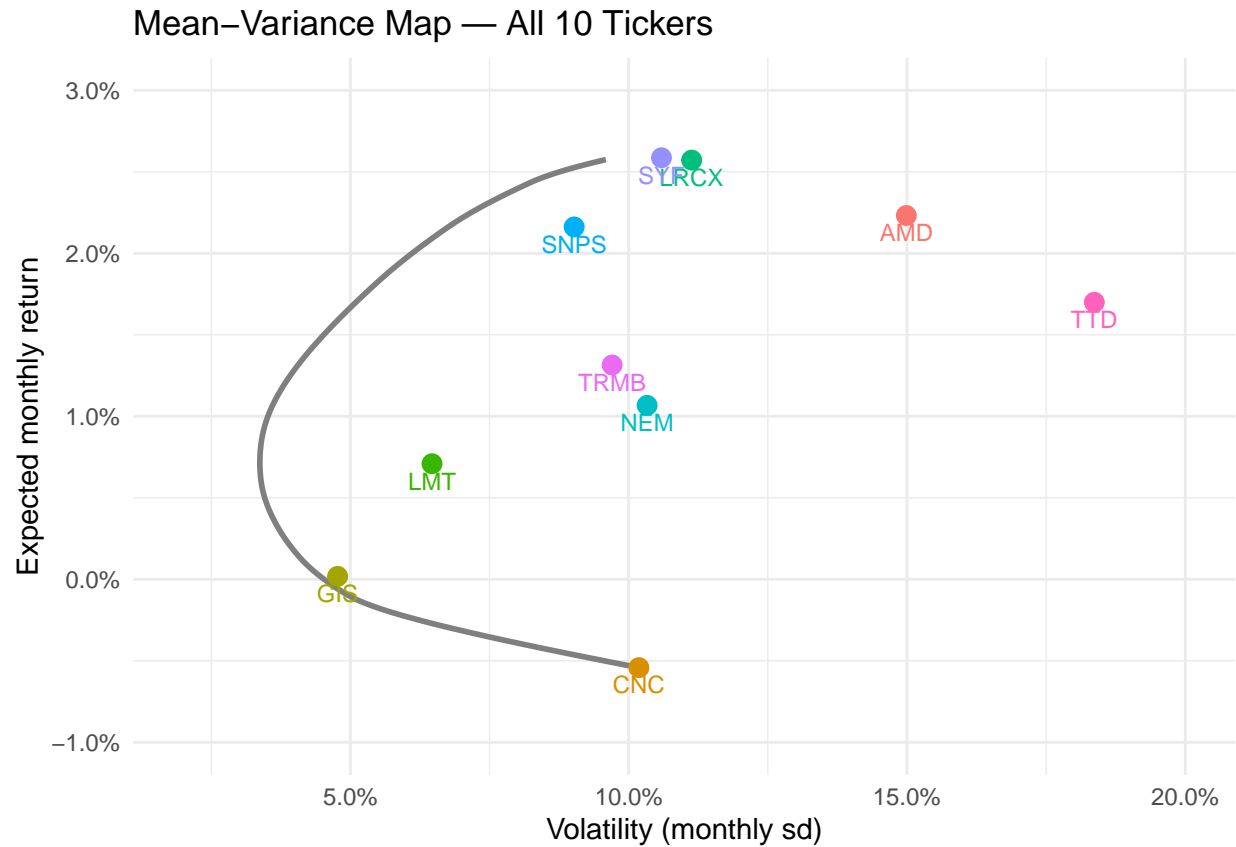
1 library(quadprog)
2
3 returns_wide_all <- monthly_returns %>%
4   filter(symbol %in% tickers) %>%
5   select(date, symbol, monthly_return) %>%
6   pivot_wider(names_from = symbol, values_from = monthly_return) %>%
7   arrange(date)
8
9 returns_matrix <- returns_wide_all %>%
10  select(-date) %>%
11  as.matrix()
12
13 mu_vec <- colMeans(returns_matrix, na.rm = TRUE)
14 cov_mat_all <- cov(returns_matrix, use = "pairwise.complete.obs")
15
16 stats_all <- tibble(
17   symbol = names(mu_vec),
18   expected_return = mu_vec,
19   sd_return = apply(returns_matrix, 2, sd, na.rm = TRUE)
20 )
21
22 target_returns <- seq(
23   from = min(mu_vec),
24   to = max(mu_vec) + .0001, # empuja la frontera hacia retornos más altos
25   length.out = 150
26 )
27
28 Amat_base <- rbind(

```

