Module12

Joe Vargovich

10/5/2020

Exercise 1 - For the following formats for a date, transform them into a date/time object. Which formats can be handled nicely and which are not?

```
#NOTE: Year-Month-Day-Hours-Minutes-Secs (ymd) is standard for dates and time storage.

#a. September 13th
birthday <- c(
  'September 13, 1978',
  'Sept 13, 1978',
  'Sep 13, 1978',
  'S 13, 1978',
  '9-13-78',
  '9-13/78',
  '9/13/78')

mdy(birthday)
```

```
## [1] "1978-09-13" "1978-09-13" "1978-09-13" "1978-09-13" "1978-09-13"
## [6] "1978-09-13" "1978-09-13"
```

```
#HANDLED: This format worked fine with the mdy() function in lubridate.
#This utilized the rounding down to parse 78 to 1978.

#b.
mdy('Sept 13, 1978')  # Why does this error out when it worked above?
```

```
## Warning: All formats failed to parse. No formats found.
```

```
## [1] NA
```

```
#ERRORED: The proper prefix for September is Sep, not Sept.
#The mdy on a vector seems to handle this case, while the mdy
# for a single string does not.

#c.
birthday <- c(
  'June 13, 1978',
  'J 13, 1978')

mdy(birthday)
```

```
## [1] "1978-06-13" "1978-07-13"
```

```r
#ERRORED: The bottom date is supposed to have its month be
#June (06) but it is autoconverted to July (07) due to ambiguity.
#July is considered to be earlier in the alphabet than June.
```

Exercise 2 - Birthday caluclations

```r
#Input my birthday and today's date.
birthday = mdy('April 21,1999')
today = lubridate::today()
#Create an interval from my birthday to today.
lifeInterval = birthday %--% today

lifeInterval
```

```
## [1] 1999-04-21 UTC--2020-10-11 UTC
```

```r
as.duration(lifeInterval)
```

```
## [1] "677721600s (~21.48 years)"
```

```r
#a. My next birthday
nextBirthday = update(birthday,year=2021)
nextBirthday
```

```
## [1] "2021-04-21"
```

```r
#b. Number of days until my next birthday
numDays = as.period(today %--% nextBirthday, unit='days')
numDays
```

```
## [1] "192d 0H 0M 0S"
```

```r
#c. Number of months and days until my next birthday
numMonths = as.period(today %--% nextBirthday, unit='months')
numMonths
```

```
## [1] "6m 10d 0H 0M 0S"
```

```r
#d. Date of my 64th birthday
imOld = birthday + years(64)
imOld
```

```
## [1] "2063-04-21"
```

```r
#e. Number of years, months, and days until Im Old.
timeLeft = as.period(today %--% imOld, unit='years')
timeLeft
```

```
## [1] "42y 6m 10d 0H 0M 0S"
```

Exercise 3 - AZ to New Zealand Timezone converison

```r
#a. 3 pm AZ time to Auckland, NZ time.
meeting = ymd_hm('2015-5-8 15:00', tz='US/Arizona')
  with_tz(meeting, 'Pacific/Auckland')
```

