Date_map

ShaoWei Lu

2025-02-24

```
library(lubridate)
```

```
##
## 載入套件：'lubridate'
```

```
## 下列物件被遮斷自 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(purrr)

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

data_sequence <- seq(ymd("2015-1-1"), ymd("2025-12-31"), by = ("2 month"))
data_information <- data.frame(
  Date = data_sequence,

  # Extract the year from each date
  Year = map_dbl(data_sequence, year),

  # Extract the quarter from each date
  Quarter = map_dbl(data_sequence, quarter),

  # Extract the ISO week number from each date
  ISOweek = map_dbl(data_sequence, isoweek)
)
data_information
```

```
##          Date Year Quarter ISOweek
## 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
## 16 2017-07-01 2017       3      26
## 17 2017-09-01 2017       3      35
## 18 2017-11-01 2017       4      44
## 19 2018-01-01 2018       1       1
## 20 2018-03-01 2018       1       9
## 21 2018-05-01 2018       2      18
## 22 2018-07-01 2018       3      26
## 23 2018-09-01 2018       3      35
## 24 2018-11-01 2018       4      44
## 25 2019-01-01 2019       1       1
## 26 2019-03-01 2019       1       9
## 27 2019-05-01 2019       2      18
## 28 2019-07-01 2019       3      27
## 29 2019-09-01 2019       3      35
## 30 2019-11-01 2019       4      44
## 31 2020-01-01 2020       1       1
## 32 2020-03-01 2020       1       9
## 33 2020-05-01 2020       2      18
## 34 2020-07-01 2020       3      27
## 35 2020-09-01 2020       3      36
## 36 2020-11-01 2020       4      44
## 37 2021-01-01 2021       1      53
## 38 2021-03-01 2021       1       9
## 39 2021-05-01 2021       2      17
## 40 2021-07-01 2021       3      26
## 41 2021-09-01 2021       3      35
## 42 2021-11-01 2021       4      44
## 43 2022-01-01 2022       1      52
## 44 2022-03-01 2022       1       9
## 45 2022-05-01 2022       2      17
## 46 2022-07-01 2022       3      26
## 47 2022-09-01 2022       3      35
## 48 2022-11-01 2022       4      44
## 49 2023-01-01 2023       1      52
## 50 2023-03-01 2023       1       9
## 51 2023-05-01 2023       2      18
## 52 2023-07-01 2023       3      26
## 53 2023-09-01 2023       3      35
## 54 2023-11-01 2023       4      44
```

```
## 55 2024-01-01 2024       1       1
## 56 2024-03-01 2024       1       9
## 57 2024-05-01 2024       2      18
## 58 2024-07-01 2024       3      27
## 59 2024-09-01 2024       3      35
## 60 2024-11-01 2024       4      44
## 61 2025-01-01 2025       1       1
## 62 2025-03-01 2025       1       9
## 63 2025-05-01 2025       2      18
## 64 2025-07-01 2025       3      27
## 65 2025-09-01 2025       3      36
## 66 2025-11-01 2025       4      44
```

```
#Exercise 2: Complex Date Arithmetic
#Question 2: Given the following dates, compute the difference in months and weeks between ea
ch consecutive pair.

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

#Calculate the difference in months between each consecutive pair of dates
months_difference <- time_length(interval(sample_dates[-length(sample_dates)], sample_dates[-
1]), "months")

#Compute the total week difference.
week_difference <- time_length(interval(sample_dates[-length(sample_dates)], sample_dates[-
1]), "weeks")

#Create a data frame to display the results.
Difference_in_months_and_weeks <- data.frame(
  Start_Date = sample_dates[-length(sample_dates)],
  End_Date = sample_dates[-1],
  Month_Difference = round(months_difference),
  Week_Difference = round(week_difference)
)
Difference_in_months_and_weeks
```

```
##   Start_Date   End_Date Month_Difference Week_Difference
## 1 2018-03-15 2020-07-20               28             123
## 2 2020-07-20 2023-01-10               30             129
## 3 2023-01-10 2025-09-05               32             138
```

```
#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))

#Compute the mean.
mean_values <- map_dbl(num_lists, mean)

#Compute the median.
median_values <- map_dbl(num_lists, stats::median)

#Compute the SD.
sd_values <- map_dbl(num_lists, sd)
data_indexs <- data.frame(
  means = mean_values,
  median = median_values,
  SD = sd_values
)
data_indexs
```

```
##    means median       SD
## 1 26.000   25.0 17.42125
## 2  6.875    6.9  3.84220
## 3 30.000   30.0 15.81139
```

```
#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.

Sys.setlocale("LC_TIME", "C")
```

```
## [1] "C"
```

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

#Create a safe function to parse dates while handling errors
safe_parse_date <- possibly(~ as.Date(.x, tryFormats = c("%Y-%m-%d", "%Y/%m/%d", "%d-%b-%Y")), NA)

#Convert date strings into Date format safely.
converted_dates <- map(date_strings, safe_parse_date)

#Extract the month name from each valid date.
map_chr(converted_dates, ~ ifelse(is.na(.x), "Invalid", as.character(month(.x, label = TRUE, abbr = TRUE))))
```

```
## [1] "Jun"     "Dec"     "Aug"     "Invalid"
```