STA_445_Assignment_6

Matthew Hashim

4/2/2024

```
library(tidyverse)
library(lubridate)
```

Problem 1

Convert the following to date or date/time objects.

a. September 13, 2010.

```
mdy("September 13, 2010")
```

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

b. Sept 13, 2010.

```
mdy("Sept 13, 2010.") # A level requires a 3 char abbriviation.
```

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

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

c. Sep 13, 2010.

```
mdy("Sep 13, 2010.")
```

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

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

```
mdy("S 13, 2010 ")
```

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

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

A month abbreviation required only the first 3 characters of the month

e. 07-Dec-1941.

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

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

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

```
dmy("1-5-1998") # The 1 and 5 could both be the month and day
```

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

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

```
dmy("21-5-1998") # 21 is too large to be a month, therefor this is the only possible way
```

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

h. 2020-May-5 10:30 am

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

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

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

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

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

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

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

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

Problem 2

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

```
Bday <- mdy("Aug, 25, 2004")
```

a. Calculate the date of your 64th birthday.

```
Bday + years(64)
```

```
## [1] "2068-08-25"
```

b. Calculate your current age (in years).

```
date <- as.period( Bday %--% today() )
date$year
```

```
## [1] 19
```

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

```
Bday.next <- years(date$year + 1) + Bday
Bday.next
```

```
## [1] "2024-08-25"
```

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

```
days <- as.period( today() %--%Bday.next, unit = "days")
days
```

```
## [1] "145d 0H 0M 0S"
```

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

```
as.period( today() %--%Bday.next )
```

```
## [1] "4m 23d 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?

```
day <- mdy_h("May 8, 2015 3pm", tz = "US/Arizona")
with_tz(day, 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.