```

29     rep(1, length(mu_vec)),
30     mu_vec,
31     diag(length(mu_vec))
32   )
33
34   frontier_points <- map_dfr(target_returns, function(tr) {
35     bvec <- c(1, tr, rep(0, length(mu_vec)))
36     sol <- tryCatch(
37       solve.QP(2 * cov_mat_all, rep(0, length(mu_vec)), t(Amat_base), bvec, meq = 2),
38       error = function(e) NULL
39     )
40     if (is.null(sol)) return(NULL)
41
42     tibble(
43       expected_return = tr,
44       sd_return = sqrt(as.numeric(t(sol$solution) %*% cov_mat_all %*% sol$solution))
45     )
46   }) %>% drop_na()
47
48   color_values_all <- setNames(scales::hue_pal()(length(tickers)), stats_all$symbol)
49
50   ggplot() +
51     geom_path(
52       data = frontier_points,
53       aes(x = sd_return, y = expected_return),
54       color = "grey50",
55       linewidth = 1
56     ) +
57     geom_point(
58       data = stats_all,
59       aes(x = sd_return, y = expected_return, color = symbol),
60       size = 3
61     ) +
62     geom_text(
63       data = stats_all,
64       aes(x = sd_return, y = expected_return, label = symbol, color = symbol),
65       vjust = 1.5,
66       size = 3,
67       show.legend = FALSE
68     ) +
69     scale_color_manual(values = color_values_all, limits = stats_all$symbol) +
70     scale_x_continuous(
71       limits = c(0.02, 0.2),
72       labels = scales::percent_format(accuracy = 0.1)
73     ) +
74     scale_y_continuous(labels = scales::percent_format(accuracy = 0.1),
75                       limits = c(-0.01, 0.03)) +
76     labs(
77       title = "Mean-Variance Map - All 10 Tickers",
78       x = "Volatility (monthly sd)",
79       y = "Expected monthly return"
80     ) +
81     guides(color = "none") +
82     theme_minimal() +
83     theme(legend.position = "none")

```



```

1  library(quadprog)
2
3  assets_all <- tickers
4  n_assets <- length(assets_all)
5
6  Dmat <- 2 * cov_mat_all
7  dvec <- rep(0, n_assets)
8
9  Amat <- cbind(
10     rep(1, n_assets),      # suma de pesos = 1 (meq = 1)
11     diag(n_assets)         # pesos 0
12 )
13 bvec <- c(1, rep(0, n_assets))
14
15 qp_mv <- solve.QP(Dmat, dvec, Amat, bvec, meq = 1)
16 weights_mv <- setNames(qp_mv$solution, assets_all)
17
18 min_var_weights <- tibble(
19   symbol = assets_all,
20   weight = round(weights_mv, 4)
21 ) %>%
22   arrange(desc(weight))
23
24 print(min_var_weights)

```

```
## # A tibble: 10 x 2
```

```
##      symbol weight
##      <chr>      <dbl>
## 1 GIS          0.540
## 2 SNPS          0.183
## 3 LMT           0.112
## 4 TRMB          0.0462
## 5 NEM           0.0452
## 6 SYF           0.0305
## 7 TTD           0.03
## 8 CNC           0.013
## 9 AMD           0
## 10 LRCX         0
```

```
1 library(quadprog)
2
3 returns_wide_all <- monthly_returns %>%
4   filter(symbol %in% tickers) %>%
5   select(date, symbol, monthly_return) %>%
6   pivot_wider(names_from = symbol, values_from = monthly_return) %>%
7   arrange(date)
8
9 returns_matrix <- returns_wide_all %>%
10  select(-date) %>%
11  as.matrix()
12
13 mu_vec <- colMeans(returns_matrix, na.rm = TRUE)
14 cov_mat_all <- cov(returns_matrix, use = "pairwise.complete.obs")
15
16 stats_all <- tibble(
17   symbol = names(mu_vec),
18   expected_return = mu_vec,
19   sd_return = apply(returns_matrix, 2, sd, na.rm = TRUE)
20 )
21
22 target_returns <- seq(
23   from = min(mu_vec),
24   to = max(mu_vec) + 0.0001,
25   length.out = 150
26 )
27
28 Amat_base <- rbind(
29   rep(1, length(mu_vec)),
30   mu_vec,
31   diag(length(mu_vec))
32 )
33
34 frontier_points <- map_dfr(target_returns, function(tr) {
35   bvec <- c(1, tr, rep(0, length(mu_vec)))
36   sol <- tryCatch(
37     solve.QP(2 * cov_mat_all, rep(0, length(mu_vec)), t(Amat_base), bvec, meq = 2),
38     error = function(e) NULL
39   )
40 })
```

