HOBO Logger Data for Sceloporus Humidity Acclimation Study, 2021

Savannah Weaver

Contents

Packages

```
if (!require("tidyverse")) install.packages("tidyverse")
library("tidyverse")
```

Load in Data

This data was collected using Onset HOBO temperature and humidity dataloggers during the course of our experiment. See ___ for full details.

The data is in a separate file for each download for each logger, so I need to compile each of those into one dataset.

To do this, first I compile a list of the filenames I need to read-in.

```
# make a list of file names of all data to load in
filenames <- list.files(path = "data/HOBOs")
```

Next, I make a function that will read in the data from each csv, name and organize the data correctly.

```
# make a function to read in data from each csv file and add correct identifiers
read_HOBO_files <- function(filename) {

  # edit the filename inputted to funtion
  # to make a unique identifier for each logger
  name <- substr(filename, 1, nchar(filename)-15)

  # also make an identifier for the site
  site <- substr(filename, 1, nchar(filename)-15)

  # read in the csv file for this given filename
  dat <- read.csv(file.path("data/HOBOs", filename),
                  # each csv has headers
                  header = TRUE,
                  # this is what I want to rename the col headers
                  col.names = c("order", "date_time_PST", "temp_C",
                                "relative_humidity", "dew_pt_C",
                                # the 6,7,8th cols are not data
                                # logger use info we don't need
                                "mostly_blank", "mostly_blank", "mostly_blank")
                  ) %>%
    # select only the cols with data we want
    # don't need order- just an arbitrary observation identifier
    # don't need "mostly_blank" cols- unnecessary logger use info
    # but get the rest of the cols with informative data
    dplyr::select(date_time_PST, temp_C, relative_humidity, dew_pt_C) %>%
    # add a column with the name of the HOBO the data is from
    dplyr::mutate(HOBO_ID = name)

  # return the dataframe for that single csv file
  dat
}
```

Finally, I apply the function I made to all of the filenames I compiled, then put all of those dataframes into one dataframe for my analyses.

This will print warnings saying that header and col.names are different lengths, because the data has extra notes on logger usage that we read-in, but get rid of.

```
# apply function to get data from all csvs
all_HOBO_data <- lapply(filenames, read_HOBO_files) %>%
  # paste all data files together into one df by row
  reduce(rbind)
```

```
## Warning in read.table(file = file, header = header, sep = sep, quote = quote, :
## header and 'col.names' are of different lengths

## Warning in read.table(file = file, header = header, sep = sep, quote = quote, :
## header and 'col.names' are of different lengths

## Warning in read.table(file = file, header = header, sep = sep, quote = quote, :
## header and 'col.names' are of different lengths

## Warning in read.table(file = file, header = header, sep = sep, quote = quote, :
```

```
## header and 'col.names' are of different lengths

## Warning in read.table(file = file, header = header, sep = sep, quote = quote, :
## header and 'col.names' are of different lengths

## Warning in read.table(file = file, header = header, sep = sep, quote = quote, :
## header and 'col.names' are of different lengths

## Warning in read.table(file = file, header = header, sep = sep, quote = quote, :
## header and 'col.names' are of different lengths

## Warning in read.table(file = file, header = header, sep = sep, quote = quote, :
## header and 'col.names' are of different lengths

## Warning in read.table(file = file, header = header, sep = sep, quote = quote, :
## header and 'col.names' are of different lengths

## Warning in read.table(file = file, header = header, sep = sep, quote = quote, :
## header and 'col.names' are of different lengths

## Warning in read.table(file = file, header = header, sep = sep, quote = quote, :
## header and 'col.names' are of different lengths
```

Data Wrangling

Now that I have a full dataframe, I want to make sure all the data is there and correct.

```r
str(all_HOBO_data)
```

```
## 'data.frame':    17035 obs. of  5 variables:
##  $ date_time_PST    : chr  "04/14/2021 12:46:33" "04/14/2021 13:16:33" "04/14/2021 13:46:33" "04/14/2
##  $ temp_C           : num  23 22.6 23.7 31.1 34.3 ...
##  $ relative_humidity: num  38.53 37.81 37.23 11.18 7.39 ...
##  $ dew_pt_C         : num  8.16 7.53 8.29 -2.61 -5.68 ...
##  $ HOBO_ID          : chr  "Dry-Cool-1 2021-05-18 11_02_00 PST" "Dry-Cool-1 2021-05-18 11_02_00 PST"
```

```r
summary(all_HOBO_data)
```

```
##  date_time_PST          temp_C      relative_humidity    dew_pt_C
##  Length:17035       Min.   :-2.16   Min.   :  1.69    Min.   :-28.140
##  Class :character   1st Qu.:23.80   1st Qu.: 14.40    1st Qu.: -2.130
##  Mode  :character   Median :24.93   Median : 43.16    Median : 10.960
##                     Mean   :26.87   Mean   : 44.25    Mean   :  9.731
##                     3rd Qu.:28.17   3rd Qu.: 69.99    3rd Qu.: 20.700
##                     Max.   :41.39   Max.   :100.00    Max.   : 41.250
##                     NA's   :18      NA's   :18        NA's   :18
##    HOBO_ID
##  Length:17035
##  Class :character
##  Mode  :character
##
##
##
##
```

I need to subset the data based on a date range to extract data for actual experiment days, since the loggers run continuously. There are also NA's which are an artifact of the logger's notes.

```
exp_dates <- seq(as.Date("2021-04-19"), as.Date("2021-05-18"), "days")

subset_HOBO_data <- all_HOBO_data %>%
  dplyr::filter(complete.cases(temp_C, relative_humidity)) %>%
  mutate(date_only = as.Date(date_time_PST, format = "%m/%d/%Y")) %>%
  dplyr::filter(date_only %in% exp_dates)
```

Next, format the data:

```
format_HOBO_data <- subset_HOBO_data %>%
  mutate(date_time_PST = as.POSIXct(date_time_PST,
                                    format = "%m/%d/%Y %H:%M:%S"),
         HOBO_ID = (substr(HOBO_ID, 1, 10)), # select first 10 characters
         HOBO_ID = str_trim(HOBO_ID), # remove trailing white space
         HOBO_ID = str_replace_all(HOBO_ID, "[^A-z0-9]", "_"), # replace any special characters with un
         HOBO_ID = as.factor(HOBO_ID) # set class as factor
         )
summary(format_HOBO_data)
```

