

Untitled

```
knitr::opts_chunk$set(
  fig.format = "png",
  fig.dpi = 300,
  fig.width = 7,
  fig.height = 5
)
library(tidyverse)
library(lubridate)

FACE <- 100
bonds <- tibble::tribble(
  ~id, ~name, ~coupon, ~maturity,
  1, "CAN 0.25 Mar 26", 0.00250, "2026-03-01",
  2, "CAN 1.00 Sep 26", 0.01000, "2026-09-01",
  3, "CAN 1.25 Mar 27", 0.01250, "2027-03-01",
  4, "CAN 2.75 Sep 27", 0.02750, "2027-09-01",
  5, "CAN 3.50 Mar 28", 0.03500, "2028-03-01",
  6, "CAN 4.00 Mar 29", 0.04000, "2029-03-01",
  7, "CAN 3.50 Sep 29", 0.03500, "2029-09-01",
  8, "CAN 2.75 Mar 30", 0.02750, "2030-03-01",
  9, "CAN 2.75 Sep 30", 0.02750, "2030-09-01",
  10, "CAN 2.75 Mar 31", 0.02750, "2031-03-01"
) %>%
  mutate(maturity = as.Date(maturity))

days <- as.Date(c("2026-01-05", "2026-01-06", "2026-01-07", "2026-01-08", "2026-01-09",
                 "2026-01-12", "2026-01-13", "2026-01-14", "2026-01-15", "2026-01-16"))

clean_mat <- matrix(c(
  99.70, 99.71, 99.71, 99.72, 99.73, 99.74, 99.74, 99.75, 99.76, 99.77,
  99.145, 99.145, 99.165, 99.160, 99.190, 99.180, 99.190, 99.200, 99.210, 99.220,
```

```

98.600,98.630,98.660,98.670,98.670,98.670,98.680,98.670,98.730,98.720,
100.22,100.30,100.28,100.31,100.30,100.32,100.30,100.31,100.35,100.37,
101.73,101.78,101.78,101.80,101.79,101.81,101.78,101.81,101.84,101.83,
103.63,103.70,103.71,103.74,103.73,103.76,103.72,103.76,103.79,103.78,
102.22,102.33,102.37,102.34,102.31,102.35,102.29,102.33,102.43,102.42,
99.493,99.423,99.563,99.498,99.580,99.528,99.503,99.658,99.663,99.613,
99.165,99.085,99.245,99.170,99.255,99.210,99.185,99.355,99.365,99.315,
98.813,98.723,98.888,98.823,98.913,98.850,98.838,99.033,99.043,98.968
), nrow = 10, byrow = TRUE)

stopifnot(nrow(clean_mat) == 10, ncol(clean_mat) == 10)

colnames(clean_mat) <- format(days, "%Y-%m-%d")
rownames(clean_mat) <- as.character(1:10)

prices <- as_tibble(clean_mat, rownames = "id") %>%
  mutate(id = as.integer(id)) %>%
  pivot_longer(-id, names_to = "settle_chr", values_to = "clean") %>%
  mutate(settle = as.Date(settle_chr)) %>%
  select(-settle_chr) %>%
  left_join(bonds, by = "id")

last_coupon_date <- function(settle, maturity) {
  d <- maturity
  while (d > settle) d <- d %m-% months(6)
  d
}

dirty_price <- function(clean, settle, maturity, coupon_rate, face = 100) {
  lc <- last_coupon_date(settle, maturity)
  n_days <- as.numeric(settle - lc)
  AI <- (n_days / 365) * (face * coupon_rate)
  clean + AI
}

cashflows <- function(settle, maturity, coupon_rate, face = 100) {
  d <- maturity
  pay_dates <- c()
  while (d > settle) {
    pay_dates <- c(d, pay_dates)
  }
}

