Homework: lubridate and purrr

Luke Govan

2025-02-24

Exercise 1: Advanced Date Manipulation with lubridate 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

```
#Create the sequence of dates
dates <- seq(ymd("2015-01-01"), ymd("2025-12-31"), by = "2 months")

#Show the year quarter and iso_week_number
date_yqI <- data.frame(
  date = dates,
  year = year(dates),
  quarter = quarter(dates),
  iso_week_number = isoweek(dates)
)
print(date_yqI)
```

```
##          date year quarter iso_week_number
## 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 each consecutive pair.

```r
#Sample dates
sample_dates <- (c("2018-03-15", "2020-07-20", "2023-01-10", "2025-09-05"))
#Parse
sample_dates <- as.Date(c("2018-03-15", "2020-07-20", "2023-01-10", "2025-09-05"))
#Compute difference
difference_result <- data.frame(
  week_date = time_length(diff(sample_dates), "week"),
  month_date = time_length(diff(sample_dates), "month")
)

print(difference_result)
```

```
##   week_date month_date
## 1  122.5714   28.18891
## 2  129.1429   29.70021
## 3  138.4286   31.83573
```

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 list.

```
#Create a numerical 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))

#Use mapdbl to find the mean median and sd
map_dbl(num_lists, mean)
```

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

```
map_dbl(num_lists, median)
```

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

```
map_dbl(num_lists, sd)
```

```
## [1] 17.42125  3.84220 15.81139
```

Exercise 4: Combining lubridate and purr 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")

#parse and then extract the month name
#in order to extract the months actual name we need to use format %B
#because we have an invalid date we need to have a NA when one of the function outputs fails
extracted_month <- possibly(function(x){
  parsed_date <- parse_date_time(x, orders = c("ymd", "dmy", "mdy"), quiet = TRUE)
  format(parsed_date, "%B")
}, otherwise = NA)

#use map to apply the function
monthnames <- map(date_strings, extracted_month)

#print results
resulting_month_name <- data.frame(
  starting_mixed_dates = unlist(date_strings),
  month_name = unlist(monthnames)
)
print(resulting_month_name)
```

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