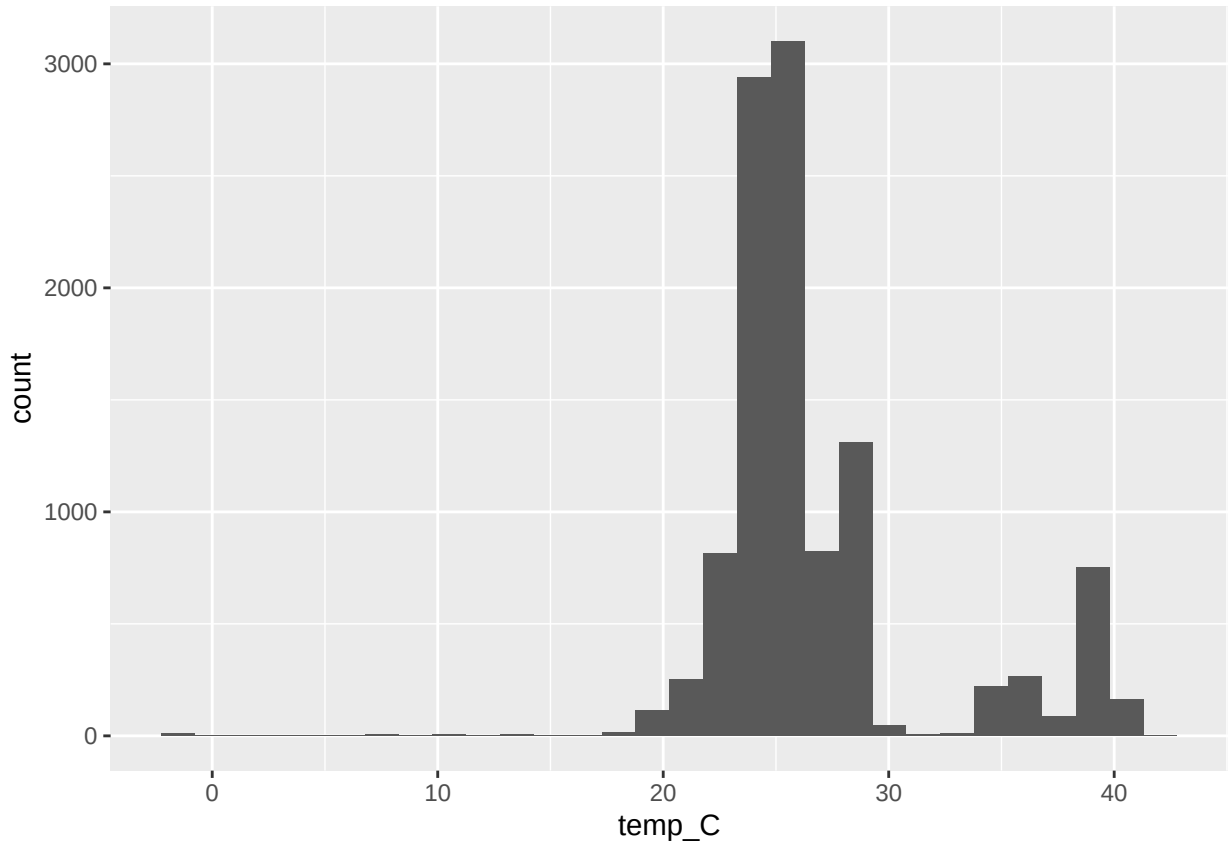
```
##  date_time_PST                    temp_C      relative_humidity
##  Min.   :2021-04-19 00:11:45   Min.   :-2.16   Min.   :  2.93
##  1st Qu.:2021-04-26 01:16:57   1st Qu.:24.01   1st Qu.: 12.71
##  Median :2021-05-03 12:16:33   Median :25.31   Median : 40.59
##  Mean   :2021-05-03 14:57:24   Mean   :26.81   Mean   : 45.75
##  3rd Qu.:2021-05-11 03:13:42   3rd Qu.:28.14   3rd Qu.: 80.95
##  Max.   :2021-05-18 23:54:56   Max.   :41.39   Max.   :100.00
##     dew_pt_C             HOBO_ID       date_only
##  Min.   :-22.510   Dry_Cool_1:2854   Min.   :2021-04-19
##  1st Qu.: -4.120   Dry_Cool_2:1440   1st Qu.:2021-04-26
##  Median : 10.910   Dry_Hot_1 :2854   Median :2021-05-03
##  Mean   :  9.328   Dry_Hot_2 : 934   Mean   :2021-05-03
##  3rd Qu.: 22.300   Humid_Cool:1414   3rd Qu.:2021-05-11
##  Max.   : 41.250   Humid_Hot :1474   Max.   :2021-05-18
```

I'm also going to omit the data for the hot treatments that failed, since we will not be presenting that. Also, when a couple of lizards died, their loggers were moved, not always to the same treatment... So I need to look for those changes. *Finally, the time of day I started the experiment on capture day and ended the experiment varied, and we also took them out of treatment for checkups. The loggers should reflect this, so I need to find those values and exclude them.* We are not removing points arbitrarily. I know that any abrupt differences between points reflects time the loggers were not in the environmental chambers, which we unfortunately did not keep track of.

Check temperature values:

```
format_HOBO_data %>%
  ggplot(data = ., aes(x = temp_C)) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
format_HOBO_data %>%
  group_by(date_only, HOBO_ID) %>%
  summarise(mean_temp = mean(temp_C)) %>%
  arrange(mean_temp, date_only)
```

```
## `summarise()` regrouping output by 'date_only' (override with `.groups` argument)
```

```
## # A tibble: 180 x 3
## # Groups:   date_only [30]
##    date_only  HOBO_ID    mean_temp
##    <date>     <fct>          <dbl>
##  1 2021-05-10 Dry_Cool_1      20.2
##  2 2021-05-06 Dry_Cool_2      22.1
##  3 2021-05-03 Dry_Cool_2      22.2
##  4 2021-05-09 Humid_Hot       22.6
##  5 2021-05-08 Humid_Hot       22.7
##  6 2021-05-17 Dry_Hot_2       22.8
##  7 2021-05-03 Dry_Hot_1       22.9
##  8 2021-05-03 Dry_Hot_2       22.9
##  9 2021-05-04 Humid_Hot       23.0
## 10 2021-05-07 Dry_Cool_2      23.1
## # ... with 170 more rows
```

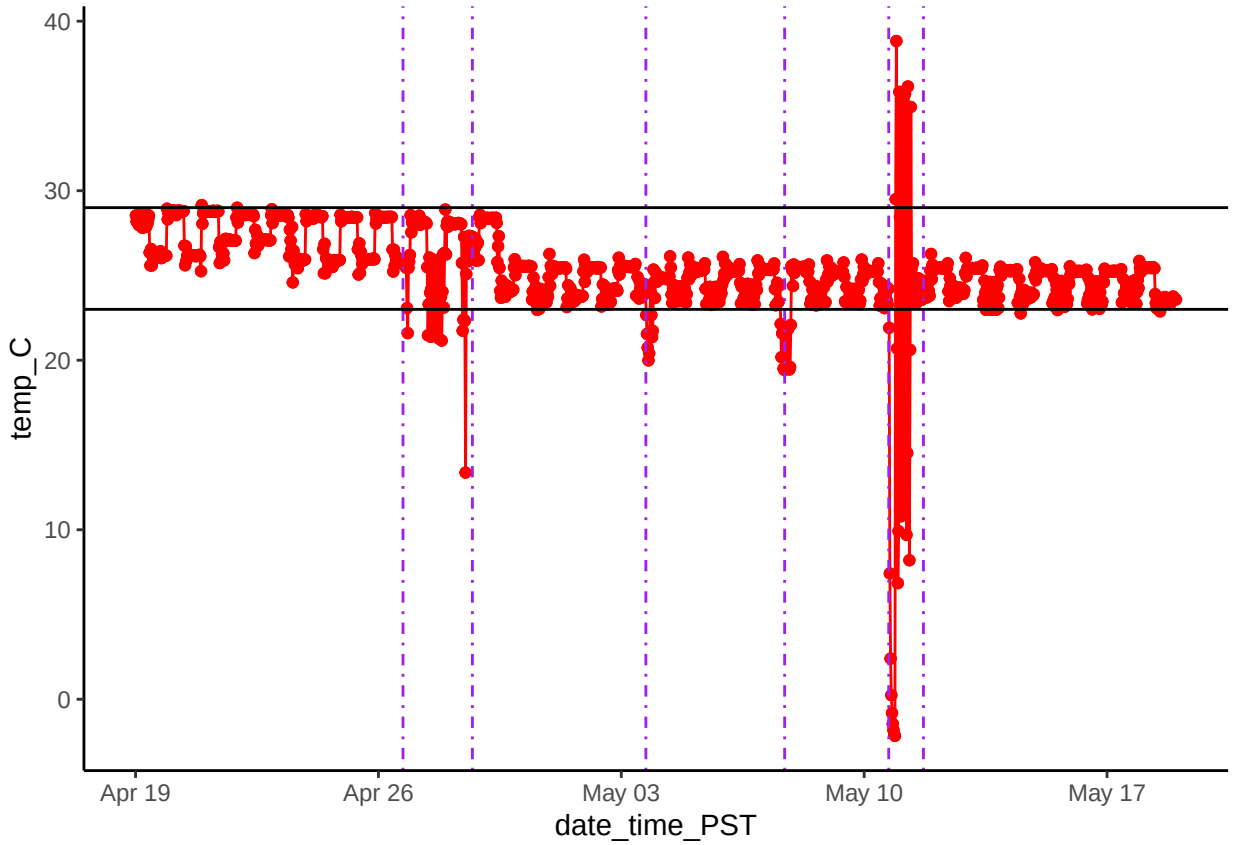
Data Cleaning

set the not in function:

