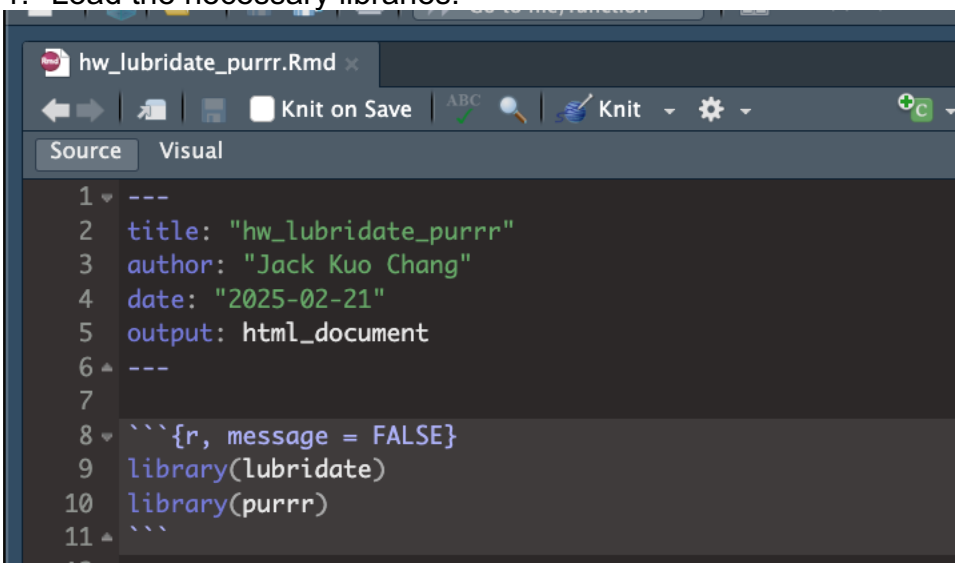
Homework: lubridate and purrr

Instructions
Complete the following exercises using the lubridate and purrr packages in R. Ensure that
your solutions are optimized and use functional programming principles where applicable.

1. Load the necessary libraries.



2. Answer each question in separate R code chunks.
3. Provide detailed explanations for your approach.
4. Submit the rendered HTML file.

Link: https://github.com/jkuoc183/hw_purrr

SSH: git@github.com:jkuoc183/hw_purrr.git

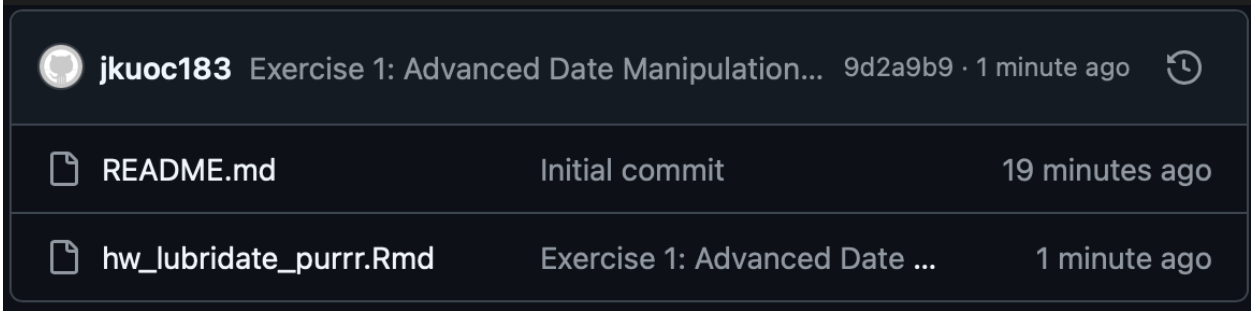
Exercise 1: Advanced Date Manipulation with lubridate (Commit)
Question 1:
Generate a sequence of dates from January 1, 2015 to December 31, 2025, spaced by every two months. Extract the year, quarter, and ISO week number for each date.

```r
#1Generate a sequence of dates from January 1, 2015 to December 31, 2025, spaced by
#every two months. Extract the year, quarter, and ISO week #number for each date.

dates_seq <- seq(ymd("2015-01-01"), ymd("2025-12-31"), by = "2 months")
year_values <- year(dates_seq)
quarter_values <- quarter(dates_seq)
iso_week_values <- isoweek(dates_seq)

result_ex1 <- data.frame(Date = dates_seq, Year = year_values, Quarter = quarter_values, ISO_Week =
iso_week_values)
print(result_ex1)
```

```
j.k.@Jacks-MacBook-Pro-2 hw_purrr % git add hw_lubridate_purrr.Rmd
j.k.@Jacks-MacBook-Pro-2 hw_purrr % git commit -m "Exercise 1: Advanced Date Manipulation with lubridate"
[main 9d2a9b9] Exercise 1: Advanced Date Manipulation with lubridate
 1 file changed, 25 insertions(+)
 create mode 100644 hw_lubridate_purrr.Rmd
j.k.@Jacks-MacBook-Pro-2 hw_purrr % git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 706 bytes | 706.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:jkuoc183/hw_purrr.git
   37e4880..9d2a9b9  main -> main
j.k.@Jacks-MacBook-Pro-2 hw_purrr %
```