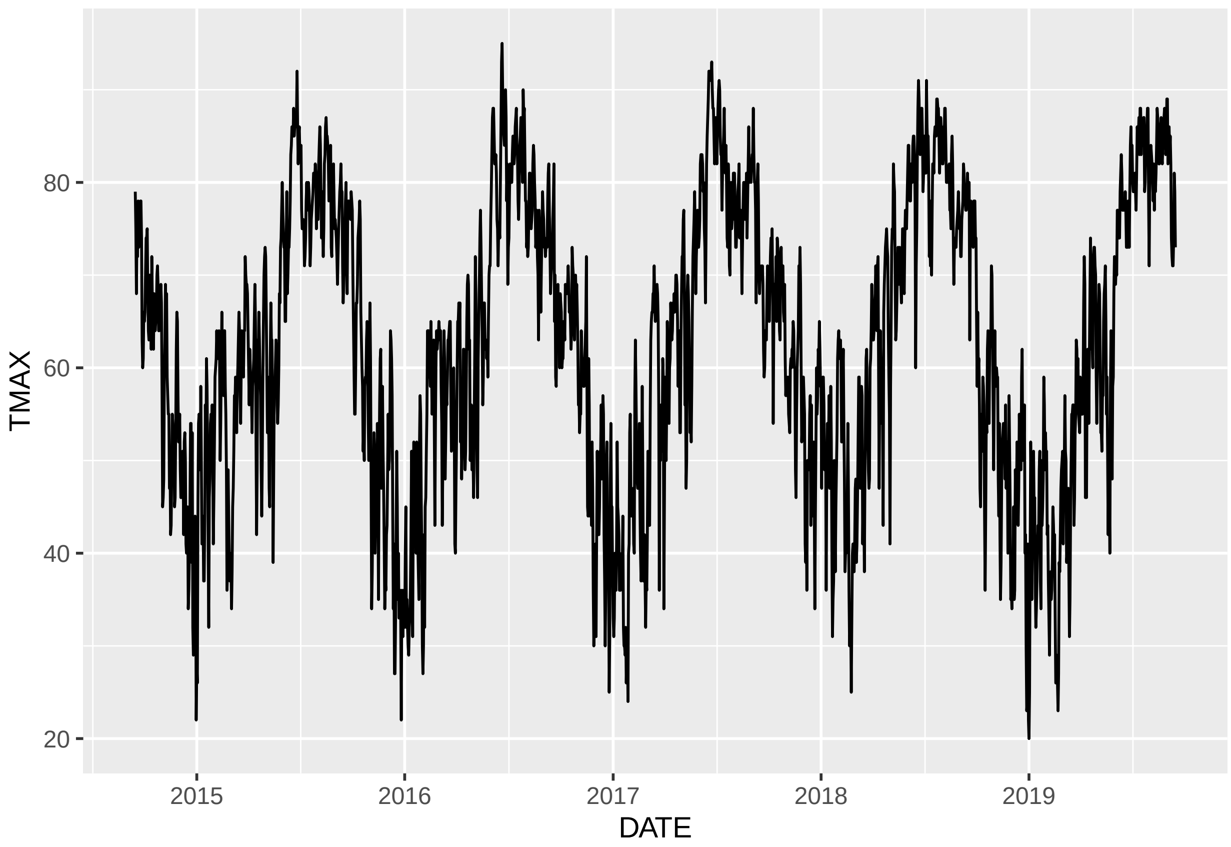
```
## [1] "2015-05-09 10:00:00 NZST"
```

Exercise 4 - Weather Station Plot Of Max Temperature

```
weatherData = invisible(read_csv("Pulliam_Airport_Weather_Station.csv"))

#Clean it up! Select just the columns we need.
weatherData = weatherData %>%
  select(DATE,TMAX)

#Convert the doubles in the date column to date objects.
weatherData = weatherData %>% mutate(
  DATE = ymd(DATE)
)
#Grab the largest date in the dataframe.
largestDate = tail(weatherData$DATE, n=1)

#Use dyears as we want an objective calculation of the temperature readings.
fiveYearsBefore = largestDate - dyears(5)

#Filter to the dates we want between the max and five years before.
fiveYearData = weatherData %>%
  filter(DATE > fiveYearsBefore  )

#Finally, plot the max temperatures over the past five years.
temperaturePlot = ggplot(fiveYearData, aes(x=DATE, y=TMAX)) +
  geom_line()

head(fiveYearData)
```

```
## # A tibble: 6 x 2
##   DATE        TMAX
##   <date>     <dbl>
## 1 2014-09-15    79
## 2 2014-09-16    75
## 3 2014-09-17    68
## 4 2014-09-18    75
## 5 2014-09-19    72
## 6 2014-09-20    78
```

```
#The plot makes sense and cyclically reflects the seasons.
temperaturePlot
```



Exercise 5 - Births on each day of the week throughout the year.