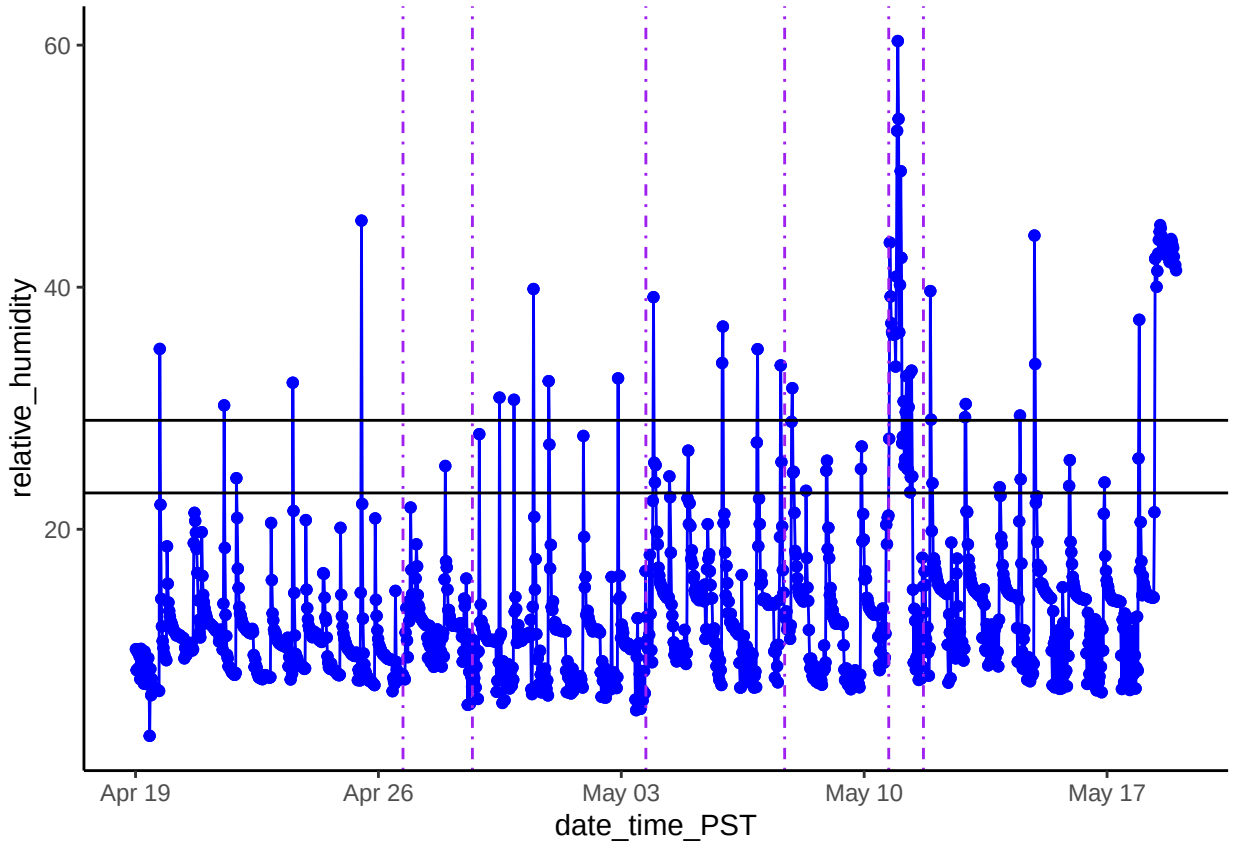
```
`%nin%` = Negate(`%in%`)
```

HOBO "dry cool 1":

```
# temperature
format_HOBO_data %>%
  dplyr::filter(HOBO_ID == "Dry_Cool_1") %>%
  ggplot(data = ., aes(x = date_time_PST,
                       y = temp_C,
                       color = HOBO_ID
                       )) +
  geom_point(color = "red") +
  geom_line(color = "red") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-04-27")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-04-29")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-04")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-08")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-11")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-12")),
             linetype = 4, color = "purple") +
  geom_hline(yintercept = 23) +
  geom_hline(yintercept = 29) +
  theme_classic()
```




```
# humidity
format_HOBO_data %>%
  dplyr::filter(HOBO_ID == "Dry_Cool_1") %>%
  ggplot(data = ., aes(x = date_time_PST,
                       y = relative_humidity,
                       color = HOBO_ID
                       )) +
  geom_point(color = "blue") +
  geom_line(color = "blue") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-04-27")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-04-29")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-04")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-08")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-11")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-12")),
             linetype = 4, color = "purple") +
  geom_hline(yintercept = 23) +
  geom_hline(yintercept = 29) +
  theme_classic()
```



There is room temperature/wacky temperature data between April 27-29, and on May 4, 8, 11, and 12. The humidity data only looks wrong on May 11 & 12.

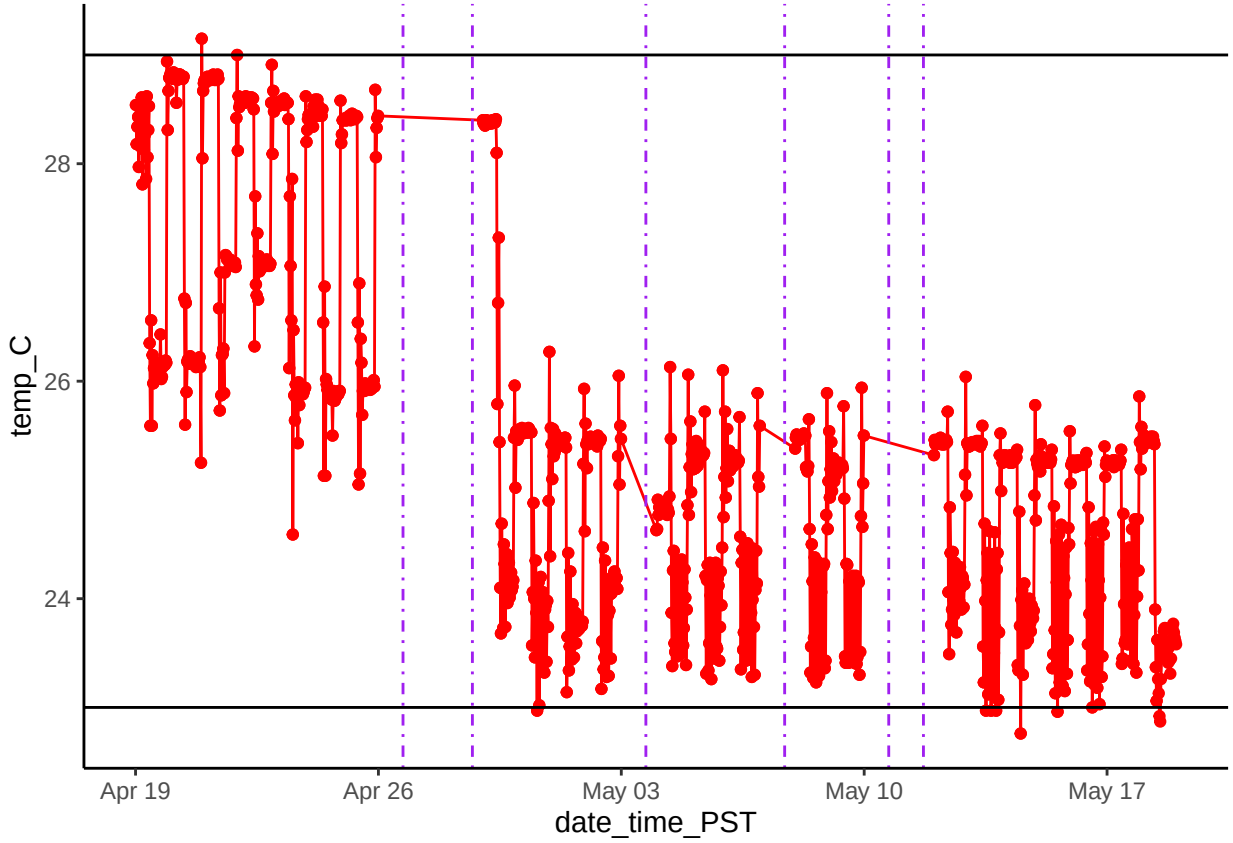
Try excluding. For some reason the dates plotted versus filtered line up differently, which is why I filter out the date before what I plotted.

```
unwanted_dates_dc1 <- as.Date(c("2021-04-26", "2021-04-27",
                        "2021-04-28",
                        "2021-05-03", "2021-05-07",
                        "2021-05-10", "2021-05-11"),
                        format = "%Y-%m-%d")
dc1_subset <- format_HOBO_data %>%
  dplyr::filter(HOBO_ID == "Dry_Cool_1") %>%
  dplyr::filter(date_only %nin% unwanted_dates_dc1)
```