```

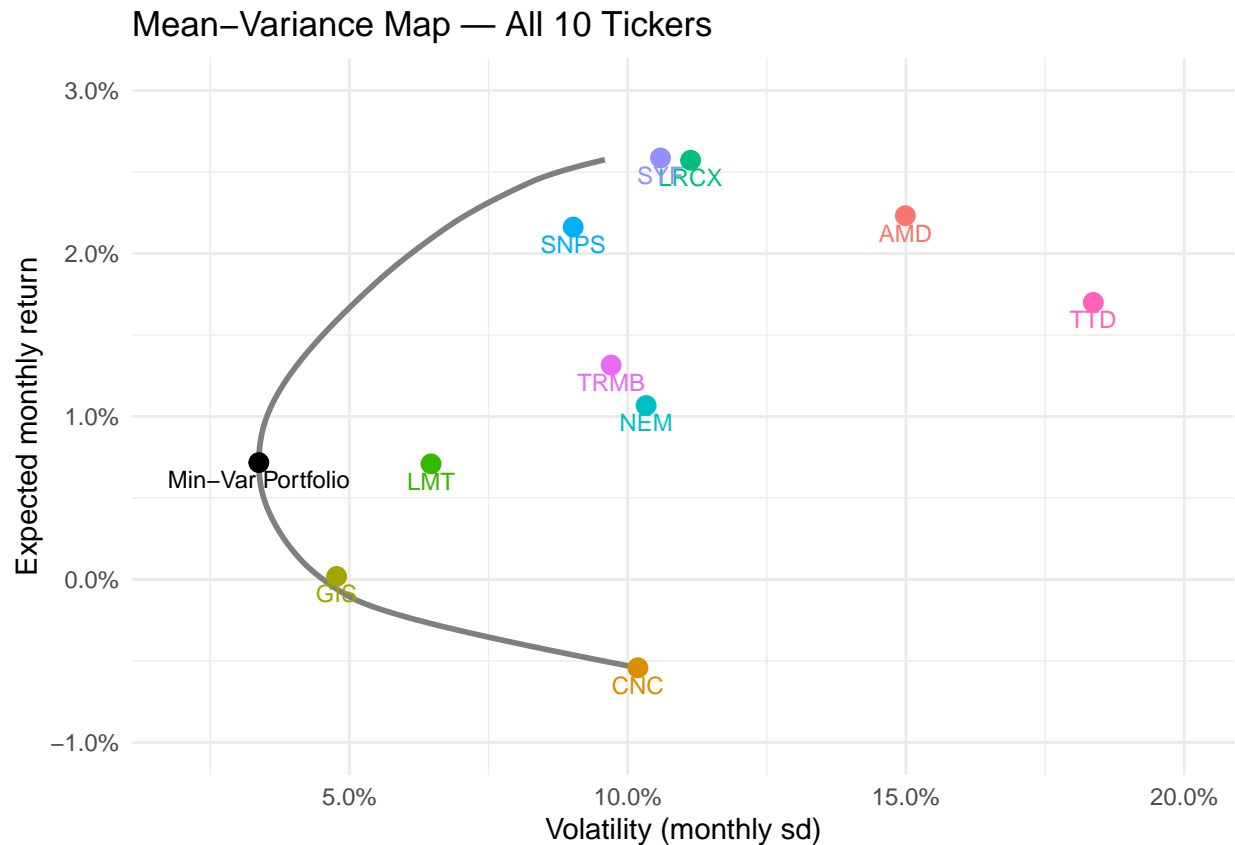
40   if (is.null(sol)) return(NULL)
41
42   tibble(
43     expected_return = tr,
44     sd_return = sqrt(as.numeric(t(sol$solution) %*% cov_mat_all %*% sol$solution))
45   )
46 } %>% drop_na()
47
48 weights_mv <- solve.QP(
49   2 * cov_mat_all,
50   rep(0, length(mu_vec)),
51   cbind(rep(1, length(mu_vec)), diag(length(mu_vec))),
52   c(1, rep(0, length(mu_vec))),
53   meq = 1
54 )$solution
55
56 min_var_point <- tibble(
57   symbol = "Min-Var Portfolio",
58   expected_return = sum(weights_mv * mu_vec),
59   sd_return = sqrt(as.numeric(t(weights_mv) %*% cov_mat_all %*% weights_mv))
60 )
61
62 stats_all_ext <- bind_rows(stats_all, min_var_point)
63
64 color_values_all <- setNames(scales::hue_pal()(length(tickers)), stats_all$symbol)
65 color_values_ext <- c(color_values_all, "Min-Var Portfolio" = "black")
66
67 ggplot() +
68   geom_path(
69     data = frontier_points,
70     aes(x = sd_return, y = expected_return),
71     color = "grey50",
72     linewidth = 1
73   ) +
74   geom_point(
75     data = stats_all_ext,
76     aes(x = sd_return, y = expected_return, color = symbol),
77     size = 3
78   ) +
79   geom_text(
80     data = stats_all_ext,
81     aes(x = sd_return, y = expected_return, label = symbol, color = symbol),
82     vjust = 1.5,
83     size = 3,
84     show.legend = FALSE
85   ) +
86   scale_color_manual(values = color_values_ext, limits = stats_all_ext$symbol) +
87   scale_x_continuous(
88     limits = c(0.02, 0.2),
89     labels = scales::percent_format(accuracy = 0.1)
90   ) +
91   scale_y_continuous(
92     limits = c(-0.01, 0.03),
93     labels = scales::percent_format(accuracy = 0.1)
94   ) +
95   labs(
96     title = "Mean-Variance Map - All 10 Tickers",
97     x = "Volatility (monthly sd)",
98     y = "Expected monthly return"

```