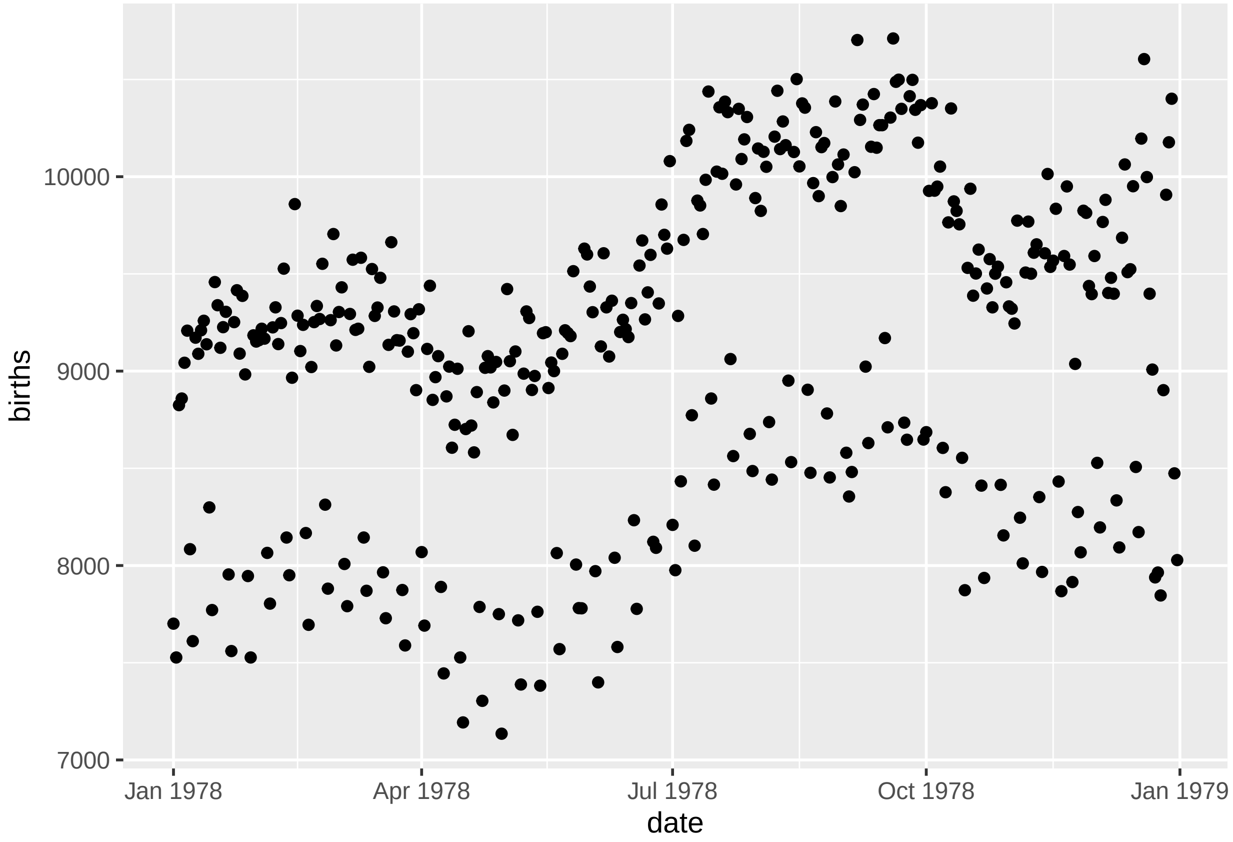
```
#a. Load the data
data('Births78', package='mosaicData')

head(Births78)
```

```
##         date births wday year month day_of_year day_of_month day_of_week
## 1 1978-01-01   7701  Sun 1978     1           1            1           1
## 2 1978-01-02   7527  Mon 1978     1           2            2           2
## 3 1978-01-03   8825  Tue 1978     1           3            3           3
## 4 1978-01-04   8859  Wed 1978     1           4            4           4
## 5 1978-01-05   9043  Thu 1978     1           5            5           5
## 6 1978-01-06   9208  Fri 1978     1           6            6           6
```

```
# Select only the dates and births
Births78 = Births78 %>%
  select(date, births)
head(Births78)
```

```
##         date births
## 1 1978-01-01   7701
## 2 1978-01-02   7527
## 3 1978-01-03   8825
## 4 1978-01-04   8859
## 5 1978-01-05   9043
## 6 1978-01-06   9208
```

