

# APM466 Assignment 1

```
# Load necessary libraries  
library(dplyr)
```

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

filter, lag

The following objects are masked from 'package:base':

intersect, setdiff, setequal, union

```
library(tidyr)  
  
# Load the CSV file  
bond_data <- read.csv("Selected Bonds.csv")  
  
# Map column names to actual dates  
date_mapping <- c(  
  "jan_6" = "2025-01-06",  
  "jan_7" = "2025-01-07",  
  "jan_8" = "2025-01-08",  
  "jan_9" = "2025-01-09",  
  "jan_10" = "2025-01-10",  
  "jan_13" = "2025-01-13",  
  "jan_14" = "2025-01-14",  
  "jan_15" = "2025-01-15",  
  "jan_16" = "2025-01-16",
```

```

"jan_17" = "2025-01-17"
)

# Convert coupon_rate to numeric
bond_data <- bond_data |>
  mutate(coupon_rate = as.numeric(coupon_rate))

# Define face value (FV)
face_value <- 100

# Define a function to calculate dirty price
calculate_dirty_price <- function(clean_price, coupon_rate, column_name, last_coupon_date, face_value) {
  # Get the pricing date from the column name
  current_date <- as.Date(date_mapping[column_name])

  # Calculate days since the last coupon
  days_since_last_coupon <- as.numeric(current_date - as.Date(last_coupon_date))

  # Calculate accrued interest
  accrued_interest <- face_value * (coupon_rate / 2) * (days_since_last_coupon / 182.5)

  # Calculate dirty price
  dirty_price <- clean_price + accrued_interest
  return(dirty_price)
}

# Iterate through jan_* columns and calculate dirty prices
for (col in names(date_mapping)) {
  bond_data[[paste0(col, "_dirty")]] <- mapply(
    calculate_dirty_price,
    clean_price = bond_data[[col]],
    coupon_rate = bond_data$coupon_rate,
    column_name = col,
    last_coupon_date = bond_data$last_coupon_date,
    MoreArgs = list(face_value = face_value)
  )
}

# Remove unnecessary columns (e.g., X, X.1)
bond_data <- bond_data |>
  select(-starts_with("X"))

```

```

# Round all price-related columns to 2 decimal places
bond_data <- bond_data |>
  mutate(across(
    where(is.numeric) & !all_of("coupon_rate"), # Exclude coupon_rate
    ~ round(., 2)
  ))

# Save the final cleaned and rounded dataset to a CSV
write.csv(bond_data, "Final_Bond_Data.csv", row.names = FALSE)

# View the first few rows of the final dataset
head(bond_data)

```

|   | ISIN         | coupon_rate | issue_date | maturity_date | last_coupon_date |
|---|--------------|-------------|------------|---------------|------------------|
| 1 | CA135087K528 | 0.0125      | 2019-10-11 | 2025-03-01    | 2024-09-01       |
| 2 | CA135087K940 | 0.0050      | 2020-04-03 | 2025-09-01    | 2024-09-01       |
| 3 | CA135087L518 | 0.0025      | 2020-10-09 | 2026-03-01    | 2024-09-01       |
| 4 | CA135087L930 | 0.0100      | 2021-04-16 | 2026-09-01    | 2024-09-01       |
| 5 | CA135087M847 | 0.0125      | 2021-10-15 | 2027-03-01    | 2024-09-01       |
| 6 | CA135087N837 | 0.0275      | 2022-05-13 | 2027-09-01    | 2024-09-01       |

  

|   | next_coupon_date | jan_6 | jan_7 | jan_8 | jan_9 | jan_10 | jan_13 | jan_14 | jan_15 | jan_16 |
|---|------------------|-------|-------|-------|-------|--------|--------|--------|--------|--------|
| 1 | 2025-03-01       | 99.73 | 99.73 | 99.74 | 99.73 | 99.74  | 99.73  | 99.73  | 99.77  | 99.78  |
| 2 | 2025-03-01       | 98.40 | 98.41 | 98.40 | 98.42 | 98.37  | 98.36  | 98.36  | 98.40  | 98.47  |
| 3 | 2025-03-01       | 96.99 | 96.98 | 96.97 | 97.00 | 96.90  | 96.91  | 96.80  | 96.90  | 97.01  |
| 4 | 2025-03-01       | 97.01 | 96.99 | 97.00 | 97.03 | 96.86  | 96.77  | 96.71  | 96.85  | 97.03  |
| 5 | 2025-03-01       | 96.62 | 96.59 | 96.58 | 96.63 | 96.39  | 96.28  | 96.19  | 96.36  | 96.60  |
| 6 | 2025-03-01       | 99.63 | 99.56 | 99.54 | 99.58 | 99.27  | 99.13  | 99.02  | 99.20  | 99.50  |

  

|   | jan_17 | jan_6_dirty | jan_7_dirty | jan_8_dirty | jan_9_dirty | jan_10_dirty |
|---|--------|-------------|-------------|-------------|-------------|--------------|
| 1 | 99.80  | 100.16      | 100.17      | 100.18      | 100.18      | 100.19       |
| 2 | 98.50  | 98.57       | 98.59       | 98.58       | 98.60       | 98.55        |
| 3 | 97.06  | 97.08       | 97.07       | 97.06       | 97.09       | 96.99        |
| 4 | 97.07  | 97.36       | 97.34       | 97.35       | 97.39       | 97.22        |
| 5 | 96.64  | 97.05       | 97.03       | 97.02       | 97.08       | 96.84        |
| 6 | 99.53  | 100.59      | 100.52      | 100.51      | 100.56      | 100.26       |

  

|   | jan_13_dirty | jan_14_dirty | jan_15_dirty | jan_16_dirty | jan_17_dirty |
|---|--------------|--------------|--------------|--------------|--------------|
| 1 | 100.19       | 100.19       | 100.24       | 100.25       | 100.27       |
| 2 | 98.54        | 98.54        | 98.59        | 98.66        | 98.69        |
| 3 | 97.00        | 96.89        | 96.99        | 97.10        | 97.15        |
| 4 | 97.14        | 97.08        | 97.22        | 97.41        | 97.45        |
| 5 | 96.74        | 96.65        | 96.83        | 97.07        | 97.11        |
| 6 | 100.14       | 100.04       | 100.22       | 100.53       | 100.57       |