```

```

    d <- d %m-% months(6)
}
cpn_amt <- face * coupon_rate / 2
cf <- rep(cpn_amt, length(pay_dates))
cf[length(cf)] <- cf[length(cf)] + face
t <- as.numeric(pay_dates - settle) / 365
list(t = t, cf = cf)
}

prices <- prices %>%
  rowwise() %>%
  mutate(
    dirty = dirty_price(clean, settle, maturity, coupon, FACE),
    tau_maturity = as.numeric(maturity - settle) / 365
  ) %>%
  ungroup()

# Q4(a)
#| fig-format: png
#| fig-dpi: 300
#| fig-width: 7
#| fig-height: 5
ytm_continuous <- function(dirty, settle, maturity, coupon_rate, face = 100) {
  cfobj <- cashflows(settle, maturity, coupon_rate, face)
  t <- cfobj$t
  cf <- cfobj$cf
  f <- function(y) sum(cf * exp(-y * t)) - dirty
  uniroot(f, lower = -0.05, upper = 0.25)$root
}

ytm_tbl <- prices %>%
  rowwise() %>%
  mutate(
    tau = tau_maturity,
    y_cont = ytm_continuous(dirty, settle, maturity, coupon, FACE)
  ) %>%
  ungroup() %>%
  filter(tau >= 0, tau <= 5)

tau_grid <- seq(0, 5, by = 0.01)

```

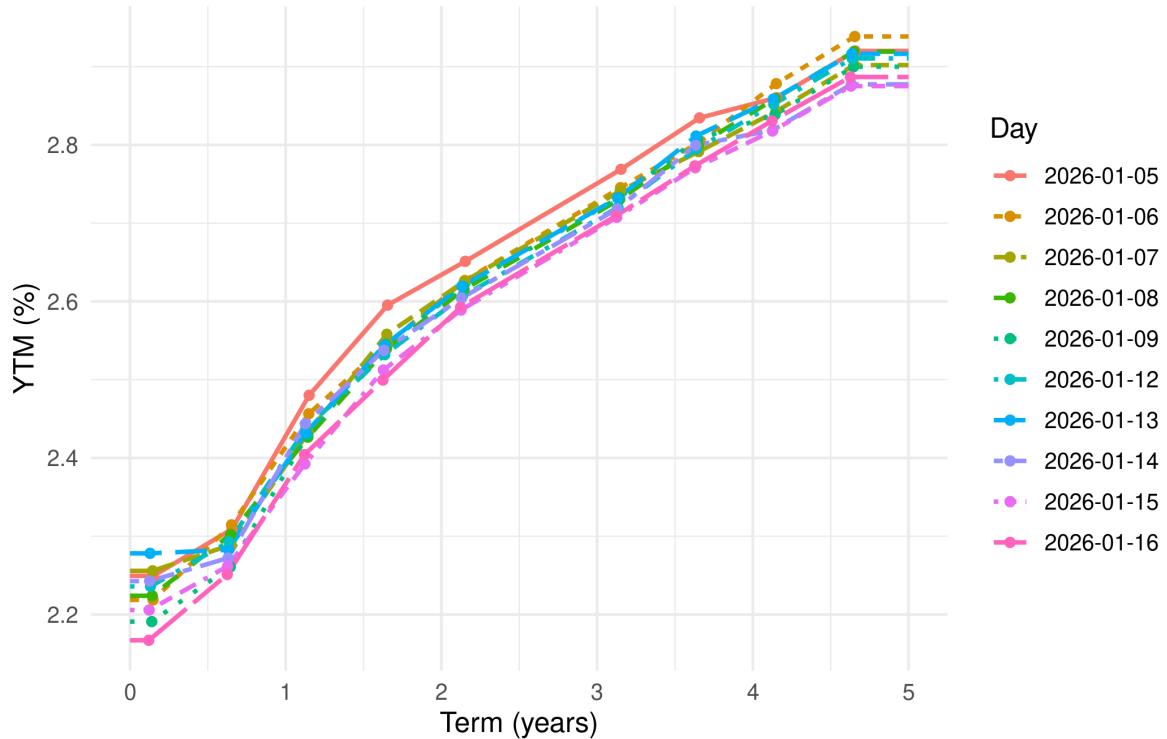
```

ytm_curve_grid <- ytm_tbl %>%
  group_by(settle) %>%
  arrange(tau, .by_group = TRUE) %>%
  reframe({
    x <- tau
    y <- y_cont
    tibble(
      tau = tau_grid,
      y_grid = approx(x = x, y = y, xout = tau_grid,
                      method = "linear", rule = 2)$y
    )
  })
}

ggplot() +
  geom_line(
    data = ytm_curve_grid,
    aes(x = tau, y = 100 * y_grid,
        group = factor(settle),
        color = factor(settle),
        linetype = factor(settle)),
    linewidth = 0.9
  ) +
  geom_point(
    data = ytm_tbl,
    aes(x = tau, y = 100 * y_cont, color = factor(settle)),
    size = 1.6
  ) +
  scale_x_continuous(breaks = 0:5) +
  labs(
    title = "Yield Curves (Continuous-Compounded YTM), 0-5 Years",
    subtitle = "10 days superimposed; linear interpolation between maturity points",
    x = "Term (years)",
    y = "YTM (%)",
    color = "Day",
    linetype = "Day"
  ) +
  theme_minimal(base_size = 12)

