date_map

Loading Libraries

```r
library(lubridate)
```

```

Attaching package: 'lubridate'

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

    date, intersect, setdiff, union
```

```r
library(purrr)
```

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
# solution
date_sequence <- seq(ymd("2015-01-01"), ymd("2025-12-31"), by = "2 months")
head(date_sequence)
```

```
[1] "2015-01-01" "2015-03-01" "2015-05-01" "2015-07-01" "2015-09-01"
[6] "2015-11-01"
```

```
# extracting date year qyarter and iso week
date_info <- data.frame(
  Date = date_sequence,
  Year = year(date_sequence),
  Quarter = quarter(date_sequence),
  ISO_Week = isoweek(date_sequence)
)
head(date_info)
```

```
        Date Year Quarter ISO_Week
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
```

Question 2

Given the following dates, compute the difference in months and weeks between each consecutive pair.

```
# Loading in the data
sample_dates <- c("2018-03-15", "2020-07-20", "2023-01-10", "2025-09-05")
sample_dates <- ymd(sample_dates)

# Dividing the dates by month w/a remainder
months_diff <- interval(sample_dates[1:3], sample_dates[2:4]) %/% months(1)

# Dividing the dates by month and week w/a remainder
weeks_diff <- interval(sample_dates[1:3], sample_dates[2:4]) %% months(1) %/% weeks(1)
head(months_diff)
```

```
[1] 28 29 31
```

```
head(weeks_diff)
```

```
[1] 0 3 3
```

```
# I think I am probably missing a command here which would make this more simple, but this so
```

```
question_2 <- data.frame(
  Start = sample_dates[1:3],
  End = sample_dates[2:4],
  Full_Months = months_diff,
  Remaining_Weeks = weeks_diff
)

question_2
```

```
       Start        End Full_Months Remaining_Weeks
1 2018-03-15 2020-07-20          28               0
2 2020-07-20 2023-01-10          29               3
3 2023-01-10 2025-09-05          31               3
```

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))

# df of mean median and sd
question_3 <- data.frame(
  Mean = map_dbl(num_lists, mean),
  Median = map_dbl(num_lists, median),
  SD = map_dbl(num_lists, sd)
)

head(question_3)
```

```
    Mean Median       SD
1 26.000   25.0 17.42125
2  6.875    6.9  3.84220
3 30.000   30.0 15.81139
```

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")

#converting
converted <- map(date_strings, possibly(~parse_date_time(.x, orders = c("ymd", "dmy")), other
```

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

```
head(converted)
```

```
[[1]]
[1] "2023-06-10 UTC"

[[2]]
[1] "2022-12-25 UTC"

[[3]]
[1] "2021-08-15 UTC"

[[4]]
[1] NA
```

```
#converting to month name
month_names <- sapply(converted, possibly(months, otherwise = NA))
head(month_names)
```

```
[1] "June"     "December" "August"   NA
```