

HW_Lubridate_Purrr

Shoubith Rajeswaran

2025-05-07

Question 1

```
# Generate sequence of dates every 2 months from Jan 1, 2015 to Dec 31, 2025
date_seq <- seq(ymd("2015-01-01"), ymd("2025-12-31"), by = "2 months")
```

```
# Extract year, quarter, and ISO week number
date_info <- tibble(
  date = date_seq,
  year = year(date_seq),
  quarter = quarter(date_seq),
  iso_week = isoweek(date_seq)
)
```

```
# Display the result
print(date_info)
```

```
## # A tibble: 66 x 4
##   date      year quarter iso_week
##   <date>    <dbl>   <int>   <dbl>
## 1 2015-01-01  2015     1       1
## 2 2015-03-01  2015     1       9
## 3 2015-05-01  2015     2      18
## 4 2015-07-01  2015     3      27
## 5 2015-09-01  2015     3      36
## 6 2015-11-01  2015     4      44
## 7 2016-01-01  2016     1      53
## 8 2016-03-01  2016     1       9
## 9 2016-05-01  2016     2      17
## 10 2016-07-01 2016     3      26
## # i 56 more rows
```

Question 2

```
# Vector of sample dates
sample_dates <- ymd(c("2018-03-15", "2020-07-20", "2023-01-10", "2025-09-05"))

# Create list of consecutive date pairs
date_pairs <- map2(sample_dates[-length(sample_dates)], sample_dates[-1], ~ interval(.x, .y))

# Calculate difference in months
diff_months <- map_dbl(date_pairs, ~ time_length(.x, "months"))
```

```

# Calculate difference in weeks
diff_weeks <- map_dbl(date_pairs, ~ time_length(., "weeks"))

# Combine results in a tibble
differences <- tibble(
  start_date = sample_dates[-length(sample_dates)],
  end_date = sample_dates[-1],
  months_between = round(diff_months, 2),
  weeks_between = round(diff_weeks, 2)
)

# Display the result
print(differences)

## # A tibble: 3 x 4
##   start_date end_date   months_between weeks_between
##   <date>      <date>         <dbl>         <dbl>
## 1 2018-03-15 2020-07-20         28.2          123.
## 2 2020-07-20 2023-01-10         29.7          129.
## 3 2023-01-10 2025-09-05         31.8          138.

```

Question 3

```

# List of numeric vectors
num_lists <- list(c(4, 16, 25, 36, 49), c(2.3, 5.7, 8.1, 11.4), c(10, 20, 30, 40, 50))

# Compute mean, median, and standard deviation for each sublist
means <- map_dbl(num_lists, mean)
medians <- map_dbl(num_lists, median)
sds <- map_dbl(num_lists, sd)

# Combine results in a tibble
summary_stats <- tibble(
  list_number = paste0("List ", seq_along(num_lists)),
  mean = means,
  median = medians,
  standard_deviation = round(sds, 2)
)

# Display the summary
print(summary_stats)

## # A tibble: 3 x 4
##   list_number mean median standard_deviation
##   <chr>      <dbl> <dbl>         <dbl>
## 1 List 1      26      25          17.4
## 2 List 2      6.88    6.9           3.84
## 3 List 3      30      30          15.8

```

Question 4

```
# List of mixed-format date strings
date_strings <- list("2023-06-10", "2022/12/25", "15-Aug-2021", "InvalidDate")

# Safe date parser using parse_date_time with broader format options
safe_parse <- possibly(
  ~ parse_date_time(.x, orders = c("ymd", "Ymd", "d-b-Y", "Y/m/d"), locale = "C"),
  otherwise = NA
)

# Apply the safe parser
parsed_dates <- map(date_strings, safe_parse)
```

```
## Warning: All formats failed to parse. No formats found.
```

```
# Extract month names (explicit string conversion)
month_names <- map_chr(parsed_dates, ~ if (!is.na(.x)) as.character(month(.x, label = TRUE)) else "Invalid")
```

```
# Combine into a tibble
date_results <- tibble(
  original_input = unlist(date_strings),
  parsed_date = parsed_dates,
  month_name = month_names
)
```

```
# Display the result
print(date_results)
```

```
## # A tibble: 4 x 3
##   original_input parsed_date month_name
##   <chr>          <list>      <chr>
## 1 2023-06-10    <dtm [1]>   Jun
## 2 2022/12/25   <dtm [1]>   Dec
## 3 15-Aug-2021  <dtm [1]>   Aug
## 4 InvalidDate  <dtm [1]>   Invalid
```