STA_445_Assignment_6

Connor Phillips

3/28/2024

```r
library(tidyverse)
library(lubridate)
library(mosaicData)
library(ggplot2)
```

Problem 1

Convert the following to date or date/time objects.

a. September 13, 2010.

```r
my.date = "September 13, 2010"
date.converted = mdy(my.date)
date.converted
```

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

```r
class(date.converted)
```

```
## [1] "Date"
```

b. Sept 13, 2010.

This abbreviation doesn't work (needs to be a three-character abbreviation, ex: Sep)

```r
my.date = "Sept 13, 2010"
date.converted = mdy(my.date)
date.converted
```

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

```r
class(date.converted)
```

```
## [1] "Date"
```

c. Sep 13, 2010.

```r
my.date = "Sep 13, 2010"
date.converted = mdy(my.date)
date.converted
```

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

```r
class(date.converted)
```

```
## [1] "Date"
```

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

This abbreviation doesn't work (needs to be a three-character abbreviation, ex: Sep)

```r
my.date = "S 13, 2010"
date.converted = mdy(my.date)
date.converted
```

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

```r
class(date.converted)
```

```
## [1] "Date"
```

e. 07-Dec-1941.

```r
my.date = "07-Dec-1941"
date.converted = dmy(my.date)
date.converted
```

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

```r
class(date.converted)
```

```
## [1] "Date"
```

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

I could be wrong because this could either mean January 5, 1998 or May 1, 1998.

```r
my.date = " 1-5-1998"
date.converted = mdy(my.date)
date.converted
```

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

```r
class(date.converted)
```

```
## [1] "Date"
```

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

I know this is correct because there is no 21st month, it can only mean May 21st.

```r
my.date = " 21-5-1998"
date.converted = dmy(my.date)
date.converted
```

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

```r
class(date.converted)
```

```
## [1] "Date"
```

h. 2020-May-5 10:30 am

```r
my.date = "2020-May-5 10:30 am"
date.converted = ymd_hm(my.date)
date.converted
```

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

```r
class(date.converted)
```

```
## [1] "POSIXct" "POSIXt"
```

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

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

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

```r
class(date.converted)
```

```
## [1] "POSIXct" "POSIXt"
```

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

```r
my.date = "2020-May-5 10:30 am AST"
date.converted = ymd_hm(my.date, tz="America/Anguilla")
date.converted
```

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

```r
class(date.converted)
```

```
## [1] "POSIXct" "POSIXt"
```

Problem 2

Using just your date of birth (ex Sep 7, 1998) and today's date calculate the following:

```r
my.birthday = mdy('May 20, 1993')
date.today = today()
my.birthday
```

```
## [1] "1993-05-20"
```

```r
date.today
```

```
## [1] "2024-04-02"
```

a. Calculate the date of your 64th birthday.

```r
my.birthday + years(64)
```

```
## [1] "2057-05-20"
```

b. Calculate your current age (in years).

```r
old.man = year(as.period(my.birthday %--% date.today))
old.man
```

```
## [1] 30
```

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

```r
next_bday = my.birthday + years(as.numeric(old.man)) + years(1)
next_bday
```

```
## [1] "2024-05-20"
```

d. The number of *days* until your next birthday.

```r
as.period(date.today %--% next_bday, unit= 'day')
```

```
## [1] "48d 0H 0M 0S"
```

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

```r
as.period(date.today %--% next_bday)
```

```
## [1] "1m 18d 0H 0M 0S"
```

Problem 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?

```r
call.time = mdy_hm("05/08/2015 3:00 pm", tz="US/Arizona")
with_tz(call.time, tzone = "Pacific/Auckland")
```

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

Problem 4

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`.

