Untitled

```r
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.0     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.1
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

```r
library(lubridate)

FACE <- 100

# -------------------------
# Bonds (your 10)
# -------------------------
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",
```

```r
   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 (10)
# -------------------------
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 prices (10x10)
# -------------------------
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)

# -------------------------
# Long table: bond-day-clean
# -------------------------
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")
```

```r
# -------------------------
# Lecture simplifications: ACT/365, semiannual coupons
# -------------------------
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()
```

```r
# ============================================================
# Q4(a): YTM (continuous comp) + superimposed 0-5y curves
# Interpolation technique: linear interpolation between observed maturities
```

```r
# ============================================================

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)
```

```
Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'Yield Curves (Continuous-Compounded YTM), 0-5 Years' in
'mbcsToSbcs': dot substituted for <e2>
```

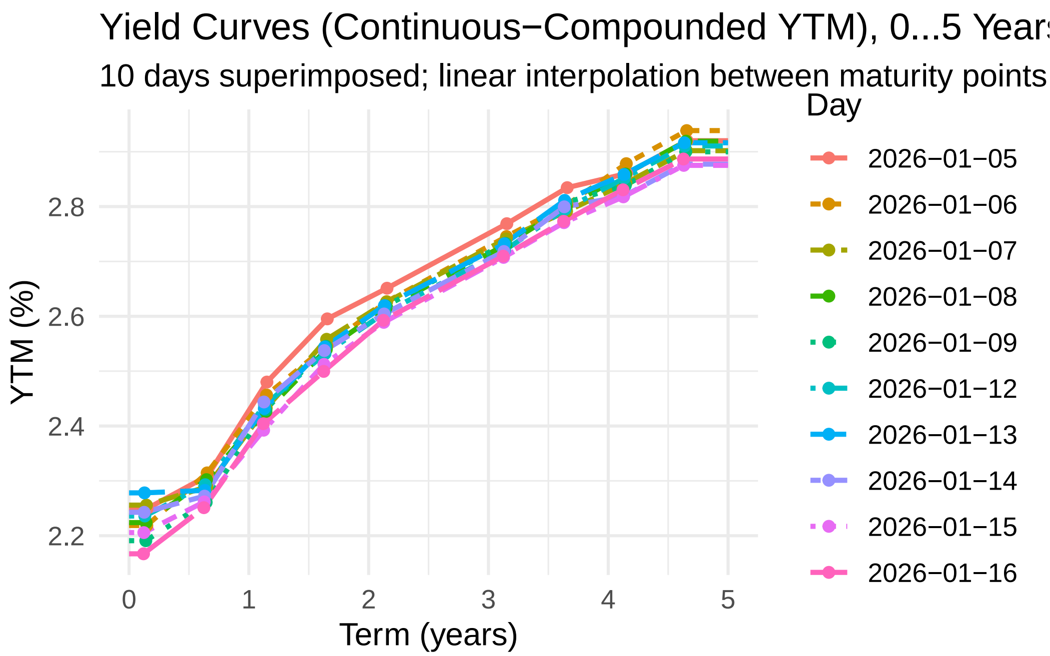
```
Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'Yield Curves (Continuous-Compounded YTM), 0-5 Years' in
'mbcsToSbcs': dot substituted for <80>
```

```
Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'Yield Curves (Continuous-Compounded YTM), 0-5 Years' in
'mbcsToSbcs': dot substituted for <93>
```

```
Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'Yield Curves (Continuous-Compounded YTM), 0-5 Years' in
'mbcsToSbcs': dot substituted for <e2>
```

```
Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'Yield Curves (Continuous-Compounded YTM), 0-5 Years' in
'mbcsToSbcs': dot substituted for <80>
```