```

Yield Curves (Continuous-Compounded YTM), 0–5 Years
10 days superimposed; linear interpolation between maturity points



```

library(tidyverse)
library(lubridate)

FACE <- 100

# --- needs these from your setup: prices, cashflows() ---
stopifnot(exists("prices"))
stopifnot(exists("cashflows"))

# Ensure tau_years exists (robust)
prices2 <- prices %>%
  mutate(
    tau_years = as.numeric(maturity - settle) / 365
  )

# --- Bootstrap spot points for ONE day (continuous comp) ---
bootstrap_spot_one_day <- function(day_df) {

```

```

# ensure tau_years exists inside day_df too
day_df <- day_df %>%
  mutate(tau_years = as.numeric(maturity - settle) / 365) %>%
  arrange(tau_years)

known_t <- numeric(0)
known_r <- numeric(0)

r_at <- function(t) {
  if (length(known_t) == 0) stop("No known spot points yet.")
  if (any(abs(t - known_t) < 1e-12)) return(known_r[which.min(abs(t - known_t))])
  if (t < min(known_t)) return(known_r[which.min(known_t)])
  if (t > max(known_t)) return(known_r[which.max(known_t)])
  approx(x = known_t, y = known_r, xout = t, method = "linear")$y
}

out <- vector("list", nrow(day_df))

for (j in seq_len(nrow(day_df))) {

  settle   <- day_df$settle[j]
  maturity <- day_df$maturity[j]
  cpn      <- day_df$coupon[j]
  Pdirty   <- day_df$dirty[j]

  cfobj <- cashflows(settle, maturity, cpn, FACE)
  t <- cfobj$t
  cf <- cfobj$cf
  n <- length(t)

  PV_known <- 0
  if (n > 1) {
    for (i in 1:(n - 1)) {
      ri <- r_at(t[i])
      PV_known <- PV_known + cf[i] * exp(-ri * t[i])
    }
  }

  Dn <- (Pdirty - PV_known) / cf[n]
  if (!is.finite(Dn) || Dn <= 0) {
    stop(paste0("BOOTSTRAP FAILED: settle=", settle,