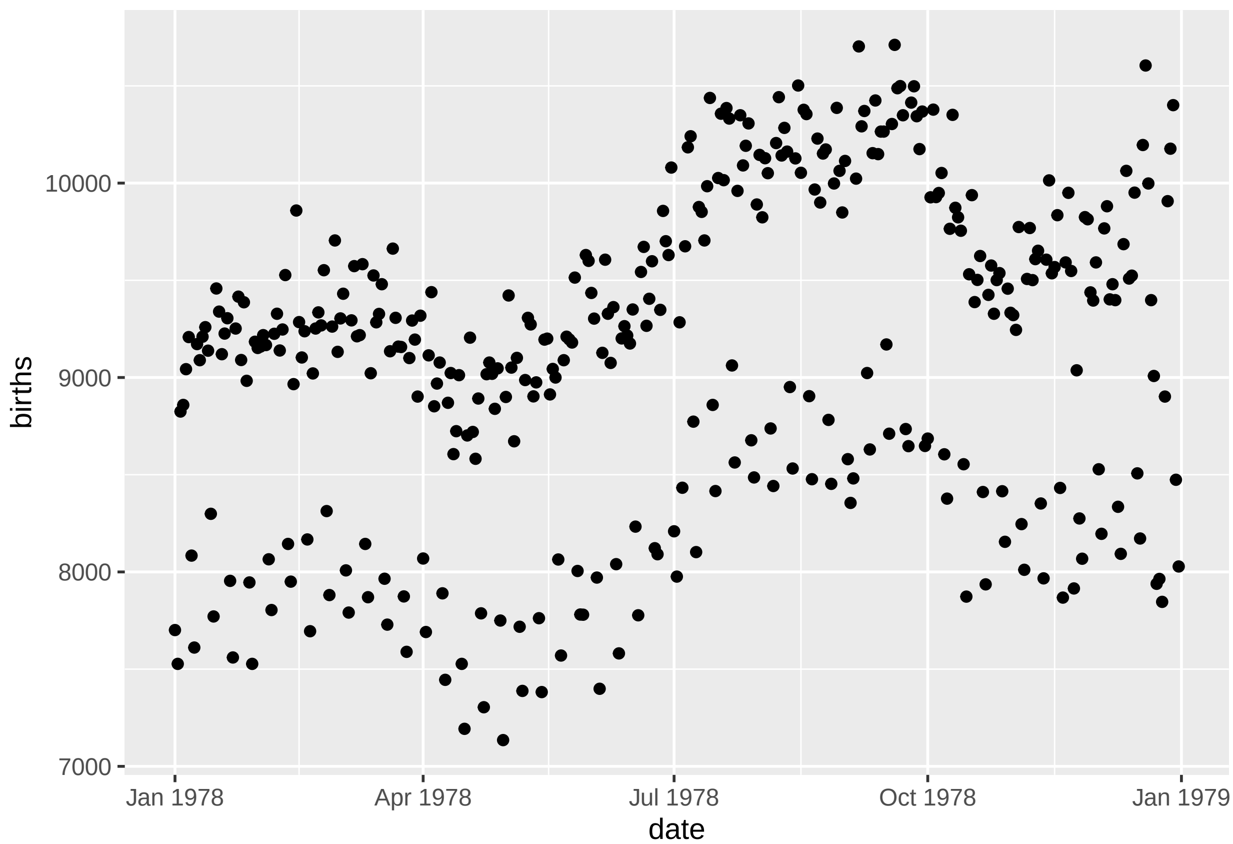
```r
p4data = Births78 %>%
  select('date', 'births')
head(p4data)
```

```
##         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. 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?

There is a clear break in the data with lower number of births vs higher number of births. We can see if this is due to the day of the week, with the higher break in the data being certain days of the week.