```

99   ) +
100   guides(color = "none") +
101   theme_minimal() +
102   theme(legend.position = "none")

```



```

1  asset_stats <- monthly_returns %>%
2    filter(symbol %in% tickers) %>%
3    group_by(symbol) %>%
4    summarise(
5      expected_return = mean(monthly_return),
6      sd_return       = sd(monthly_return),
7      sharpe_ratio    = expected_return / sd_return,
8      .groups = "drop"
9    )
10
11 weights_50_50 <- c(LRCX = 0.5, GIS = 0.5)
12 min_var_label <- "Min-Var 10-Asset"
13 weights_mv_all <- setNames(round(weights_mv, 4), tickers) # del bloque previo
14
15 portfolio_specs <- list(
16   "50/50 LRCX-GIS" = weights_50_50,
17   portfolio_label  = custom_weights
18 )
19 portfolio_specs[[min_var_label]] <- weights_mv_all
20
21
22 portfolio_stats_ext <- imap_dfr(portfolio_specs, function(w, label) {

```



```

23   returns <- monthly_returns %>%
24     filter(symbol %in% names(w)) %>%
25     mutate(weight = w[symbol]) %>%
26     group_by(date) %>%
27     summarise(portfolio_return = sum(monthly_return * weight), .groups = "drop")
28
29   tibble(
30     symbol      = label,
31     expected_return = mean(returns$portfolio_return),
32     sd_return     = sd(returns$portfolio_return),
33     sharpe_ratio  = expected_return / sd_return
34   )
35 }
36
37 table_levels <- c(tickers, names(portfolio_specs))
38
39 final_table <- bind_rows(asset_stats, portfolio_stats_ext) %>%
40   mutate(symbol = factor(symbol, levels = table_levels)) %>%
41   arrange(symbol)
42
43 print(final_table)

```

```
## # A tibble: 13 x 4
```

	symbol	expected_return	sd_return	sharpe_ratio
	<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	portfolio_label	0.00526	0.0401	0.131
## 13	Min-Var 10-Asset	0.00717	0.0337	0.213

```

1 portfolio_defs <- list(
2   "50/50 LRCX-GIS" = weights_50_50,
3   portfolio_label  = custom_weights,
4   min_var_label    = weights_mv_all
5 )
6
7 portfolio_cum_returns <- imap_dfr(portfolio_defs, function(w, label) {
8   returns <- monthly_returns %>%

```