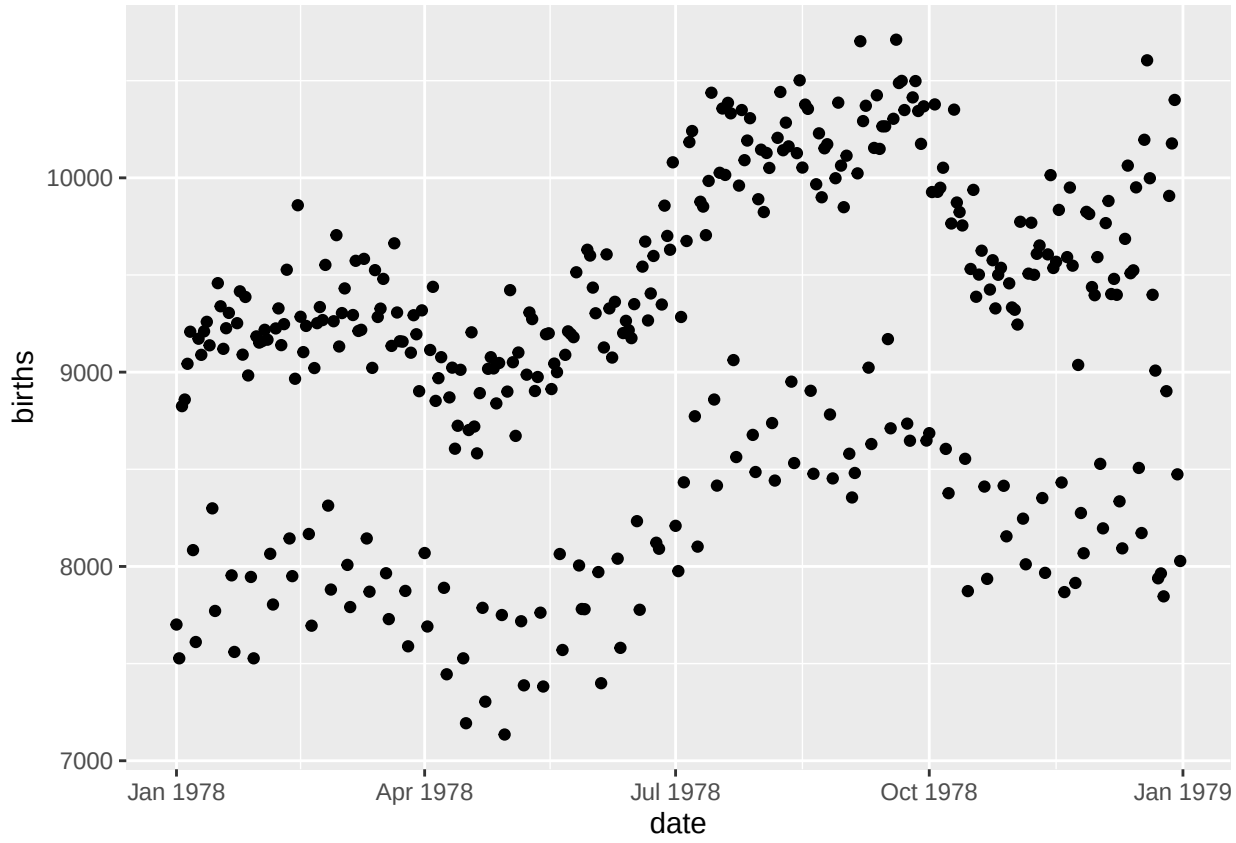
a. Using the `mosaicData` package, load the data set `Births78` which records the number of children born on each day in the United States in 1978. Because this problem is intended to show how to calculate the information using the `date`, remove all the columns *except* `date` and `births`.

```
library(mosaicData)
data(Births78)
birth <- Births78[1:2]
head(birth)
```

```
##         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?

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



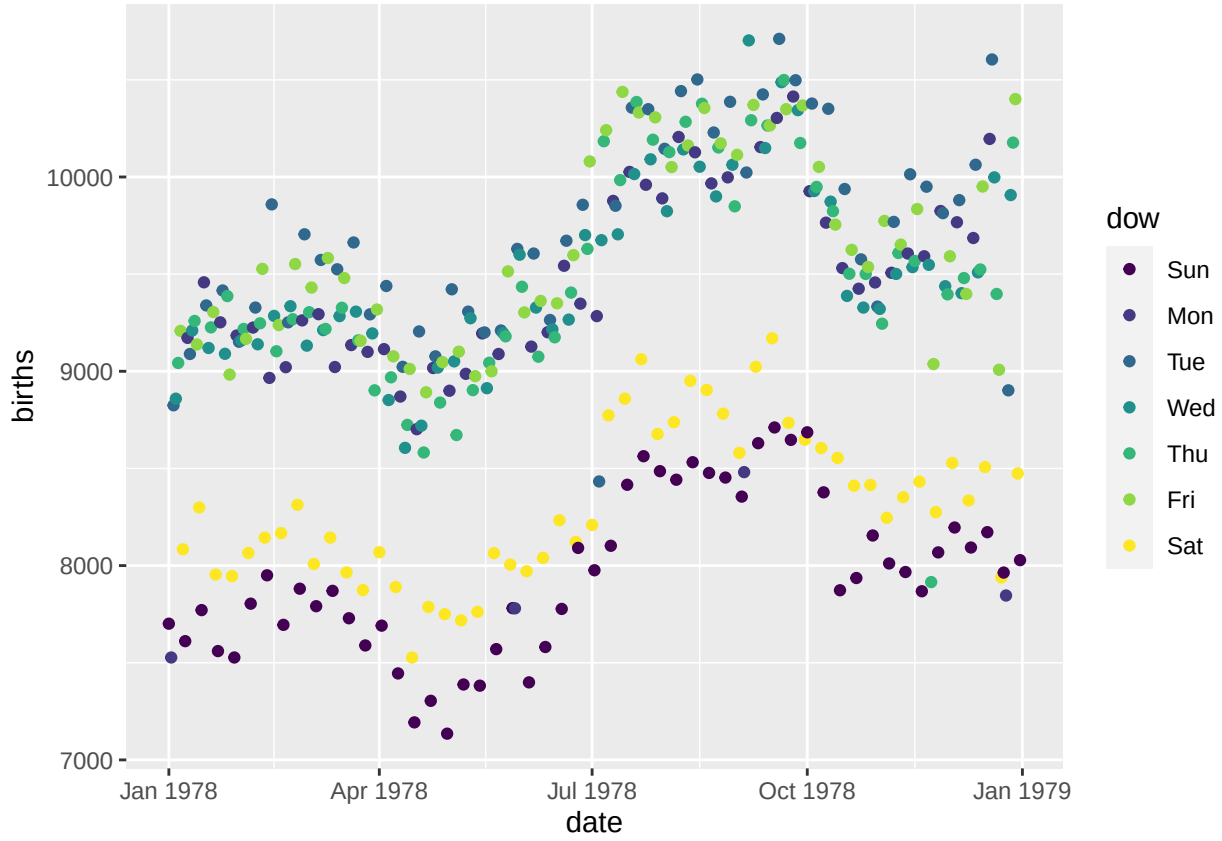
There is a big hump around late july and early october. Also there are two seperate chains of dots that have over a 1k difference in average.

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.

```
birth.dow <- dplyr::mutate(birth, "dow" = lubridate::wday(date, label = TRUE))
```

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

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



We can now see that Saterday and Sunday births are significatly lower.