Date and Time

Justin Baumann

Table of contents

1 Dealing with Date and Time in R

Date and time are often important variables in scientific data analysis. We are often interested in change over time and we also often do time series sampling. Learning how to manage dates and times in R is essential! Luckily, there is a user friendly and tidyverse friendly package that can help us with dates, times, and datetimes. That package is called 'lubridate' and we will learn all about it below.

First, we need to load packages

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

2 Date and Time in R

R and really all programming languages have a difficult time with dates and times. Luckily, programmers have developed ways to get computer to understand dates and times as time series (so we can plot them on a graph axis and do analysis, for example).

There are several common formats of date and time that we don't need to get into, but for many tools we use in the field we have a timestamp that includes day, month, year, and time (hours, minutes, and maybe seconds). When all of that info ends up in 1 column of a .csv it can be annoying and difficult to get R to understand what that column means. There are tons of ways to solve this problem but the easiest is definitely to just use some simple functions in the Lubridate package!

2.1 Read in some data to practice with

```
dat<-read.csv('https://raw.githubusercontent.com/jbaumann3/Intro-to-R-for-Ecology/main/fin
head(dat) #take a look at the data to see how it is formatted
```

```
  X                date probe_name probe_type value
1 1 07/01/2021 00:00:00      B2_T2       Temp 18.10
2 2 07/01/2021 00:00:00     B2_pH2         pH  4.53
3 3 07/01/2021 00:00:00     B1_pH2         pH  8.12
4 4 07/01/2021 00:00:00      B1_T2       Temp 17.70
5 5 07/01/2021 00:00:00      B1_T1       Temp 17.70
6 6 07/01/2021 00:00:00     B1_pH1         pH  8.12
```

```
str(dat) #what are the attributes of each column (NOTE the attirbutes of the date column -
```

```
'data.frame':   47200 obs. of  5 variables:
 $ X         : int  1 2 3 4 5 6 7 8 9 10 ...
 $ date      : chr  "07/01/2021 00:00:00" "07/01/2021 00:00:00" "07/01/2021 00:00:00" "07/01/
 $ probe_name: chr  "B2_T2" "B2_pH2" "B1_pH2" "B1_T2" ...
 $ probe_type: chr  "Temp" "pH" "pH" "Temp" ...
 $ value     : num  18.1 4.53 8.12 17.7 17.7 8.12 19.7 7.99 18.1 4.53 ...
```

2.2 Change date column (factor) to date/time format

To do this we just need to recognize the order of or date/time. For example, we might have year, month, day, hours, minutes OR day, month, year, hours, minutes in order from left to right.

In this case we have: 07/01/2021 00:00:00 or month/day/year hours:minutes:seconds. We care about the order of these. So to simply, we have mdy_hms Lubridate has functions for all

combinations of these formats. So, mdy_hms() is one. You may also have ymd_hm() or any other combo. You just enter your date info followed by an underscore and then your time info. Here's how you apply this!

```
  str(dat)
```

```
'data.frame':   47200 obs. of  5 variables:
 $ X         : int  1 2 3 4 5 6 7 8 9 10 ...
 $ date      : chr  "07/01/2021 00:00:00" "07/01/2021 00:00:00" "07/01/2021 00:00:00" "07/01/
 $ probe_name: chr  "B2_T2" "B2_pH2" "B1_pH2" "B1_T2" ...
 $ probe_type: chr  "Temp" "pH" "pH" "Temp" ...
 $ value     : num  18.1 4.53 8.12 17.7 17.7 8.12 19.7 7.99 18.1 4.53 ...
```

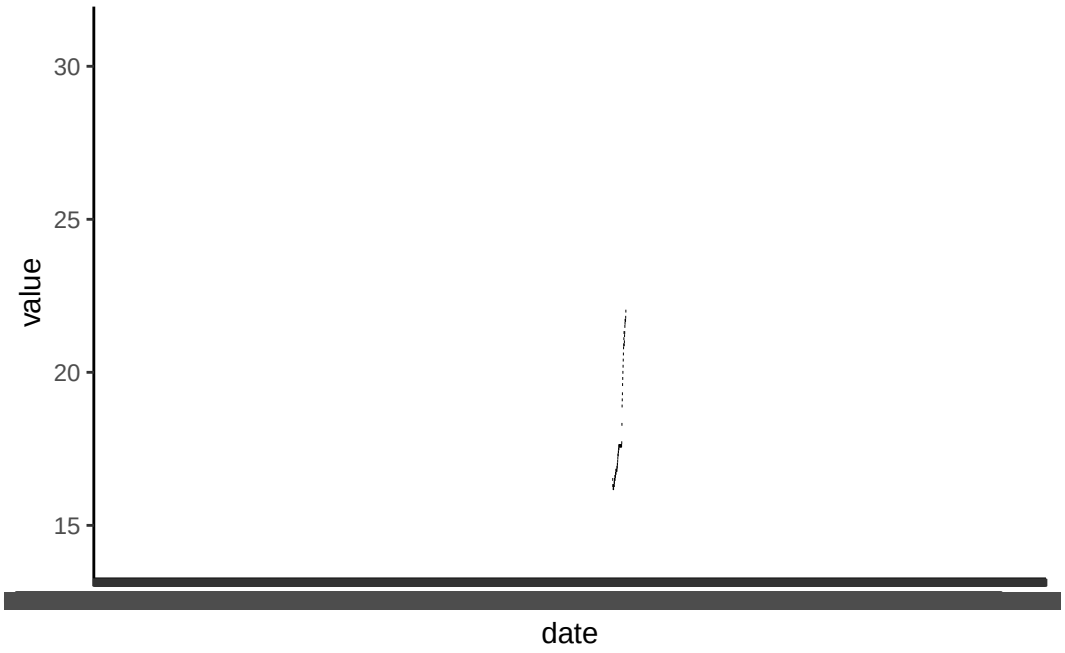
```
  dat$date<-mdy_hms(dat$date) #converts our date column into a date/time object based on the

  str(dat)# date is no longer a factor but is now a POSIXct object, which means it is in dat
```

```
'data.frame':   47200 obs. of  5 variables:
 $ X         : int  1 2 3 4 5 6 7 8 9 10 ...
 $ date      : POSIXct, format: "2021-07-01 00:00:00" "2021-07-01 00:00:00" ...
 $ probe_name: chr  "B2_T2" "B2_pH2" "B1_pH2" "B1_T2" ...
 $ probe_type: chr  "Temp" "pH" "pH" "Temp" ...
 $ value     : num  18.1 4.53 8.12 17.7 17.7 8.12 19.7 7.99 18.1 4.53 ...
```

2.3 Why this matters

Here we have two example graphs that show why dates are annoying and how using lubridate helps us!

A graph using the raw data alone (not changing date to a date/time object)



same graph after making date into a date/time object