```

```

    " bond_id=", day_df$id[j],
    " Dn=", signif(Dn, 8)))
}

rn <- -log(Dn) / t[n]

known_t <- c(known_t, t[n])
known_r <- c(known_r, rn)

out[[j]] <- tibble(settle = settle, t_maturity = t[n], r_cont = rn)
}

bind_rows(out) %>% arrange(t_maturity)
}

# --- Bootstrap ALL days (NO group_modify) ---
prices_by_day <- split(prices2, prices2$settle)
spot_points <- purrr::map_dfr(prices_by_day, bootstrap_spot_one_day)

# --- Attach maturity dates by rank within each settle day ---
spot_curve_date <- prices2 %>%
  distinct(settle, id, name, maturity, tau_years) %>%
  arrange(settle, tau_years) %>%
  group_by(settle) %>%
  mutate(rank = row_number()) %>%
  ungroup() %>%
  left_join(
    spot_points %>%
      group_by(settle) %>%
      arrange(t_maturity, .by_group = TRUE) %>%
      mutate(rank = row_number()) %>%
      ungroup(),
    by = c("settle", "rank")
  ) %>%
  mutate(settle_f = factor(settle))

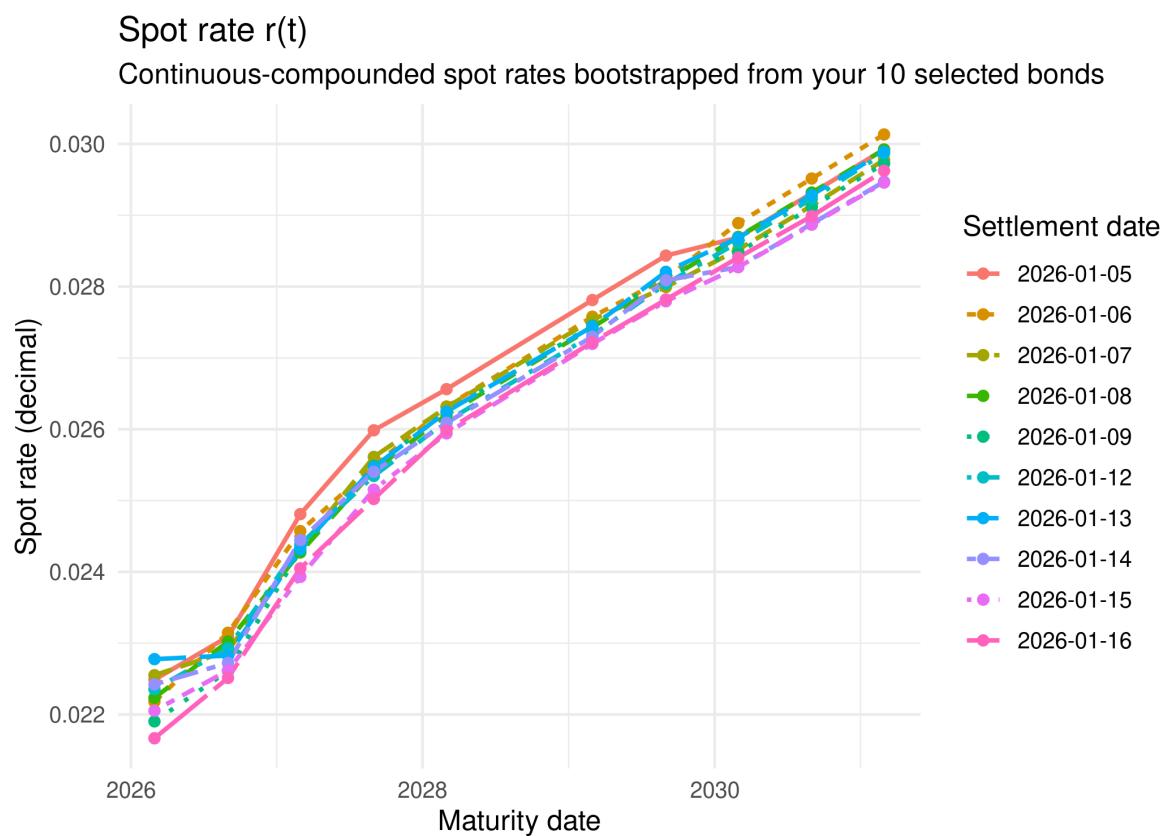
# --- Plot: doc-style axis = maturity date ---
ggplot(
  spot_curve_date,
  aes(
    x = maturity,

```

```

y = r_cont,
group = settle_f,
color = settle_f,
linetype = settle_f
)
) +
geom_line(linewidth = 0.9) +
geom_point(size = 1.8) +
labs(
  title = "Spot rate r(t)",
  subtitle = "Continuous-compounded spot rates bootstrapped from your 10 selected bonds",
  x = "Maturity date",
  y = "Spot rate (decimal)",
  color = "Settlement date",
  linetype = "Settlement date"
) +
theme_minimal(base_size = 12)

