

STA 445 - Assignment #5

Richard McCormick

2023-10-18

Convert the following to date or date/time objects.

a. September 13, 2010.

```
date <- mdy("September 13, 2010")
date
```

```
## [1] "2010-09-13"
```

b. Sept 13, 2010.

```
date <- mdy( sub( "t", "", "Sept, 13 2010" ) )
date
```

```
## [1] "2010-09-13"
```

c. Sep 13, 2010.

```
date <- mdy("Sep 13, 2010")
date
```

```
## [1] "2010-09-13"
```

d. S 13, 2010. Comment on the month abbreviation needs.

```
date <- mdy( sub( "S", "Sep", "S 13, 2010" ) )
date
```

```
## [1] "2010-09-13"
```

Months need to be properly abbreviated, or use integer abbreviations. There is no way for lubidater to recognize the month from a single character, or from improperly abbreviated characters. If dates are improperly formatted, we need to make them match a recognized pattern.

e. 07-Dec-1941.

```
date <- dmy("07-Dec-1941")
date
```

```
## [1] "1941-12-07"
```

f. 1-5-1998. Comment on why you might be wrong.

```
date <- mdy("1-5-1998")
date
```

```
## [1] "1998-01-05"
```

This answer might be wrong because it is possible the first two digits representing month and day might be interchanged with each other. It isn't possible to know which of them is the day and which is the month.

g. 21-5-1998. Comment on why you know you are correct.

```
date <- dmy("21-5-1998")
date
```

```
## [1] "1998-05-21"
```

We know that this format is correct because there are only 12 months, so the first position with a value of 21 has to be the day.

h. 2020-May-5 10:30 am

```
date <- ymd_hm("2020-May-5 10:30 am")
date
```

```
## [1] "2020-05-05 10:30:00 UTC"
```

i. 2020-May-5 10:30 am PDT (ex Seattle)

```
date <- ymd_hm("2020-May-5 10:30 am PDT", tz="US/Pacific")
date
```

```
## [1] "2020-05-05 10:30:00 PDT"
```

j. 2020-May-5 10:30 am AST (ex Puerto Rico)

```
date <- ymd_hm("2020-May-5 10:30 am AST", tz="America/Puerto_Rico")  
date
```

```
## [1] "2020-05-05 10:30:00 AST"
```

2. Using just your date of birth (ex Sep 7, 1998) and today's date calculate the following Write your code in a manner that the code will work on any date after you were born.:

```
birthday <- mdy("June 21, 2000")
birthday
```

```
## [1] "2000-06-21"
```

a. Calculate the date of your 64th birthday.

```
birthday_64 <- birthday + dyears( 64 )
birthday_64
```

```
## [1] "2064-06-21"
```

b. Calculate your current age (in years). Hint: Check your age is calculated correctly if your birthday was yesterday and if it were tomorrow!

```
now <- lubridate::now()
current_age <- interval( birthday, now )

print(
  paste( round( as.numeric( as.duration( current_age ), 'years' ), 2 ),
    "years old." ) )
```

```
## [1] "23.34 years old."
```

c. Using your result in part (b), calculate the date of your next birthday.

```
next.bday <- birthday

while( next.bday < now )
{
  next.bday <- next.bday + dyears( 1 )
}

next.bday
```

```
## [1] "2024-06-21 UTC"
```

d. The number of days until your next birthday.

```
as.period( interval(now, next.bday ), unit='day' )
```

```
## [1] "241d 7H 40M 36.2017729282379S"
```

e. The number of months and days until your next birthday.

```
as.period( interval(now, next.bday ), unit='month' )
```

```
## [1] "7m 28d 7H 40M 36.2017729282379S"
```