Exercise 2: Complex Date Arithmetic
Question 2:
Given the following dates, compute the difference in months and weeks between each consecutive pair.
sample_dates <- c("2018-03-15", "2020-07-20", "2023-01-10", "2025-09-05")

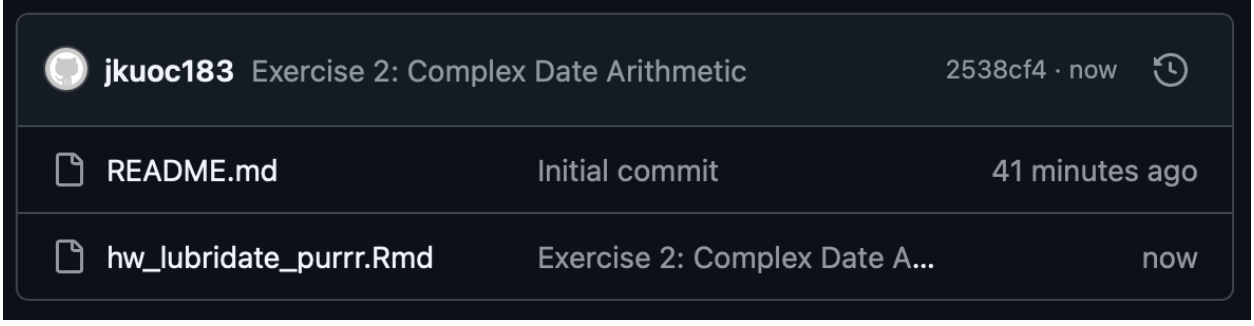
```r
#2
sample_dates <- ymd(c("2018-03-15", "2020-07-20", "2023-01-10", "2025-09-05"))

month_diffs <- diff(sample_dates) / dmonths(1)
week_diffs <- diff(sample_dates) / dweeks(1)

result_ex2 <- data.frame(
  Start_Date = sample_dates[-length(sample_dates)],
  End_Date = sample_dates[-1],
  Month_Difference = month_diffs,
  Week_Difference = week_diffs
)
print(result_ex2)
```

```
j.k.@Jacks-MacBook-Pro-2 hw_purrr % git  add hw_lubridate_purrr.Rmd
j.k.@Jacks-MacBook-Pro-2 hw_purrr % git commit -m "Exercise 2: Complex Date Arithmetic"

[main 2538cf4] Exercise 2: Complex Date Arithmetic
 1 file changed, 16 insertions(+)
j.k.@Jacks-MacBook-Pro-2 hw_purrr % git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 554 bytes | 554.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:jkuoc183/hw_purrr.git
   9d2a9b9..2538cf4  main -> main
```



