Date_map

Jojo

2025-02-25

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(purrr)
library(tidyverse)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr   1.1.4     v stringr 1.5.1
## v forcats 1.0.0     v tibble  3.2.1
## v ggplot2 3.5.1     v tidyr   1.3.0
## v readr   2.1.5
```

```
## -- 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 become errors
```

```
#1.Wrote this code and used the data.frame command to present it well
date <- seq(ymd("2015-01-01"), ymd("2025-12-31"), by="2 months")
data.frame(year = year(date),
           quart = quarter(date),
           iso = isoweek(date)
           )
```

```
##    year quart iso
## 1  2015     1   1
## 2  2015     1   9
## 3  2015     2  18
## 4  2015     3  27
## 5  2015     3  36
## 6  2015     4  44
```

```
## 7  2016     1  53
## 8  2016     1   9
## 9  2016     2  17
## 10 2016     3  26
## 11 2016     3  35
## 12 2016     4  44
## 13 2017     1  52
## 14 2017     1   9
## 15 2017     2  18
## 16 2017     3  26
## 17 2017     3  35
## 18 2017     4  44
## 19 2018     1   1
## 20 2018     1   9
## 21 2018     2  18
## 22 2018     3  26
## 23 2018     3  35
## 24 2018     4  44
## 25 2019     1   1
## 26 2019     1   9
## 27 2019     2  18
## 28 2019     3  27
## 29 2019     3  35
## 30 2019     4  44
## 31 2020     1   1
## 32 2020     1   9
## 33 2020     2  18
## 34 2020     3  27
## 35 2020     3  36
## 36 2020     4  44
## 37 2021     1  53
## 38 2021     1   9
## 39 2021     2  17
## 40 2021     3  26
## 41 2021     3  35
## 42 2021     4  44
## 43 2022     1  52
## 44 2022     1   9
## 45 2022     2  17
## 46 2022     3  26
## 47 2022     3  35
## 48 2022     4  44
## 49 2023     1  52
## 50 2023     1   9
## 51 2023     2  18
## 52 2023     3  26
## 53 2023     3  35
## 54 2023     4  44
## 55 2024     1   1
## 56 2024     1   9
## 57 2024     2  18
## 58 2024     3  27
## 59 2024     3  35
## 60 2024     4  44
```

```
## 61 2025     1   1
## 62 2025     1   9
## 63 2025     2  18
## 64 2025     3  27
## 65 2025     3  36
## 66 2025     4  44
```

```
#2. We wanted to take the sample dates and show the difference so we created a function to take the beg
sample_dates <- c("2018-03-15", "2020-07-20", "2023-01-10", "2025-09-05")
date_diffs <- tibble(
  start = sample_dates,
  end = lead(sample_dates)
) %>%
  filter(!is.na(end)) %>%
  mutate(
    diff_months = interval(start, end) / months(1),
    diff_weeks  = interval(start, end) / weeks(1)
  )
date_diffs
```

```
## # A tibble: 3 x 4
##   start      end        diff_months diff_weeks
##   <chr>      <chr>            <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.
```

```
#3. We took the numbers from the list and just simply used the mean, median and sd functions to find th
num_lists <- list(c(4, 16, 25, 36, 49), c(2.3, 5.7, 8.1, 11.4), c(10, 20, 30, 40, 50))

statis <- list(
  avg = map_dbl(num_lists, mean),
  med = map_dbl(num_lists, median),
  dev = map_dbl(num_lists, sd)
)
data.frame(statis)
```

```
##      avg  med      dev
## 1 26.000 25.0 17.42125
## 2  6.875  6.9  3.84220
## 3 30.000 30.0 15.81139
```

```
#4. We had to use the possibly function to safely convert these all to the date format.  Then we create
date_strings <- list("2023-06-10", "2022/12/25", "15-Aug-2021", "InvalidDate")
safe_parse_date <- possibly(function(x) {
  parse_date_time(x, orders = c("ymd", "mdy", "dmy", "d-b-Y"))
}, otherwise = NA)

parsed_dates <- map(date_strings, safe_parse_date)
```

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

```r
month_names <- map_chr(parsed_dates, function(dt) {
  if (is.na(dt)) {
    NA_character_
  } else {
    format(dt, "%B")
  }
})

month_names
```

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