```



```

library(tidyverse)
library(lubridate)

stopifnot(exists("spot_points"))

spot_int <- spot_points %>%
  group_by(settle) %>%
  arrange(t_maturity, .by_group = TRUE) %>%
  reframe({
    x <- t_maturity
    y <- r_cont
    tibble(
      T = 1:5,
      rT = approx(x = x, y = y, xout = 1:5, method = "linear", rule = 2)$y
    )
  })
}

# Forward (continuous comp): f_{T-1,T} = r(T)*T - r(T-1)*(T-1), T=2..5
fwd_1yr <- spot_int %>%
  group_by(settle) %>%
  arrange(T, .by_group = TRUE) %>%
  reframe({
    r <- rT
    endT <- 2:5
    f <- r[2:5] * (2:5) - r[1:4] * (1:4)
    tibble(endT = endT, fwd = f)
  }) %>%
  ungroup() %>%
  mutate(
    settle_date = as.Date(settle),
    end_date = settle_date + round(365 * endT), # for plotting only
    settle_f = factor(settle)
  )

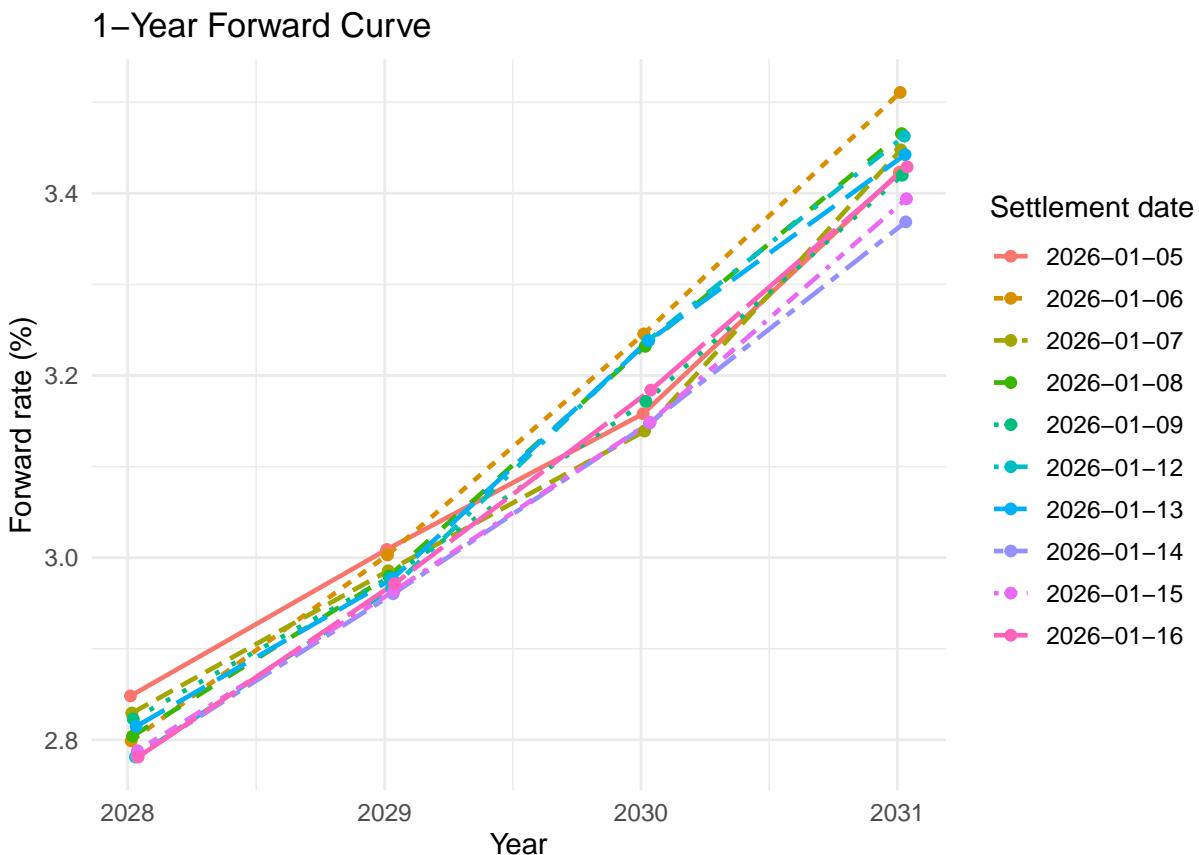
ggplot(
  fwd_1yr,
  aes(
    x = end_date,
    y = 100 * fwd,
    group = settle_f,
    color = settle_f,