3. Suppose you have arranged for a phone call to be at 3 pm on May 8, 2015 at Arizona time. However, the recipient will be in Auckland, NZ. What time will it be there?

```
az.time <- mdy_h( "May 8, 2015 3 pm", tz="US/Arizona" )
nz.time <- with_tz( time=az.time, tzzone="Pacific/Auckland" )

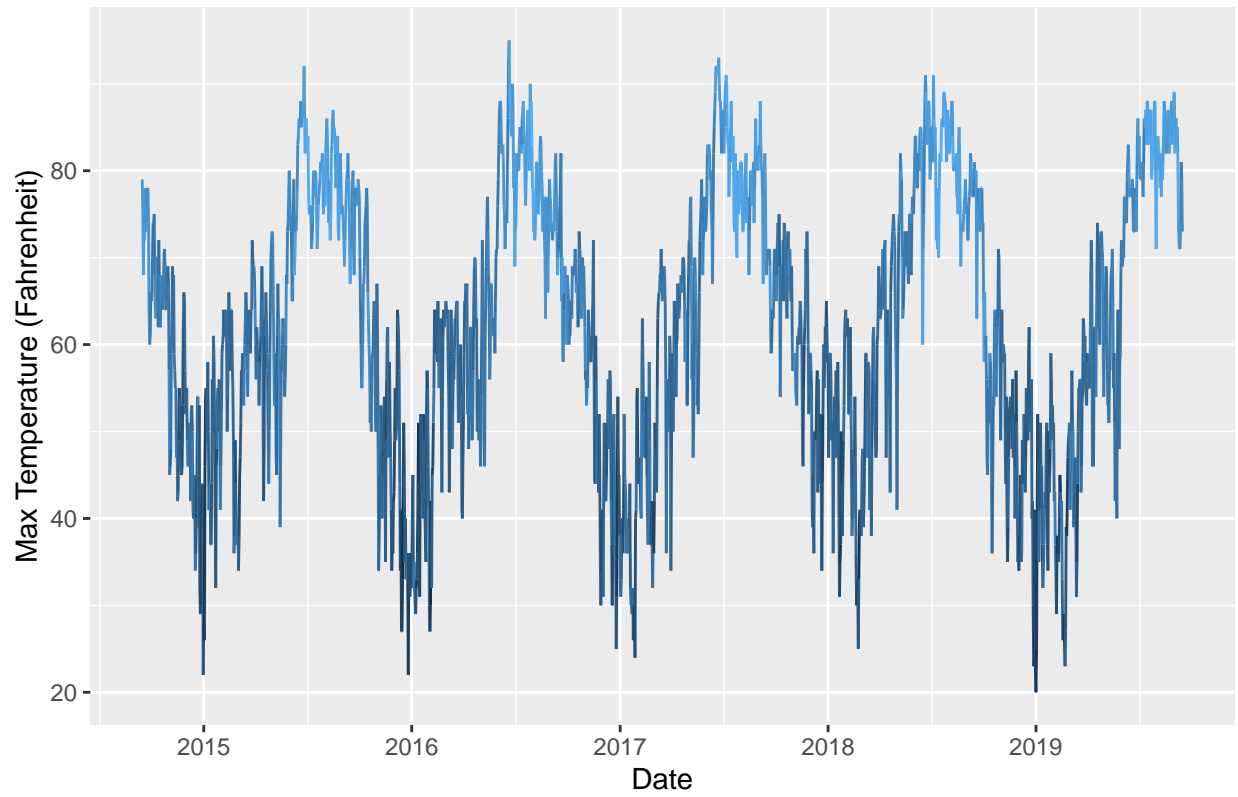
nz.time
```

```
## [1] "2015-05-09 10:00:00 NZST"
```

4. From this book's GitHub directory, navigate to the data-raw directory and then download the Pulliam_Airport_Weather_Station.csv data file. (There are several weather station files. Make sure you get the correct one!) There is a DATE column (is it of type date when you import the data?) as well as the Maximum and Minimum temperature. For the last 5 years of data we have (exactly, not just starting at Jan 1, 2014!), plot the time series of daily maximum temperature with date on the x-axis. Write your code so that it will work if I update the dataset. Hint: Find the maximum date in the data set and then subtract 5 years. Will there be a difference if you use dyears(5) vs years(5)? Which seems more appropriate here?

```
suppressWarnings({  
  # Import the data from CSV sheet  
  weather.data <- read_csv( "Pulliam_Airport_Weather_Station.csv",  
                           show_col_types = FALSE )  
  
  # Modify the data - select only the 3 columns needed  
  weather.data <- weather.data[ c( "DATE", "TMIN", "TMAX" ) ]  
  max.date <- tail( weather.data[ "DATE" ], 1 )[1,] # Get the most recent date  
  min.date <- max.date[ 1 , 1 ] - dyears( 5 ) # Find the date 5 years before  
  min.date$DATE <- as.Date(min.date$DATE)  
  
  # Filter the data so only the 5 most recent years are in the dataframe.  
  weather.data <- weather.data %>% filter( DATE>min.date )  
  # Code that generates warning messages  
})  
weather_plot <- ggplot( data=weather.data ) +  
  geom_line( aes( x=DATE, y=TMAX, color=TMIN ), show.legend = FALSE ) +  
  labs( title="Pulliam Airport Max Temperature (5 Year)",  
        x="Date",  
        y="Max Temperature (Fahrenheit)" )  
  
weather_plot
```

Pulliam Airport Max Temperature (5 Year)