Exercise 3: Higher-Order Functions with purrr
Question 3:
Using map() and map_dbl(), compute the mean, median, and standard deviation for each
numeric vector in the following list:
num_lists <- list(c(4, 16, 25, 36, 49), c(2.3, 5.7, 8.1, 11.4), c(10, 20, 30, 40, 50))

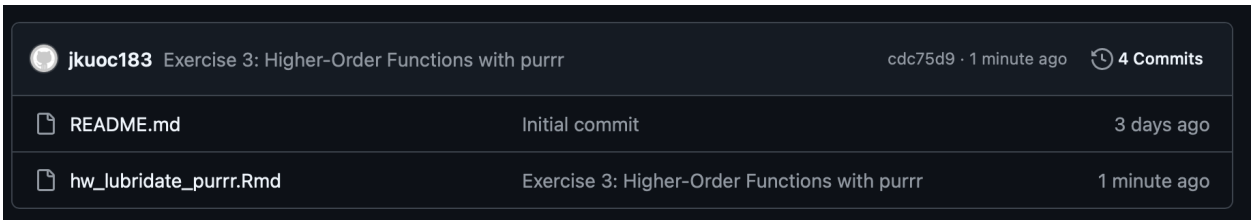
```r
#Question 3:
#Using map() and map_dbl(), compute the mean, median, and standard deviation for each
#numeric vector in the following list:
#num_lists <- list(c(4, 16, 25, 36, 49), c(2.3, 5.7, 8.1, 11.4), c(10, 20, 30, 40,
50))


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

stats <- tibble(
  mean = map_dbl(num_lists, mean),
  median = map_dbl(num_lists, median),
  sd = map_dbl(num_lists, sd)
)

print(stats)
```

```
j.k.@MacBookAir ~ % cd hw_purrr
j.k.@MacBookAir hw_purrr % git add hw_purr
fatal: pathspec 'hw_purr' did not match any files
j.k.@MacBookAir hw_purrr % git add hw_lubridate_purrr.Rmd
j.k.@MacBookAir hw_purrr % git commit -m "Exercise 3: Higher-Order Functions wit
h purrr"
[main cdc75d9] Exercise 3: Higher-Order Functions with purrr
 1 file changed, 23 insertions(+), 1 deletion(-)
j.k.@MacBookAir hw_purrr % git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 692 bytes | 692.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:jkuoc183/hw_purrr.git
   2538cf4..cdc75d9  main -> main
```



Exercise 4: Combining lubridate and purrr
Question 4:
Given a list of mixed date formats, use map() and possibly() from purrr to safely convert them to Date format and extract the month name.
date_strings <- list("2023-06-10", "2022/12/25", "15-Aug-2021", "InvalidDate")

```r
#Question 4:
#Given a list of mixed date formats, use map() and possibly() from purrr to safely convert
#them to Date format and extract the month name.
#date_strings <- list("2023-06-10", "2022/12/25", "15-Aug-2021", "InvalidDate")

date_strings <- list("2023-06-10", "2022/12/25", "15-Aug-2021", "InvalidDate")

extract_month <- function(date_str) {

 d <- parse_date_time(date_str, orders =c("ymd", "dmy"))

 if(is.na(d)) {
  stop("Date conversion failed")
 }
 month(d, label = TRUE, abbr = FALSE) %>% as.character()
}

safe_extract_month <- possibly(extract_month, otherwise = NA_character_)

month_names <- map(date_strings, safe_extract_month)

print(month_names)
```

```
[j.k.@MacBookAir hw_purrr % git add hw_lubridate_purrr.Rmd                    ]
[j.k.@MacBookAir hw_purrr % git commit -m "Exercise 4: Combining lubridate and pu]
rrr"
[main faf5bf5] Exercise 4: Combining lubridate and purrr
 1 file changed, 26 insertions(+)
[j.k.@MacBookAir hw_purrr % git push                                          ]
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 732 bytes | 732.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:jkuoc183/hw_purrr.git
   cdc75d9..faf5bf5  main -> main
```