```

9     filter(symbol %in% names(w)) %>%
10     mutate(weight = w[symbol]) %>%
11     group_by(date) %>%
12     summarise(portfolio_return = sum(monthly_return * weight), .groups = "drop") %>%
13     arrange(date) %>%
14     mutate(index_level = cumprod(1 + portfolio_return)) %>%
15     mutate(symbol = label) %>%
16     select(date, symbol, index_level)
17
18   bind_rows(
19     tibble(date = price_start, symbol = label, index_level = 1),
20     returns
21   )
22 })
23
24 last_values_port <- portfolio_cum_returns %>%
25   arrange(date) %>%
26   group_by(symbol) %>%
27   summarise(final_value = last(index_level), .groups = "drop") %>%
28   arrange(desc(final_value))
29
30 portfolio_colors <- c(
31   "50/50 LRCX-GIS" = "#1f77b4",
32   portfolio_label   = "#d62728",
33   min_var_label     = "#2ca02c"
34 )
35
36 plot_portfolios <- portfolio_cum_returns %>%
37   mutate(symbol = factor(symbol, levels = last_values_port$symbol))
38
39 label_data_port <- plot_portfolios %>%
40   group_by(symbol) %>%
41   filter(date == max(date)) %>%
42   ungroup() %>%
43   mutate(label = paste0("$", formatC(index_level, format = "f", digits = 2)))
44
45 ggplot(plot_portfolios, aes(x = date, y = index_level, color = symbol)) +
46   geom_line(linewidth = 1) +
47   geom_text(
48     data = label_data_port,
49     aes(label = label),
50     hjust = -0.05,
51     vjust = 0.5,
52     size = 3.2,
53     show.legend = FALSE
54   ) +
55   scale_color_manual(values = portfolio_colors, limits = last_values_port$symbol) +
56   scale_x_date(expand = expansion(mult = c(0, 0.12))) +
57   scale_y_continuous(labels = scales::dollar_format(prefix = "$")) +
58   labs(
59     title = "Cumulative Growth of $1 - Three Portfolios",
60     y = "Growth of $1",
61     x = NULL,
62     color = "Portfolio"
63   ) +
64   theme_minimal() +
65   theme(
66     legend.position = c(0, 1),
67     legend.justification = c(0, 1),