```
Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
conversion failure on 'Yield Curves (Continuous-Compounded YTM), 0-5 Years' in
'mbcsToSbcs': dot substituted for <93>
```

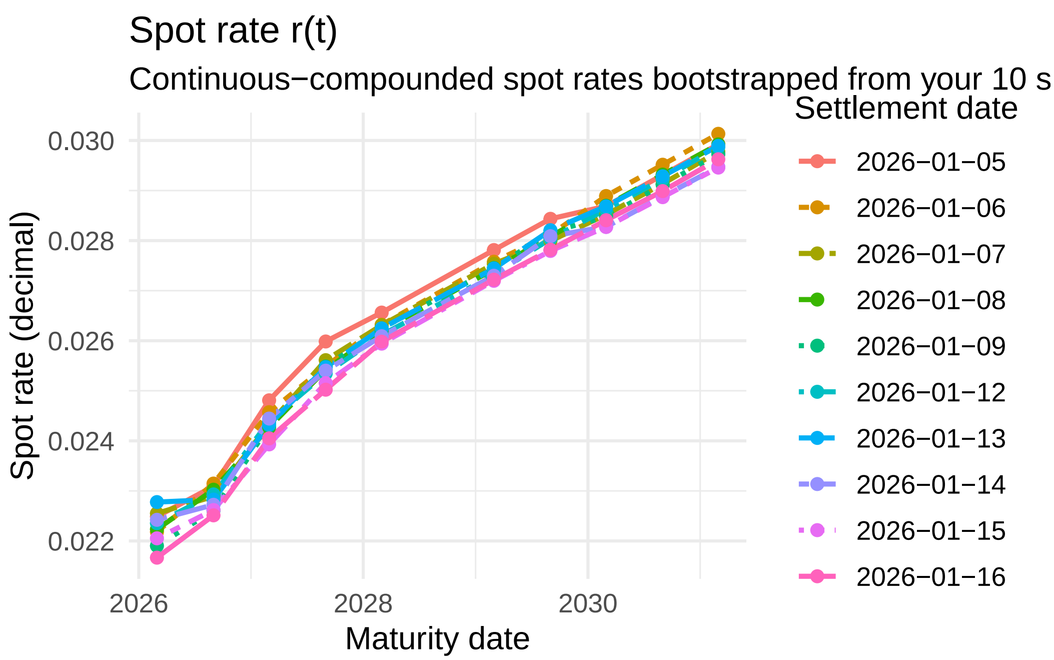



```
# ============================================================
# ============================================================
# CHUNK B (COMPLETE + FIXED) - SPOT CURVE r(t) with DOC-STYLE X-AXIS (MATURITY DATE)
# 10 daily spot curves superimposed
#
# Does NOT assume tau_years exists; computes it as (maturity - settle)/365
# ============================================================

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
  )
```

```r
# --- 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])
      }
    }
```

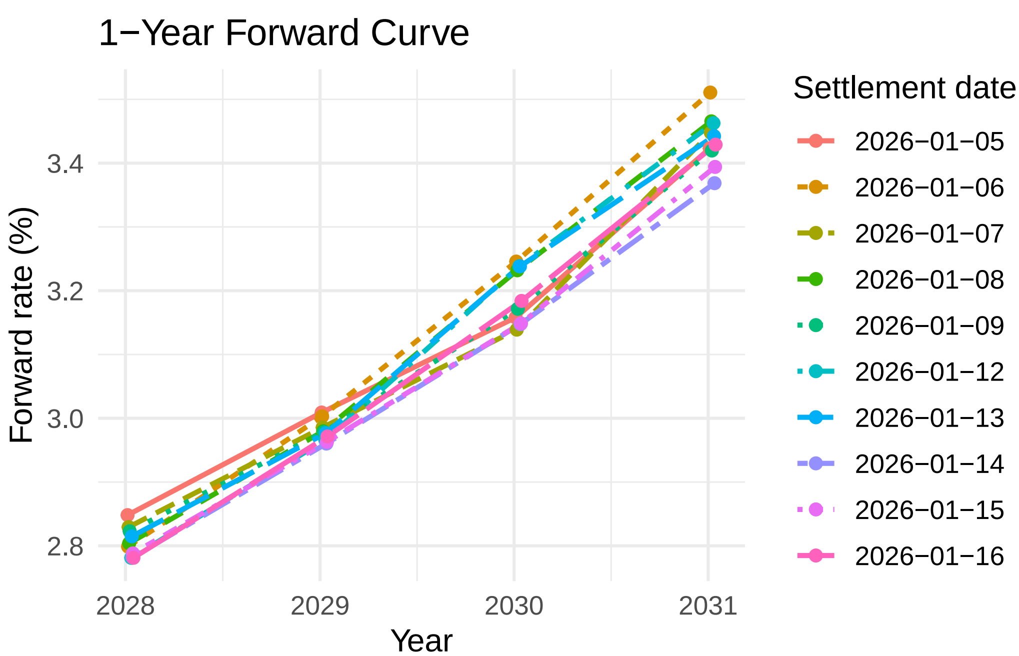
```r
    Dn <- (Pdirty - PV_known) / cf[n]
    if (!is.finite(Dn) || Dn <= 0) {
      stop(paste0("BOOTSTRAP FAILED: settle=", settle,
                  " bond_id=", day_df$id[j],
                  " Dn=", signif(Dn, 8)))
    }