plot again to show that we removed erroneous data:

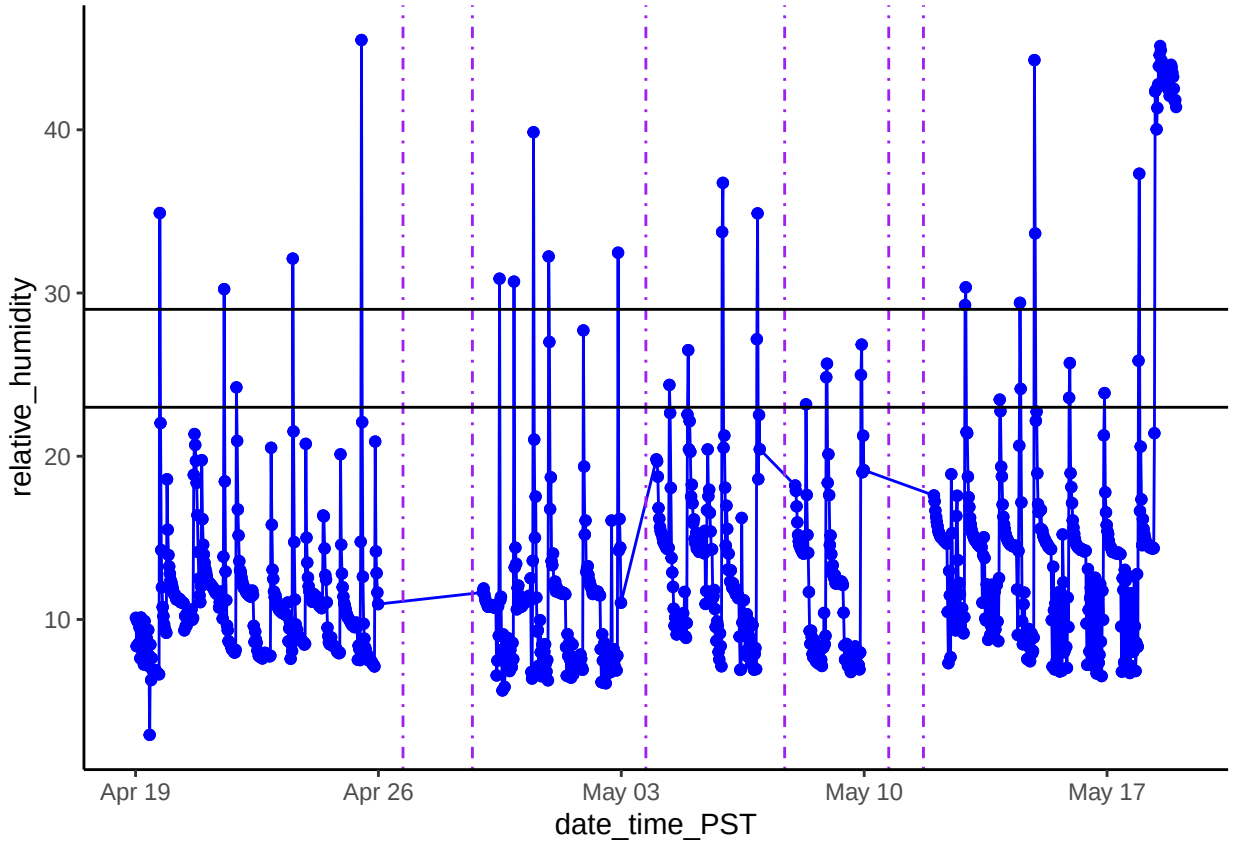
```
# temperature
dc1_subset %>%
  ggplot(data = ., aes(x = date_time_PST,
                       y = temp_C,
                       color = HOBO_ID
                       )) +
  geom_point(color = "red") +
  geom_line(color = "red") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-04-27")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-04-29")),
             linetype = 4, color = "purple") +
```

```
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-04")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-08")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-11")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-12")),
             linetype = 4, color = "purple") +
  geom_hline(yintercept = 23) +
  geom_hline(yintercept = 29) +
  theme_classic()
```




```
# humdity
dc1_subset %>%
  ggplot(data = ., aes(x = date_time_PST,
                       y = relative_humidity,
                       color = HOBO_ID
                       )) +
  geom_point(color = "blue") +
  geom_line(color = "blue") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-04-27")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-04-29")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-04")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-08")),
```

```
            linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-11")),
            linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-12")),
            linetype = 4, color = "purple") +
  geom_hline(yintercept = 23) +
  geom_hline(yintercept = 29) +
  theme_classic()
```