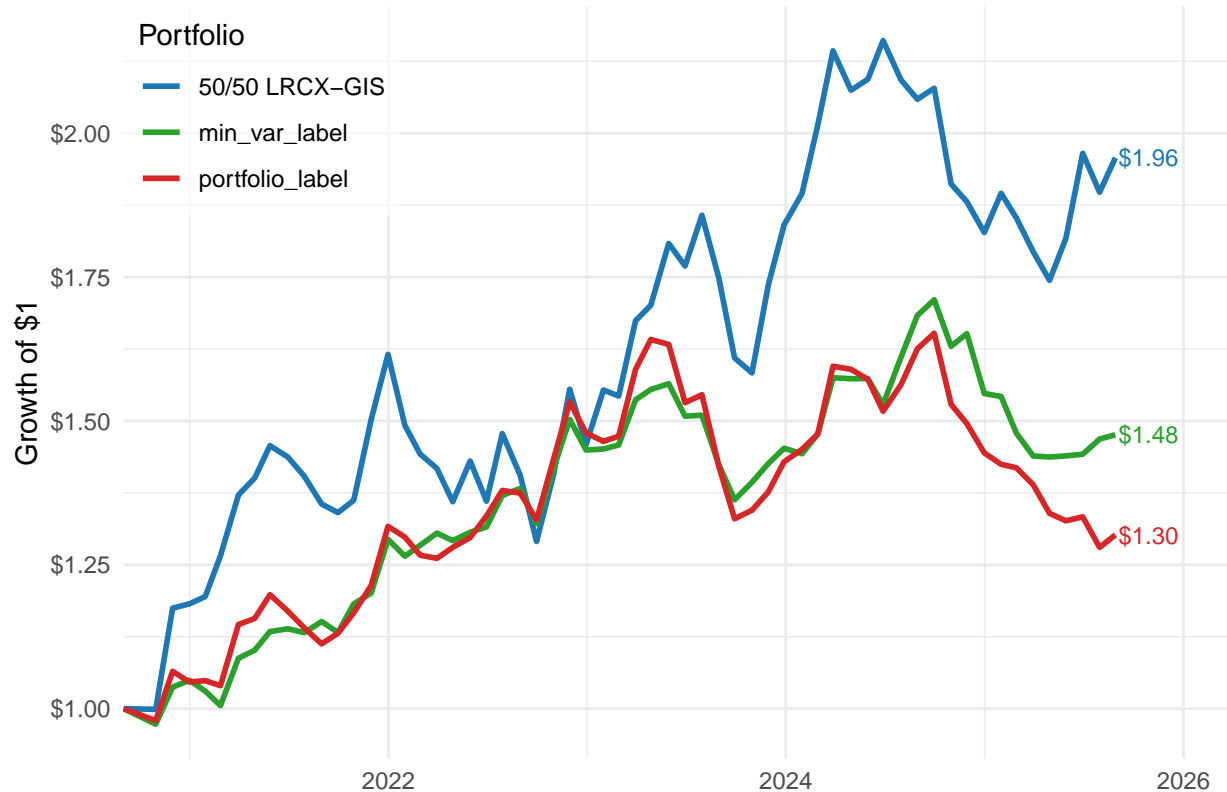
```

```

68   legend.background = element_rect(fill = scales::alpha("white", 0.6), color = NA)
69 )

```

Cumulative Growth of \$1 — Three Portfolios



```

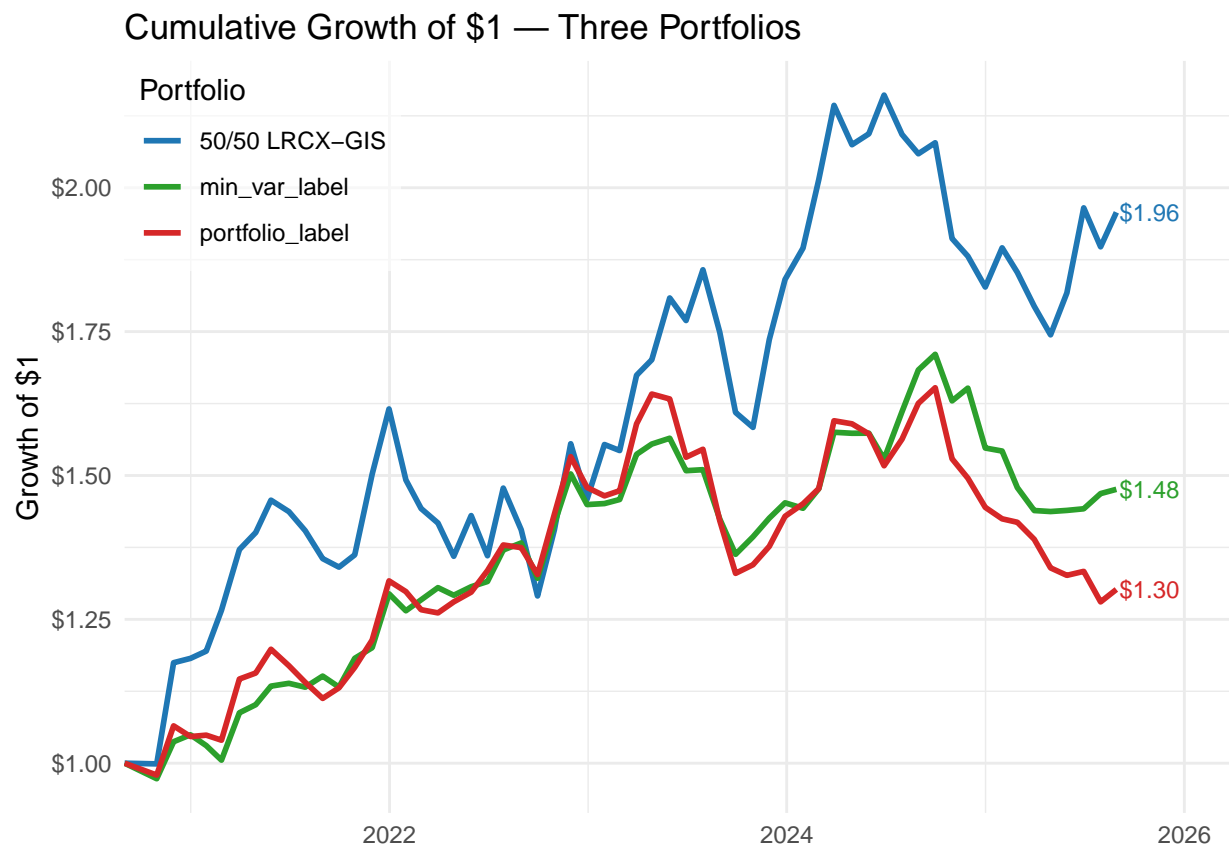
1  weights_50_50 <- c(LRCX = 0.5, GIS = 0.5) # ya definido antes
2  custom_weights <- c(LRCX = 0.1990055, GIS = 0.8009945) # o el vector que uses
3  portfolio_label <- "19.90% LRCX / 80.10% GIS" # etiqueta actual
4
5  portfolio_points <- tribble(
6    ~symbol, ~weights,
7    "50/50 LRCX-GIS", weights_50_50,
8    portfolio_label, custom_weights
9  ) %>%
10 mutate(
11   expected_return = map_dbl(weights, ~ sum(.x * mu_vec[names(.x)])),
12   sd_return = map_dbl(weights, ~ sqrt(as.numeric(t(.x) %*% cov_mat_all[names(.x),
13     ~ names(.x)] %*% .x)))
14 ) %>%
15 select(-weights)
16 stats_all_ext <- bind_rows(stats_all, portfolio_points, min_var_point)
17
18 color_values_ext <- c(
19   color_values_all,
20   "50/50 LRCX-GIS" = "#1f77b4",
21   portfolio_label = "#d62728",
22   "Min-Var Portfolio" = "black"
23 )

```