5. It turns out there is some interesting periodicity regarding the number of births on particular days of the year.

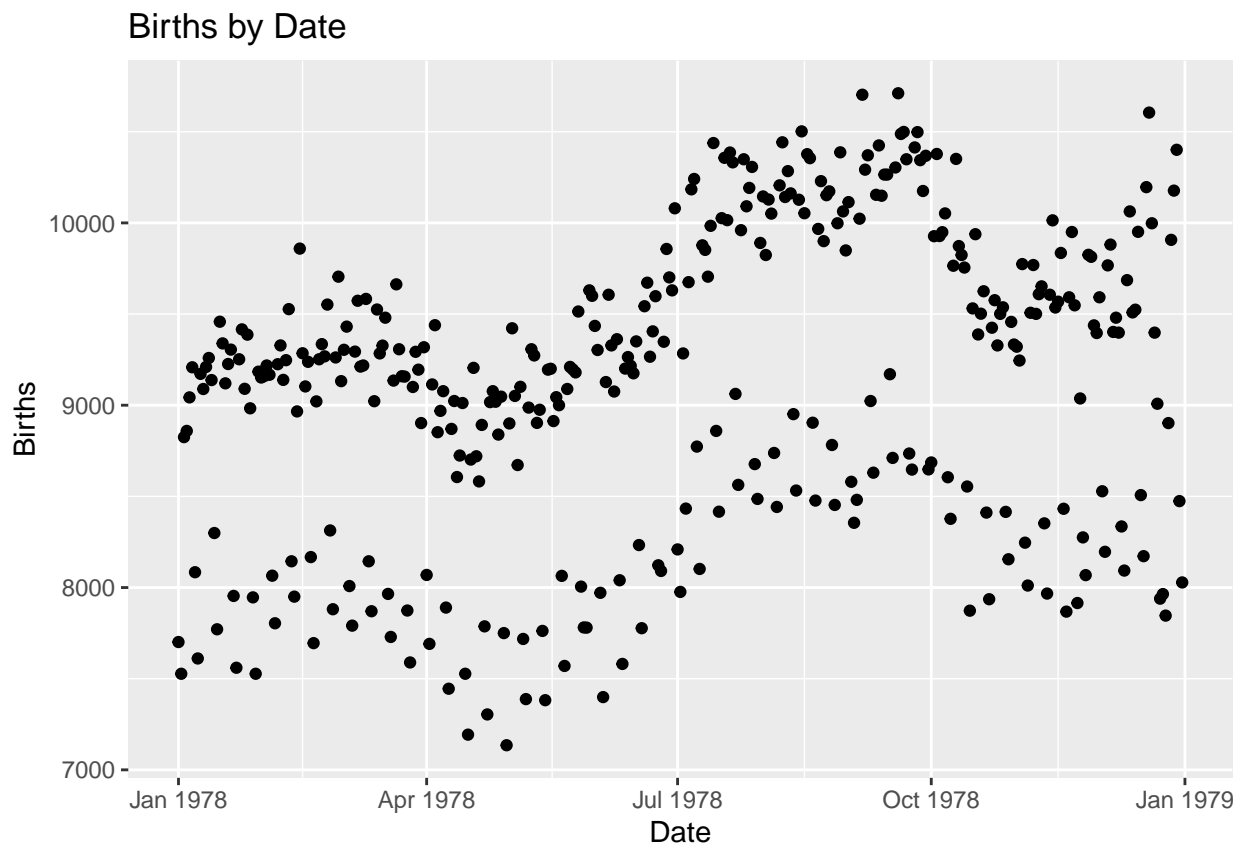
a. Using the mosaicData package, load the data set Births78 which records the number of children born on each day in the United States in 1978. Because this problem is intended to show how to calculate the information using the date, remove all the columns except date and births.

```
birth.data <- Births78
birth.data <- birth.data[ c( "date", "births" ) ]
```

b. Graph the number of births vs the date with date on the x-axis. What stands out to you? Why do you think we have this trend?

```
birth.plot <- ggplot( data=birth.data ) +
  geom_point( aes( x=date, y=births ) ) +
  labs( title="Births by Date", x="Date", y="Births" )

birth.plot
```



It seems that there are two separate groups of births that both follow roughly the same pattern. It could be that there are significantly less births on weekends rather than on week days.

c. To test your assumption, we need to figure out the what day of the week each observation is. Use `dplyr::mutate` to add a new column named `day` that is the day of the week (Monday, Tuesday, etc). This calculation will involve some function in the `lubridate` package and the `date` column.

```
birth.data <- birth.data %>% mutate( week.day = wday( date, label=TRUE ) )
```

d. Plot the data with the point color being determined by the day of the week variable.

```
birth.plot <- ggplot( data=birth.data ) +  
  geom_point( aes( x=date, y=births, color=week.day ) ) +  
  labs( title="Births by Date", x="Date", y="Births", color="Week Day" )  
  
birth.plot
```