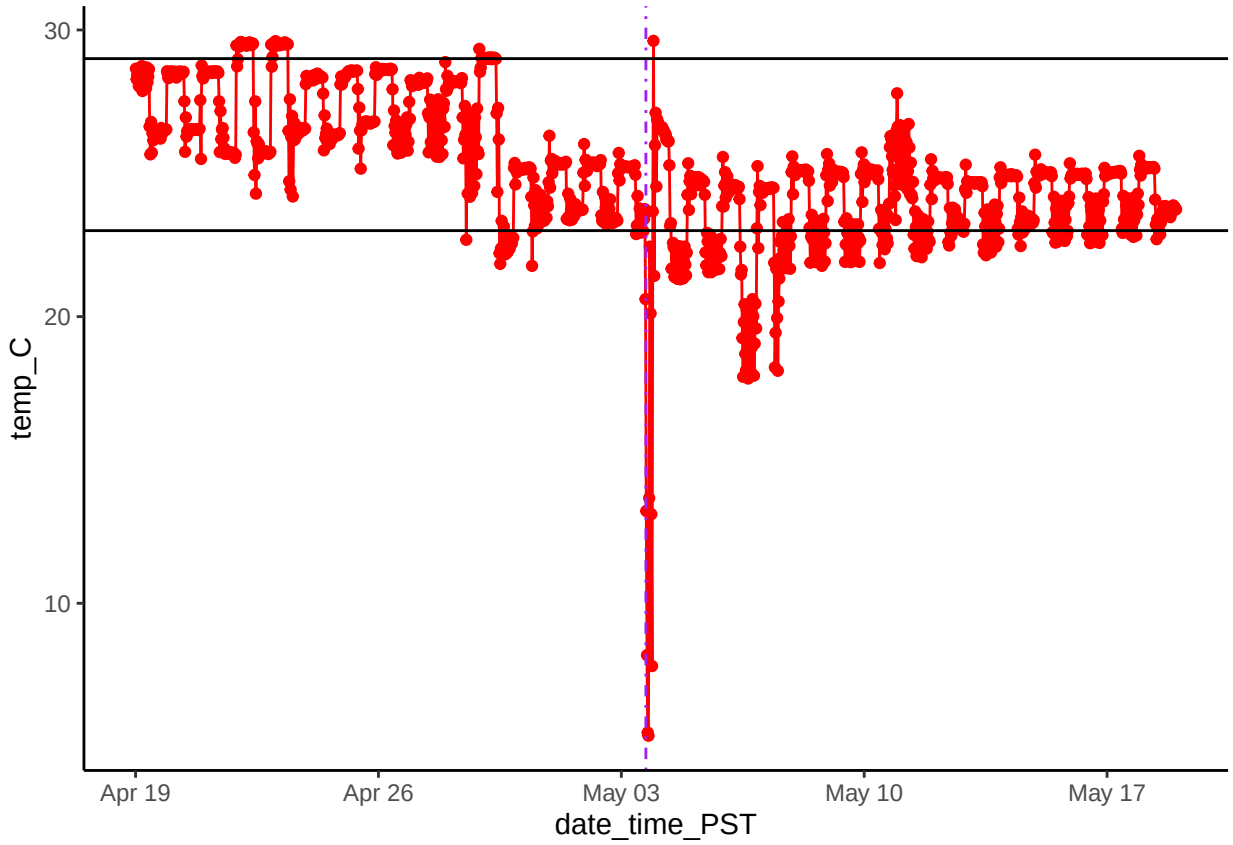


And, we confirmed that HOBO "dry cool 1" was indeed in the dry and cool treatment.

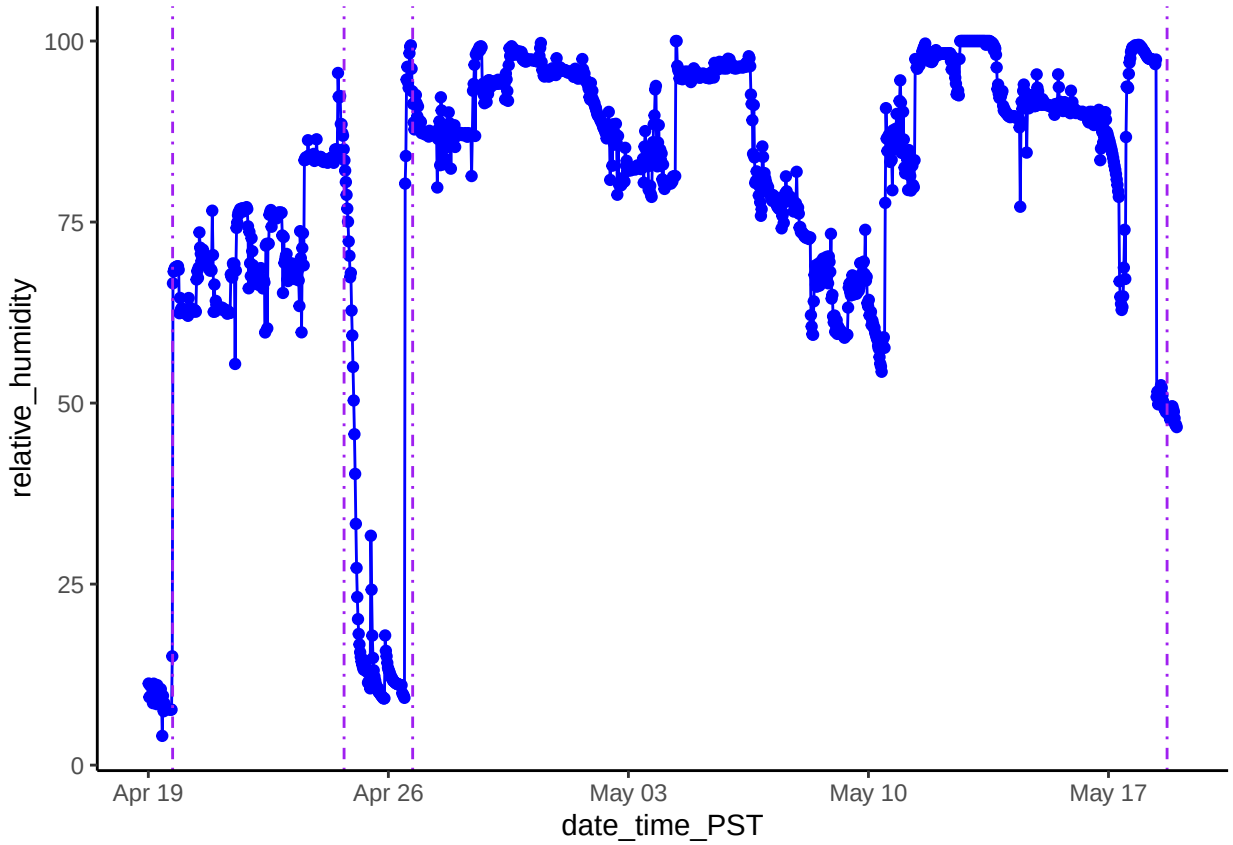
HOBO "dry cool 2":

```
# temperature
format_HOBO_data %>%
  dplyr::filter(HOBO_ID == "Dry_Cool_2") %>%
  ggplot(data = ., aes(x = date_time_PST,
                       y = temp_C,
                       color = HOBO_ID
                       )) +
  geom_point(color = "red") +
  geom_line(color = "red") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-04")),
            linetype = 4, color = "purple") +
  geom_hline(yintercept = 23) +
  geom_hline(yintercept = 29) +
```

```
  theme_classic()
```




```
# humidity
format_HOBO_data %>%
  dplyr::filter(HOBO_ID == "Dry_Cool_2") %>%
  ggplot(data = ., aes(x = date_time_PST,
                       y = relative_humidity,
                       color = HOBO_ID
                       )) +
  geom_point(color = "blue") +
  geom_line(color = "blue") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-04-20")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-04-25")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-04-27")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-19")),
             linetype = 4, color = "purple") +
  theme_classic()
```



