STA_445_Assignment_6

April Meadows

2024-03-28

```
library(tidyverse)
library(lubridate)
library(ggplot2)
```

Problem 1

Convert the following to date or date/time objects.

a. September 13, 2010.

```
# converts "September 13, 2010" to a date object
mdy("September 13, 2010")
```

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

b. Sept 13, 2010.

```
# converts "Sept 13, 2010" to a date object
mdy("Sept 13, 2010")
```

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

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

```
# month is in wrong format (part d)
```

c. Sep 13, 2010.

```
# converts "Sep 13, 2010" to a date object
mdy("Sep 13, 2010")
```

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

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

```
# converts "s 13, 2010" to a date object
mdy("S 13, 2010")
```

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

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

```
# month is in wrong format - abbreviation needs to be first 3 letters of month name
```

e. 07-Dec-1941.

```
# converts "07-Dec-1941" to a date object
dmy("07-Dec-1941")
```

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

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

```
# converts "1-5-1998" to a date object
dmy("1-5-1998")
```

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

```
# using mdy, this would be converted to 1998-01-05,
#so we need more context for the correct date
```

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

```
# converts "21-5-1998" to a date object
dmy("21-5-1998")
```

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

```
# months only go up to 12, so 21 cannot be confused for a month
```

h. 2020-May-5 10:30 am

```
# converts "2020-May-5 10:30 am" to a date/time object
ymd_hm("2020-May-5 10:30")
```

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

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

```
# converts "2020-May-5 10:30 am" to a date/time object in the PDT timezone
ymd_hm("2020-May-5 10:30",tz="US/Pacific")
```

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

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

```
# converts "2020-May-5 10:30 am" to a date/time object in the AST timezone
ymd_hm("2020-May-5 10:30",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:

```
# creates a date object of my birthday
bday = make_date(year=1998, month=08, day=21)
format(bday,"I was born on %B %d, %Y.")
```

```
## [1] "I was born on August 21, 1998."
```

a. Calculate the date of your 64th birthday.

```
# adds 64 years to bday to get the date of my 64th birthday
old <- bday + dyears(64)
format(old,"My 64th birthday is %B %d, %Y.")
```

```
## [1] "My 64th birthday is August 21, 2062."
```

b. Calculate your current age (in years).

```
# calculates the year of the time interval between bday and now
age <- year(as.period(bday %--% now()))
sprintf("My current age is %i.",age)
```

```
## [1] "My current age is 25."
```

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

```
# calculates my next bday by adding my current age + 1
nextbday <- bday + years(age+1)
format(nextbday,"My next birthday is %B %d, %Y.")
```

```
## [1] "My next birthday is August 21, 2024."
```

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

```
# returns the number of days between now and my next bday
d <- day(as.period(now() %--% nextbday, unit="days"))
sprintf("There are %i days until my next birthday.",d)
```

```
## [1] "There are 145 days until my next birthday."
```

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

```
# prints the number of months and days between now and my next bday
m <- month(as.period(now() %--% nextbday, unit="months"))
d <- day(as.period(now() %--% nextbday, unit="months"))
sprintf("There are %i months and %i days until my next birthday.",m,d)
```

```
## [1] "There are 4 months and 23 days until my next birthday."
```

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?

```
# creates a datetime object for the time of the phone call
phonecall <- make_datetime(year=2015,month=05,day=08,hour=15,min=0,sec=0,tz="US/Arizona")

# converts the date and time of the phone call to New Zealand time
nz <- with_tz(phonecall,tz="NZ")

# prints out the new time
format(nz,"The phone call is on %B %d, %Y at %I %P %Z.")
```

```
## [1] "The phone call is on May 09, 2015 at 10 am 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`.