```r
ggplot(p4data, aes(y=births, x=date)) +
  geom_point()
```



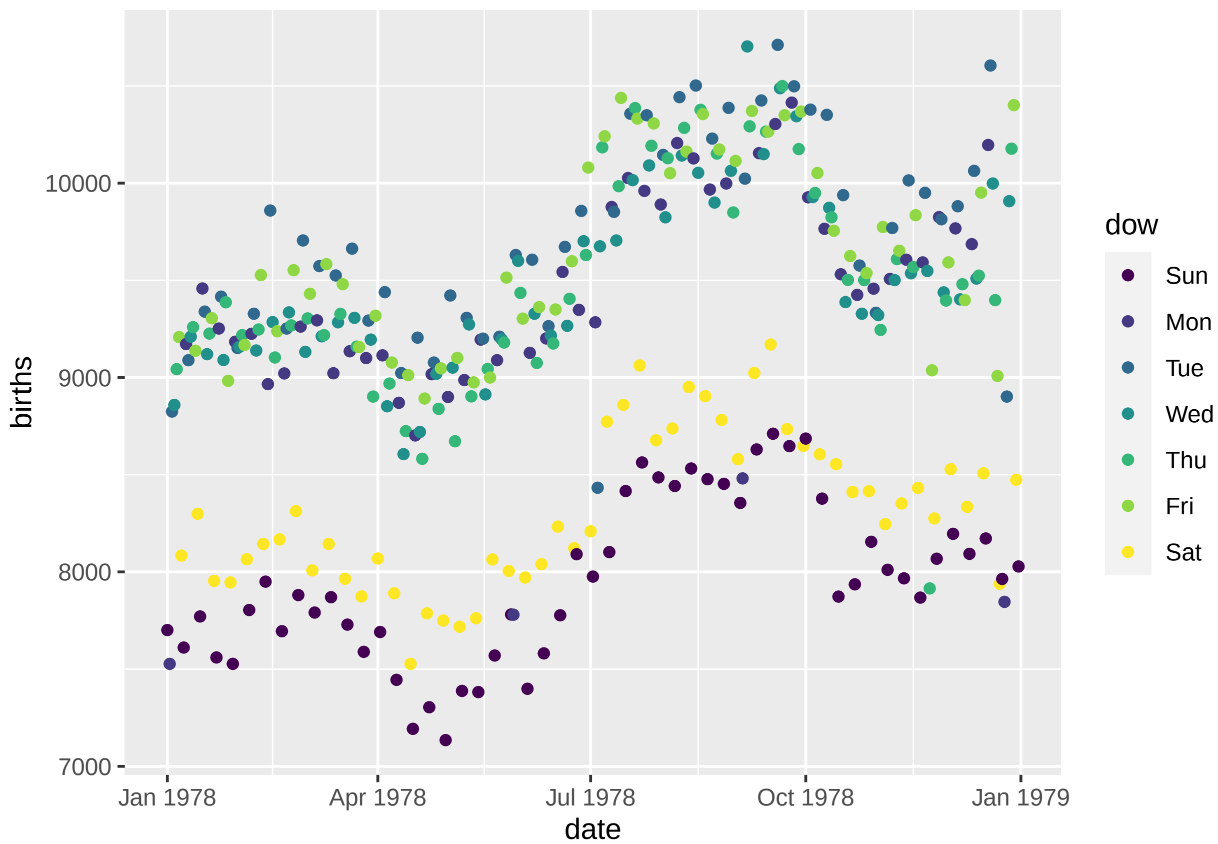
c. To test your assumption, we need to figure out the what day of the week each observation is. Use `dplyr::mutate` to add a new column named `dow` that is the day of the week (Monday, Tuesday, etc). This calculation will involve some function in the `lubridate` package and the `date` column.

```
p4data = p4data %>%
  mutate(dow = wday(date, label=TRUE))
```

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

Here we can clearly see that most births are scheduled to happen on weekdays, not weekends.

```
ggplot(p4data, aes(y=births, x=date)) +
  geom_point(aes(color=dow))
```



births
10000
9000
8000
7000
Jan 1978
Apr 1978
Jul 1978
Oct 1978
Jan 1979
date
dow
Sun
Mon
Tue
Wed
Thu
Fri
Sat