```

```

        linetype = settle_f
    )
) +
geom_line(linewidth = 0.9) +
geom_point(size = 1.8) +
labs(
  title = "1-Year Forward Curve",
  x = "Year",
  y = "Forward rate (%)",
  color = "Settlement date",
  linetype = "Settlement date"
) +
theme_minimal(base_size = 12)

```



```

# Q5: Covariance matrices of daily LOG-RETURNS
#   (A) YIELDS: r1..r5 = 1y..5y yields (NO spot rates)

```

```

#   (B) FORWARDS: 1y-1y, 1y-2y, 1y-3y, 1y-4y
#
# Definition: X_{i,j} = log( r_{i,j+1} / r_{i,j} ), j = 1..9
# =====

library(tidyverse)

stopifnot(exists("ytm_tbl"), exists("fwd_1yr"))

# -----
# (A) Yield covariance (X1..X5)
# -----
yield_terms <- 1:5

# get r_i,j = yield at i-year term for each day (interpolate within day)
yield_wide <- ytm_tbl %>%
  group_by(settle) %>%
  arrange(tau, .by_group = TRUE) %>%
  reframe({
    x <- tau
    y <- y_cont
    tibble(
      term = yield_terms,
      r = approx(x = x, y = y, xout = yield_terms, method = "linear", rule = 2)$y
    )
  }) %>%
  ungroup() %>%
  mutate(var = paste0("X", term)) %>%
  select(settle, var, r) %>%
  pivot_wider(names_from = var, values_from = r) %>%
  arrange(settle)

# compute log-returns X_{i,j} = log(r_{i,j+1}/r_{i,j}) for j=1..9
yield_logret <- yield_wide %>%
  mutate(across(starts_with("X"), ~ log(lead(.x) / .x))) %>%
  slice(1:(n() - 1)) %>% # keep j = 1..9
  select(starts_with("X"))

Cov_Yield <- cov(as.matrix(yield_logret), use = "complete.obs")
rownames(Cov_Yield) <- colnames(Cov_Yield) <- paste0("X", 1:5)

```

```

# -----
# (B) Forward covariance (X1..X4 forwards)
# X1 = 1y-1y, X2 = 1y-2y, X3 = 1y-3y, X4 = 1y-4y
# -----
fwd_wide <- fwd_1yr %>%
  mutate(var = case_when(
    endT == 2 ~ "X1", # 1y-1y
    endT == 3 ~ "X2", # 1y-2y
    endT == 4 ~ "X3", # 1y-3y
    endT == 5 ~ "X4" # 1y-4y
  )) %>%
  select(settle, var, r = fwd) %>%
  pivot_wider(names_from = var, values_from = r) %>%
  arrange(settle)

# log-returns for forwards
fwd_logret <- fwd_wide %>%
  mutate(across(starts_with("X"), ~ log(lead(.x) / .x))) %>%
  slice(1:(n() - 1)) %>%
  select(starts_with("X"))

Cov_Forward <- cov(as.matrix(fwd_logret), use = "complete.obs")
rownames(Cov_Forward) <- colnames(Cov_Forward) <- paste0("X", 1:4)

# -----
# Print results
# -----
cat("\n==== Covariance matrix: daily log-returns of YIELDS (X1..X5 = 1y..5y) ===\n")

==== Covariance matrix: daily log-returns of YIELDS (X1..X5 = 1y..5y) ===

print(Cov_Yield)