```

24
25
26 ggplot(plot_portfolios, aes(x = date, y = index_level, color = symbol)) +
27   geom_line(linewidth = 1) +
28   geom_text(
29     data = label_data_port,
30     aes(label = label),
31     hjust = -0.05,
32     vjust = 0.5,
33     size = 3.2,
34     show.legend = FALSE
35   ) +
36   scale_color_manual(values = portfolio_colors, limits = last_values_port$symbol) +
37   scale_x_date(expand = expansion(mult = c(0, 0.12))) +
38   scale_y_continuous(labels = scales::dollar_format(prefix = "$")) +
39   labs(
40     title = "Cumulative Growth of $1 - Three Portfolios",
41     y = "Growth of $1",
42     x = NULL,
43     color = "Portfolio"
44   ) +
45   theme_minimal() +
46   theme(
47     legend.position = c(0, 1),
48     legend.justification = c(0, 1),
49     legend.background = element_rect(fill = scales::alpha("white", 0.6), color = NA)
50   )

```



```

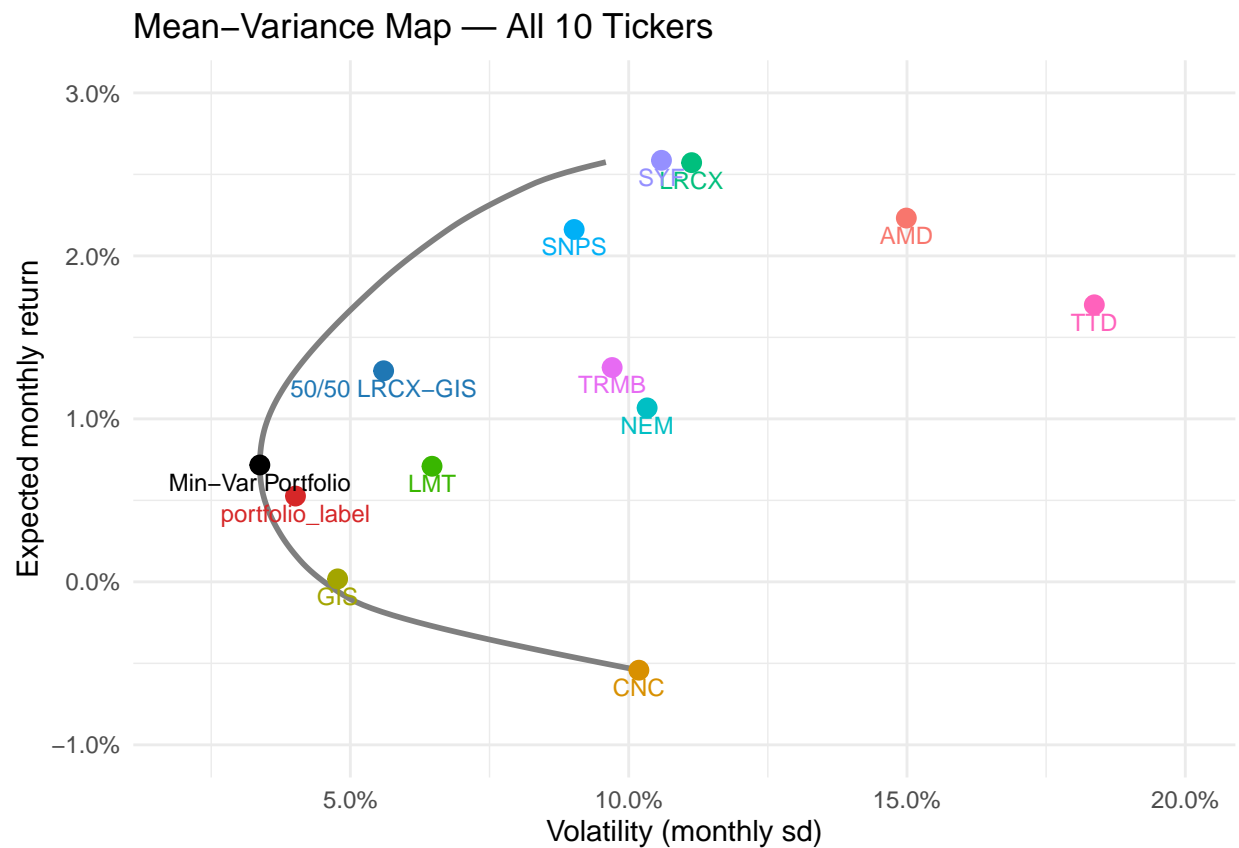
1 library(quadprog)
2
3 weights_50_50 <- c(LRCX = 0.5, GIS = 0.5)
4 portfolio_label <- "19.90% LRCX / 80.10% GIS" # usa el texto que ya venías ocupando
5 custom_weights <- c(LRCX = 0.1990055, GIS = 0.8009945)
6
7 portfolio_points <- list(
8   "50/50 LRCX-GIS" = weights_50_50,
9   portfolio_label = custom_weights
10 ) %>%
11   imap_dfr(function(w, label) {
12     assets <- names(w)
13     mu <- mu_vec[assets]
14     covar <- cov_mat_all[assets, assets]
15
16     tibble(
17       symbol = label,
18       expected_return = sum(w * mu),
19       sd_return = sqrt(as.numeric(t(w) %*% covar %*% w))
20     )
21   })
22
23 stats_all_ext <- bind_rows(stats_all, portfolio_points, min_var_point)
24
25 color_values_ext <- c(
26   color_values_all,
27   "50/50 LRCX-GIS" = "#1f77b4",
28   portfolio_label = "#d62728",
29   "Min-Var Portfolio" = "black"
30 )
31
32 ggplot() +
33   geom_path(
34     data = frontier_points,
35     aes(x = sd_return, y = expected_return),
36     color = "grey50",
37     linewidth = 1
38   ) +
39   geom_point(
40     data = stats_all_ext,
41     aes(x = sd_return, y = expected_return, color = symbol),
42     size = 3
43   ) +
44   geom_text(
45     data = stats_all_ext,
46     aes(x = sd_return, y = expected_return, label = symbol, color = symbol),
47     vjust = 1.5,
48     size = 3,
49     show.legend = FALSE
50   ) +
51   scale_color_manual(values = color_values_ext, limits = stats_all_ext$symbol) +
52   scale_x_continuous(
53     limits = c(0.02, 0.2),
54     labels = scales::percent_format(accuracy = 0.1)
55   ) +
56   scale_y_continuous(
57     limits = c(-0.01, 0.03),
58     labels = scales::percent_format(accuracy = 0.1)
59   ) +

```

```

60   labs(
61     title = "Mean-Variance Map - All 10 Tickers",
62     x = "Volatility (monthly sd)",
63     y = "Expected monthly return"
64   ) +
65   guides(color = "none") +
66   theme_minimal() +
67   theme(legend.position = "none")

```



3 Conclusión.

- Los rendimientos logarítmicos y porcentuales generan resultados prácticamente equivalentes.
- El enfoque no condicional, asume una volatilidad constante en el tiempo y, por tanto, no captura la naturaleza dinámica y heterocedástica observada en los mercados financieros.