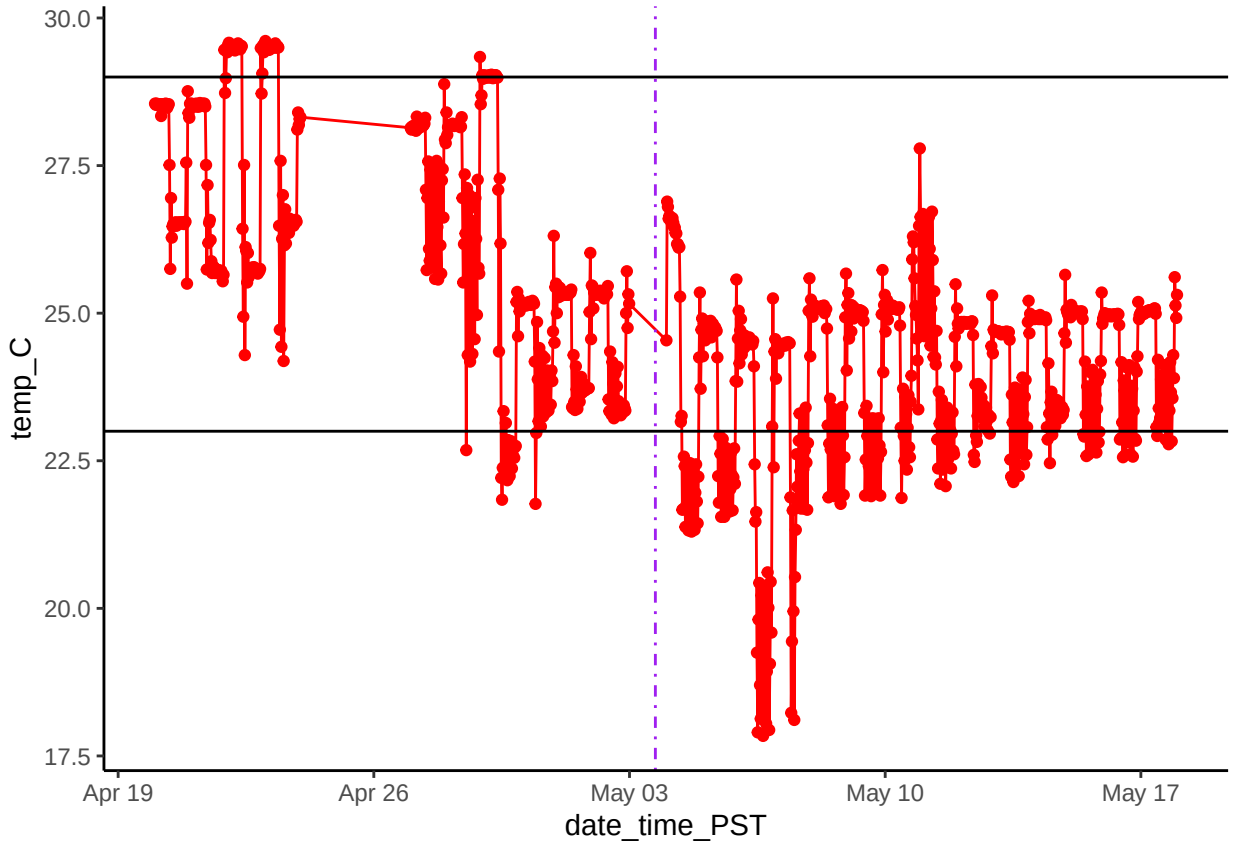
Temperature is wrong on May 4. The humidity is wrong April 19-20, April 25-27, and on May 19.

Try excluding. For some reason the dates plotted versus filtered line up differently, which is why I filter out the date before what I plotted.

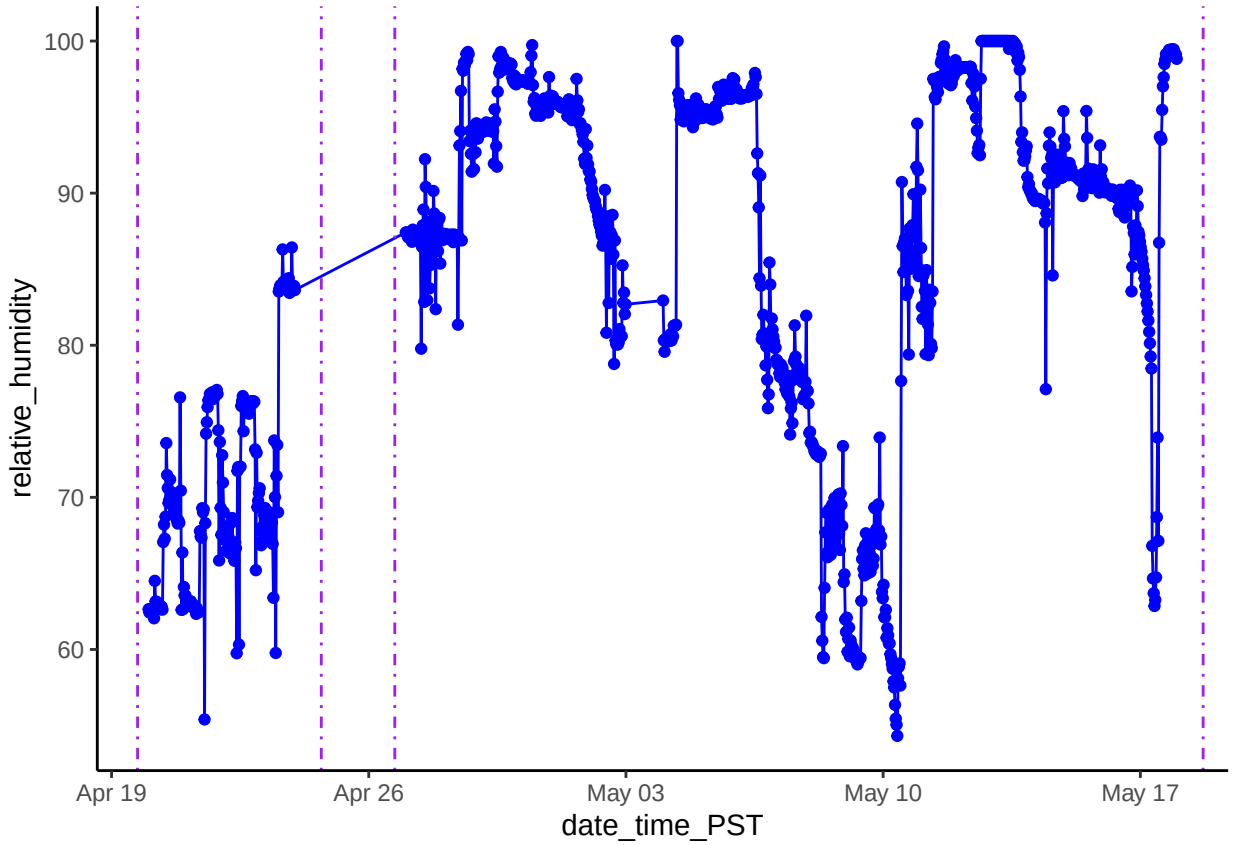
```
unwanted_dates_dc2 <- as.Date(c("2021-04-18", "2021-04-19",
                          "2021-04-24", "2021-04-25", "2021-04-26",
                          "2021-05-03", "2021-05-18"),
                          format = "%Y-%m-%d")
dc2_subset <- format_HOBO_data %>%
  dplyr::filter(HOBO_ID == "Dry_Cool_2") %>%
  dplyr::filter(date_only %nin% unwanted_dates_dc2)
```

plot again to show that we removed erroneous data:

```
# temperature
dc2_subset %>%
  ggplot(data = ., aes(x = date_time_PST,
                       y = temp_C,
                       color = HOBO_ID
                       )) +
  geom_point(color = "red") +
  geom_line(color = "red") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-04")),
             linetype = 4, color = "purple") +
  geom_hline(yintercept = 23) +
  geom_hline(yintercept = 29) +
  theme_classic()
```