```

	X1	X2	X3	X4	X5
X1	5.462531e-05	4.231525e-06	1.534532e-06	1.335811e-05	1.079575e-05
X2	4.231525e-06	2.767099e-05	2.000900e-05	-1.485856e-06	-1.245030e-05
X3	1.534532e-06	2.000900e-05	1.576227e-05	3.572018e-06	-3.022616e-06
X4	1.335811e-05	-1.485856e-06	3.572018e-06	3.652394e-05	4.476728e-05
X5	1.079575e-05	-1.245030e-05	-3.022616e-06	4.476728e-05	5.962618e-05

```

cat("\n==== Covariance matrix: daily log-returns of FORWARDS (X1..X4 = 1y-1y..1y-4y) ===\n"

==== Covariance matrix: daily log-returns of FORWARDS (X1..X4 = 1y-1y..1y-4y) ===

print(Cov_Forward)

      X1          X2          X3          X4
X1  1.252076e-04 1.475132e-05 -1.518390e-04 -9.725355e-05
X2  1.475132e-05 1.167729e-05  2.535662e-05  1.692211e-05
X3 -1.518390e-04 2.535662e-05  5.543492e-04  3.361380e-04
X4 -9.725355e-05 1.692211e-05  3.361380e-04  2.487606e-04

stopifnot(exists("Cov_Yield"), exists("Cov_Forward"))

print_eigen_clean <- function(CovMat, row_names, title) {

  eg <- eigen(as.matrix(CovMat))

  cat("=====\n")
  cat(title, "\n")
  cat("=====\\n")

  # Eigenvalues
  eigvals <- data.frame(
    Eigenvalue = paste0("lambda_", seq_along(eg$values)),
    Value = signif(eg$values, 6),
    stringsAsFactors = FALSE
  )
  cat("\\nEigenvalues (largest -> smallest):\n")
  print(eigvals, row.names = FALSE)

  # Eigenvectors: columns are v1, v2, ...
  eigvecs <- as.data.frame(round(eg$vectors, 6))
  colnames(eigvecs) <- paste0("v", 1:ncol(eigvecs))
  eigvecs <- cbind(Component = row_names, eigvecs)

  cat("\\nEigenvectors (each column v_k corresponds to lambda_k above):\n")
  print(eigvecs, row.names = FALSE)
}

```

```

    invisible(eg)
}

print_eigen_clean(
  Cov_Yield,
  row_names = c("1Y yield", "2Y yield", "3Y yield", "4Y yield", "5Y yield"),
  title = "YIELD covariance matrix: eigenvalues and eigenvectors"
)

```

```
=====
YIELD covariance matrix: eigenvalues and eigenvectors
=====
```

Eigenvalues (largest -> smallest):

Eigenvalue	Value
lambda_1	1.01494e-04
lambda_2	5.41164e-05
lambda_3	3.81469e-05
lambda_4	3.28577e-07
lambda_5	1.22484e-07

Eigenvectors (each column v_k corresponds to lambda_k above):

Component	v1	v2	v3	v4	v5
1Y yield	-0.322704	0.794782	0.508666	0.067904	0.028843
2Y yield	0.124665	0.468156	-0.608712	-0.492792	0.389763
3Y yield	0.025335	0.294674	-0.532152	0.757632	-0.235231
4Y yield	-0.577196	-0.010873	-0.261197	-0.370751	-0.679004
5Y yield	-0.739272	-0.249401	-0.138993	0.202691	0.575215

```

print_eigen_clean(
  Cov_Forward,
  row_names = c("1y-1y fwd", "1y-2y fwd", "1y-3y fwd", "1y-4y fwd"),
  title = "FORWARD covariance matrix: eigenvalues and eigenvectors"
)

```

```
=====
FORWARD covariance matrix: eigenvalues and eigenvectors
=====
```

Eigenvalues (largest -> smallest):

Eigenvalue	Value
lambda_1	8.18563e-04
lambda_2	8.47789e-05
lambda_3	3.23184e-05
lambda_4	4.33434e-06

Eigenvectors (each column v_k corresponds to lambda_k above):

Component	v1	v2	v3	v4
1y-1y fwd	-0.250923	0.930175	-0.013187	0.267653
1y-2y fwd	0.031960	0.284447	0.009470	-0.958112
1y-3y fwd	0.813499	0.189300	-0.544342	0.077956
1y-4y fwd	0.523676	0.134275	0.838706	0.065621