    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)
```



```r
# ============================================================
# CHUNK C (COMPLETE + FIXED) - 1-YEAR FORWARD CURVE with DOC-STYLE X-AXIS (DATE)
# Uses spot_points created in CHUNK B.
# ============================================================

library(tidyverse)
library(lubridate)

stopifnot(exists("spot_points"))

# Interpolate spot r(1)..r(5) per day
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(
```

```r
  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)
```




```r
print(fwd_1yr %>% arrange(settle, endT), n = Inf)
```

```
# A tibble: 40 x 6
   settle      endT    fwd settle_date end_date   settle_f
   <date>     <int>  <dbl> <date>      <date>     <fct>
 1 2026-01-05     2 0.0285 2026-01-05  2028-01-05 2026-01-05
 2 2026-01-05     3 0.0301 2026-01-05  2029-01-04 2026-01-05
 3 2026-01-05     4 0.0316 2026-01-05  2030-01-04 2026-01-05
 4 2026-01-05     5 0.0342 2026-01-05  2031-01-04 2026-01-05
 5 2026-01-06     2 0.0280 2026-01-06  2028-01-06 2026-01-06
 6 2026-01-06     3 0.0300 2026-01-06  2029-01-05 2026-01-06
 7 2026-01-06     4 0.0325 2026-01-06  2030-01-05 2026-01-06
 8 2026-01-06     5 0.0351 2026-01-06  2031-01-05 2026-01-06
 9 2026-01-07     2 0.0283 2026-01-07  2028-01-07 2026-01-07
10 2026-01-07     3 0.0299 2026-01-07  2029-01-06 2026-01-07
11 2026-01-07     4 0.0314 2026-01-07  2030-01-06 2026-01-07
12 2026-01-07     5 0.0345 2026-01-07  2031-01-06 2026-01-07
13 2026-01-08     2 0.0280 2026-01-08  2028-01-08 2026-01-08
14 2026-01-08     3 0.0298 2026-01-08  2029-01-07 2026-01-08
15 2026-01-08     4 0.0323 2026-01-08  2030-01-07 2026-01-08
16 2026-01-08     5 0.0347 2026-01-08  2031-01-07 2026-01-08
17 2026-01-09     2 0.0282 2026-01-09  2028-01-09 2026-01-09
18 2026-01-09     3 0.0298 2026-01-09  2029-01-08 2026-01-09
19 2026-01-09     4 0.0317 2026-01-09  2030-01-08 2026-01-09
20 2026-01-09     5 0.0342 2026-01-09  2031-01-08 2026-01-09
21 2026-01-12     2 0.0278 2026-01-12  2028-01-12 2026-01-12
22 2026-01-12     3 0.0297 2026-01-12  2029-01-11 2026-01-12
23 2026-01-12     4 0.0324 2026-01-12  2030-01-11 2026-01-12
24 2026-01-12     5 0.0346 2026-01-12  2031-01-11 2026-01-12
25 2026-01-13     2 0.0281 2026-01-13  2028-01-13 2026-01-13
26 2026-01-13     3 0.0298 2026-01-13  2029-01-12 2026-01-13
27 2026-01-13     4 0.0324 2026-01-13  2030-01-12 2026-01-13
28 2026-01-13     5 0.0344 2026-01-13  2031-01-12 2026-01-13
29 2026-01-14     2 0.0278 2026-01-14  2028-01-14 2026-01-14
30 2026-01-14     3 0.0296 2026-01-14  2029-01-13 2026-01-14
31 2026-01-14     4 0.0315 2026-01-14  2030-01-13 2026-01-14
32 2026-01-14     5 0.0337 2026-01-14  2031-01-13 2026-01-14
33 2026-01-15     2 0.0279 2026-01-15  2028-01-15 2026-01-15
34 2026-01-15     3 0.0296 2026-01-15  2029-01-14 2026-01-15
35 2026-01-15     4 0.0315 2026-01-15  2030-01-14 2026-01-15
36 2026-01-15     5 0.0339 2026-01-15  2031-01-14 2026-01-15
37 2026-01-16     2 0.0278 2026-01-16  2028-01-16 2026-01-16
38 2026-01-16     3 0.0297 2026-01-16  2029-01-15 2026-01-16
39 2026-01-16     4 0.0318 2026-01-16  2030-01-15 2026-01-16
40 2026-01-16     5 0.0343 2026-01-16  2031-01-15 2026-01-16
```

```r
# ============================================================
# 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"))
```

```r
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) ===
```

```r
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
```