Homework: Lubridate and Purr

Danny Tapp

Question 1:

```
library(tidyverse)
```

```
# First we'll generate a sequence of dates going by 2 months
dates <- seq.Date(as.Date("2015-01-01"), as.Date("2025-12-31"),by = "2 months")

#Then we extract the year, quarter, and ISO week number from that sequence
extracted_year <- year(dates)

extracted_quarter <-quarter(dates)

extracted_ISO <- isoweek(dates)
```

Question 2:

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

```
### creates an empty vector to store the differences we'll calculate
week_diff <- numeric(3)

### for loop to go through each consecutive pair and calculate the difftime in weeks
### difftime is then stored in the empty vector week_diff
for (i in 1:3) {
  week_diff[i] <- difftime(sample_dates[i+1],sample_dates[i],units = "weeks")
}

week_diff
```

```
[1] 122.5714 129.1429 138.4286
```

```r
### creates an empty vector to store the differences we'll calculate
month_diff <- numeric(3)

### for loop but this time with time_length instead of difftime
### month difference stored in empty vector month_diff
for (i in 1:3) {
  month_diff[i] <- time_length(interval(sample_dates[i],sample_dates[i+1]),"months")
}

month_diff
```

```
[1] 28.16129 29.67742 31.83871
```

Question 3:

```r
### Provided list below
num_lists <- list(c(4,16,25,36,49),c(2.3,5.7,8.1,11.4), c(10,20,30,40,50))

### Getting the means, medians, and standard deviations for each of the list's vectors
### Using map() and assigning them to n_l_*function*
n_l_mean <- map(num_lists,mean)

n_l_median <- map(num_lists,median)

n_l_sd <- map(num_lists,sd)
```

Question 4:

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

### Creating a function with possibly() to check for each date in list
extract_month_name <- possibly(function(d_string) {
  date <- ymd(d_string) ### First checking to see if ymd words throughout the list
  if (is.na(date)) {
    date <- dmy(d_string)
  }
### If we get an NA with ymd, then we try with dmy through the list
  if (!is.na(date)) {
    return(format(date,"%B")) ### If we don't get NA and instead get a date,
### this will reformat it into returning the month's name
  } else {
    return(NA) ### This is to give us an NA if ymd and dmy don't work
  }
})

month_names <- map(date_strings,extract_month_name)
### map will have our newly created function go through all of date_strings
month_names ### print out those names
```

```
[[1]]
[1] "June"

[[2]]
[1] "December"

[[3]]
[1] "August"

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