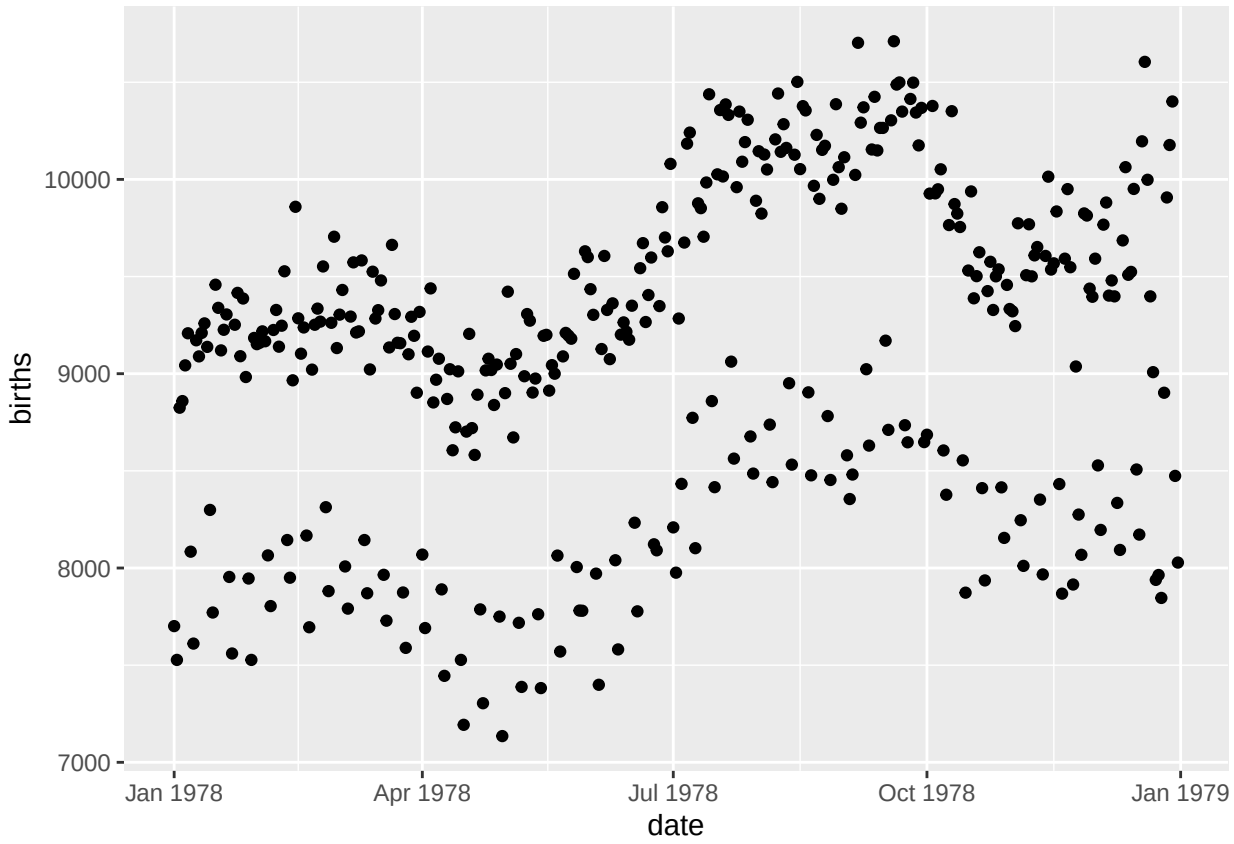
```
# install.packages("mosaicData")

# imports the Births78 data from the mosaicData package
data("Births78",package = "mosaicData")

# removes the unnecessary columns from the data
Births78 <- Births78[-c(3:8)]
```

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?

```
# plots day of the year vs births
Births78 %>%
ggplot(aes(x=date,y=births)) +
  geom_point()
```