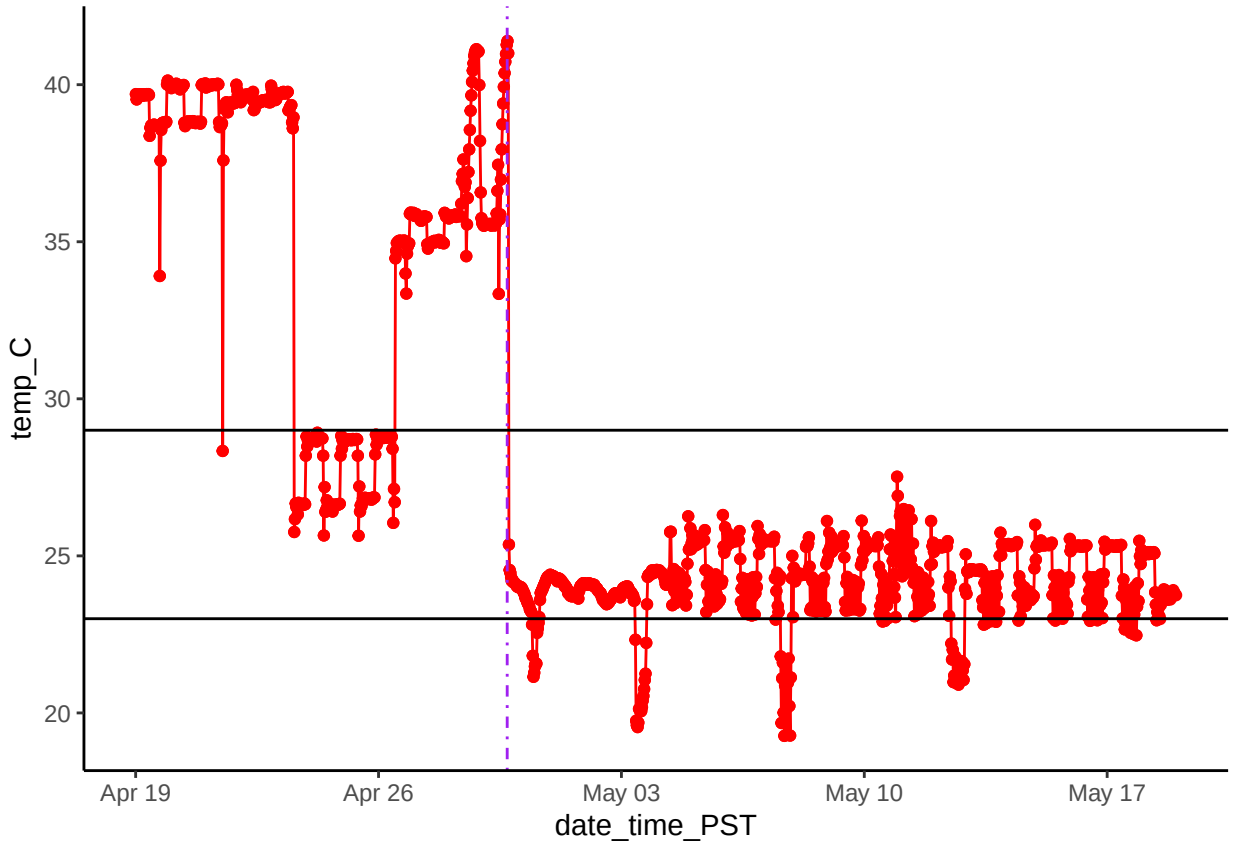

```
# humdity
dc2_subset %>%
  ggplot(data = ., aes(x = date_time_PST,
                       y = relative_humidity,
                       color = HOBO_ID
                       )) +
  geom_point(color = "blue") +
  geom_line(color = "blue") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-04-20")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-04-25")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-04-27")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-19")),
             linetype = 4, color = "purple") +
  theme_classic()
```



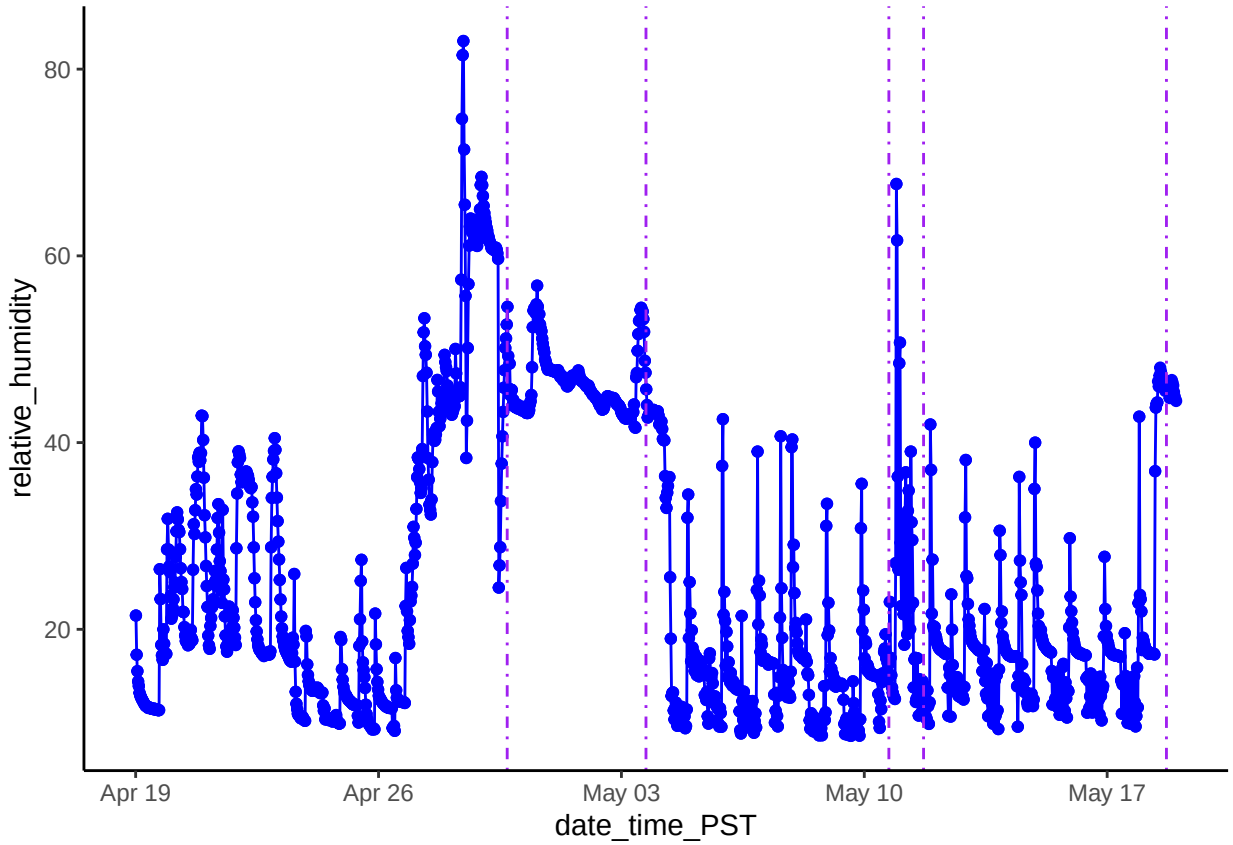
That looks much better. However, HOBO "dry cool 2" is actually from the humid cool treatment.

HOBO "dry hot 1":

```
# temperature
format_HOBO_data %>%
  dplyr::filter(HOBO_ID == "Dry_Hot_1") %>%
  ggplot(data = ., aes(x = date_time_PST,
                       y = temp_C,
                       color = HOBO_ID
                       )) +
  geom_point(color = "red") +
  geom_line(color = "red") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-04-30")),
             linetype = 4, color = "purple") +
  geom_hline(yintercept = 23) +
  geom_hline(yintercept = 29) +
  theme_classic()
```




```
# humidity
format_HOBO_data %>%
  dplyr::filter(HOBO_ID == "Dry_Hot_1") %>%
  ggplot(data = ., aes(x = date_time_PST,
                       y = relative_humidity,
                       color = HOBO_ID
                       )) +
  geom_point(color = "blue") +
  geom_line(color = "blue") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-04-30")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-04")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-11")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-12")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-19")),
             linetype = 4, color = "purple") +
  theme_classic()
```



This logger was definitely in the (failed) hot treatments until April 30, so I don't want to use data from before then. Between April 30 and May 4, humidity is a little high. It also does not have the typical diel variation which is from the 12-12 light cycle, so the logger must have been out of the experiment. Likely, it was not placed in the cool treatment immediately after being removed from the hot, so I will exclude this period as well. I will also exclude the spikes at May 11, 12, and 19.