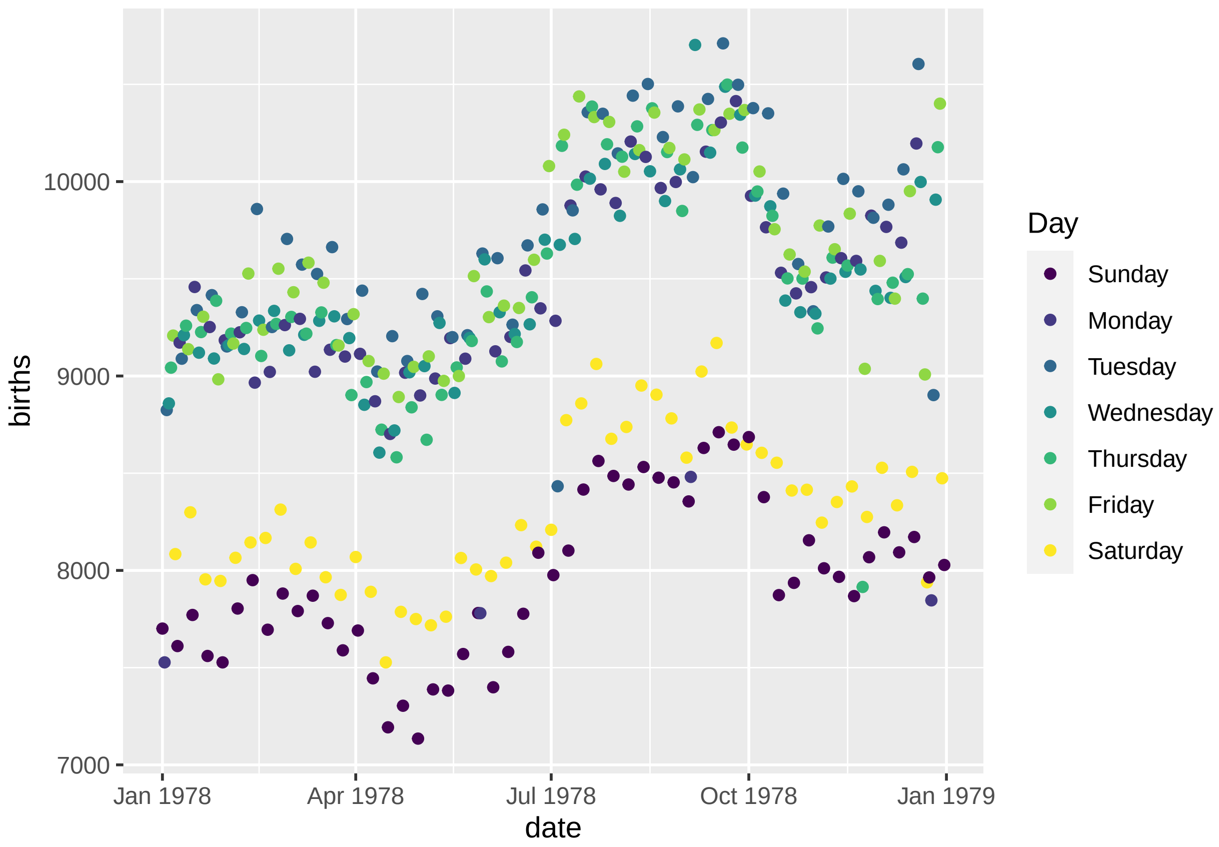
```
#b. Plot date and births with a scatterplot
birthsPlot = ggplot(Births78, aes(x=date, y=births)) +
  geom_point()
birthsPlot
```



```
#c. Add a column to Births78 to represent the day of the week.
Births78 = Births78 %>% mutate(
  Day = wday(date, label=TRUE, abbr=FALSE))
head(Births78)
```

```
##         date births       Day
## 1 1978-01-01   7701    Sunday
## 2 1978-01-02   7527    Monday
## 3 1978-01-03   8825   Tuesday
## 4 1978-01-04   8859 Wednesday
## 5 1978-01-05   9043  Thursday
## 6 1978-01-06   9208    Friday
```

```
#d. Plot the data again but this time label each day by color.
birthsPlot = ggplot(Births78, aes(x=date, y=births)) +
  geom_point(aes(color=Day))

birthsPlot
```



```
#Explanation: Most births are planned, so they take place on weekdays.
#Some months are busier than others overall.
```