date_map_data413

Sydney Ball

2025-02-24

#Github Link: https://github.com/sball32/date_map_hw.git

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.2
## -- 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
```

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.

```
#Generate the dates from 2015-01-01 for every 2 months in the future
date_seq <- seq(from = ymd("2015-01-01"),
                to = ymd("2025-12-31"),
                by = "2 months")

#Extract the year, quarter, and ISO week number
date_info <- data.frame(
  Date = date_seq,
  Year = year(date_seq),
  Quarter = quarter(date_seq),
  ISO_Week = isoweek(date_seq)
)

#Print to see the correct output
head(date_info, 15) #table is long, so only show 15 outputs
```

```
##          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
## 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
## 11 2016-09-01 2016       3       35
## 12 2016-11-01 2016       4       44
## 13 2017-01-01 2017       1       52
## 14 2017-03-01 2017       1        9
## 15 2017-05-01 2017       2       18
```

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")

#Parse into dates
sample_dates <- as.Date(c("2018-03-15", "2020-07-20", "2023-01-10", "2025-09-05"))

# Compute the difference in Months and Weeks
diff_results <- data.frame(
  Week_Date = time_length(diff(sample_dates), "week"),
  Month_Date = time_length(diff(sample_dates), "month")
  )

diff_results
```

```
##   Week_Date Month_Date
## 1  122.5714   28.18891
## 2  129.1429   29.70021
## 3  138.4286   31.83573
```

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))

#Compute Mean:
map_dbl(num_lists, mean)
```

```
## [1] 26.000  6.875 30.000
```

```r
#Compute Median:
map_dbl(num_lists, median)
```

```
## [1] 25.0  6.9 30.0
```

```r
#Compute Standard Deviation:
map_dbl(num_lists, sd)
```

```
## [1] 17.42125  3.84220 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.

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

# Parse the dates into correct formatting
months_datestrings <- possibly(function(x) month(parse_date_time(x, orders = c("ymd", "dmy")), label =

# Use map to return a list
month_names <- map(date_strings, months_datestrings)

# Print the final findings of the dates and their assigned months
final_months <-data.frame(
    Original = unlist(date_strings),
    Month = unlist(month_names)
    )

final_months
```

```
##      Original    Month
## 1  2023-06-10     June
## 2  2022/12/25 December
## 3 15-Aug-2021   August
## 4 InvalidDate     <NA>
```