

# assignment5

Noah Plant

2024-10-28

## Exercise 1

Convert the following to date or date/time objects.

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v readr      2.1.5
## v forcats    1.0.0      v stringr   1.5.1
## v ggplot2    3.5.1      v tibble    3.2.1
## v lubridate  1.9.3      v tidyr     1.3.1
## v purrr      1.0.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(lubridate)
library(mosaicData)
```

a) September 13, 2010.

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

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

b) Sept 13, 2010.

```
as.Date("Sept 13, 2010",format='%b %d, %Y')
```

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

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

```
## 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
```

The date must be abbreviated to 3 letters, spelled out completely or represented by a 2 digit number. There is no way to format the month as a single letter as that would cause confusion I imagine.

e) 07-Dec-1941.

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

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

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

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

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

```
dmy("1-5-1998")
```

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

I may be wrong in my date making because I have no way of knowing whether the date is supplied in day month year or year month day.

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

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

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

I know I am right because if I try to parse the data in any other way I get an error. Also there is no way this date can be written in mdy because there are only 12 months in a year.

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')
```

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

```
ymd_hm('2020-May-5 10:30 am PDT',tz="America/Los_Angeles")
```

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

The time zone does not parse unless I include the tz="PDT" part of the code. Even then if I do that it does not register. I have to include the Olson name which is America/Los\_Angeles

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"
```

### Exercise 3

Suppose you have arranged for a phone call to be at 3 pm on May 8, 2025 at Arizona time. However, the recipient will be in Auckland, NZ. What time will it be there?

```
arizona<-mdy_hm('May 8, 2025 3:00 pm',tz='US/Arizona')
```

```
auckland<-with_tz(arizona,tz="Pacific/Auckland")
```

```
auckland
```

```
## [1] "2025-05-09 10:00:00 NZST"
```

### Exercise 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.

```
data<-mosaicData::Births78
```

```
data.2<-select(data,date,births)
```

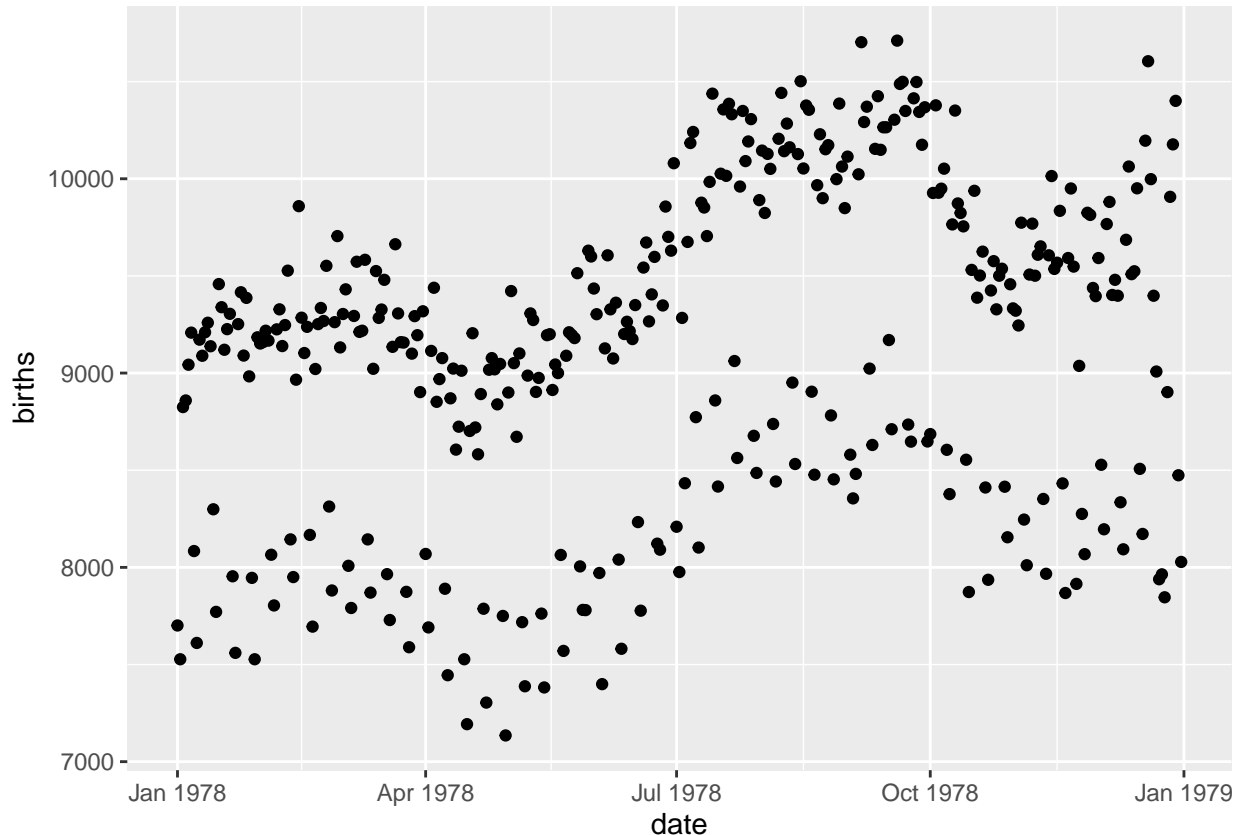
```
head(data.2)
```

```
##           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?

```
library(ggplot2)

ggplot(data=data.2, aes(x=date,y=births) )+
  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.

```
data.2 <- data.2 %>% mutate(dow=(wday(date,label=TRUE, abbr=FALSE)))

head(data.2)
```

##	date	births	dow
## 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 with the point color being determined by the day of the week variable.

```
ggplot(data=data.2, aes(x=date,y=births,color=dow) )+
  geom_point()
```