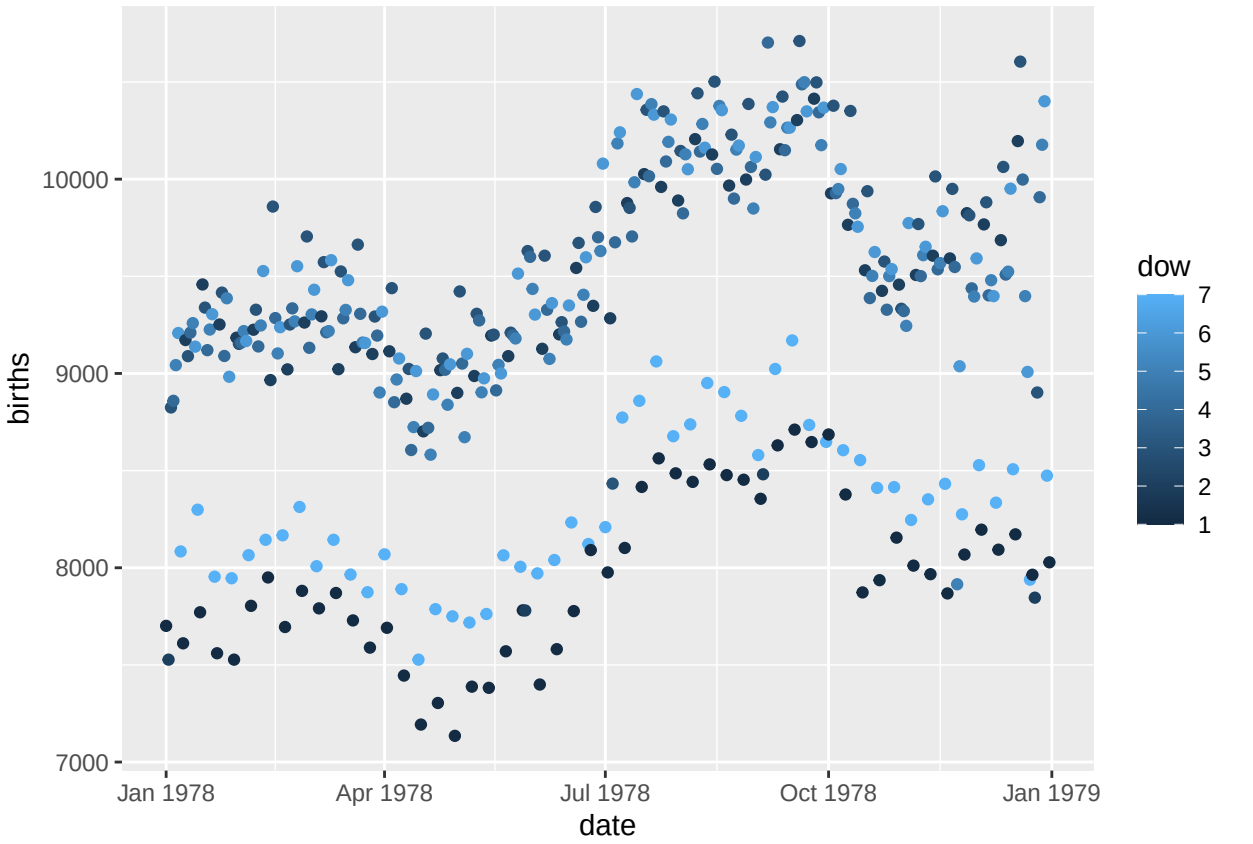
There are two trend lines with similar shape. The line with lower births contains fewer data points than the higher births line. This could mean that there was a type of day where fewer births happened consistantly.

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.

```
# adds dow column to Births78
Births78 <- Births78 %>%
              mutate(dow = wday(date))
```

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

```
# plots date vs births along with point color based on day of the week
Births78 %>%
ggplot(aes(x=date,y=births)) +
  geom_point(aes(color=dow))
```



It appears that the fewest number of births occurred on weekends (dow = 1 and 7). My guess as to why this happened is because fewer c-sections and inductions were scheduled on weekends, leading to less births on those days.