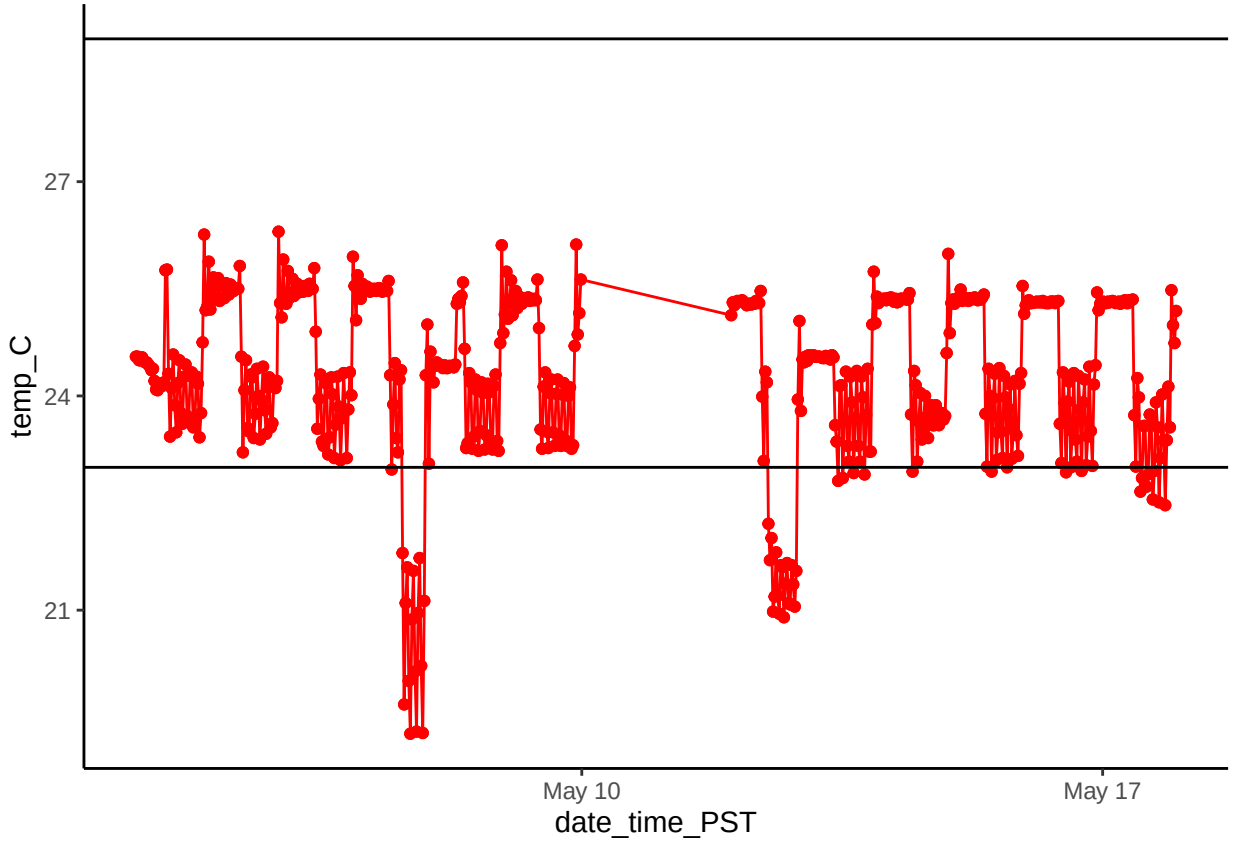
Try excluding. For some reason the dates plotted versus filtered line up differently, which is why I filter out the date before what I plotted.

```
unwanted_dates_dh1 <- as.Date(c(seq(as.Date("2021-04-19"),
                                    as.Date("2021-05-03"), "days"),
                                "2021-05-10", "2021-05-11", "2021-05-18"),
                              format = "%Y-%m-%d")
dh1_subset <- format_HOBO_data %>%
  dplyr::filter(HOBO_ID == "Dry_Hot_1") %>%
  dplyr::filter(date_only %nin% unwanted_dates_dh1)
```

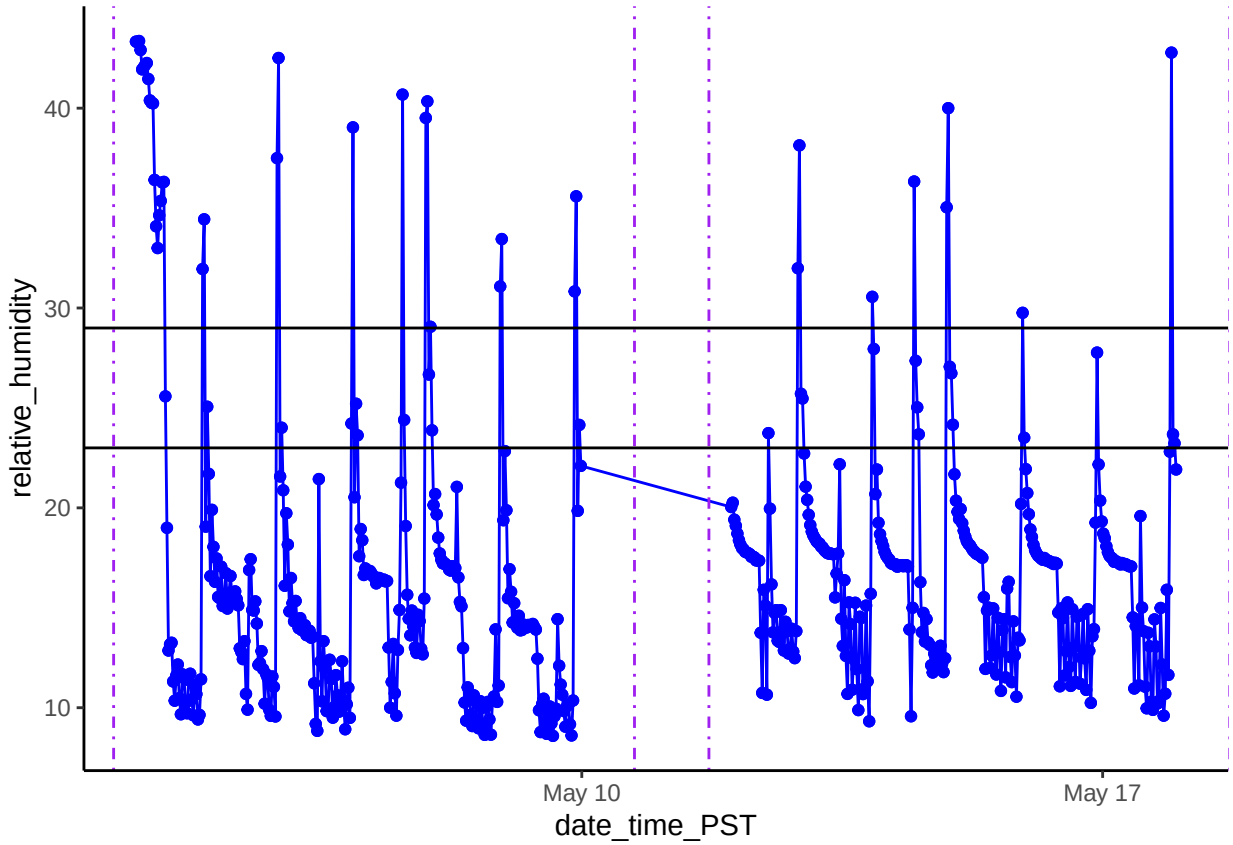
plot again to show that we removed erroneous data:

```
# temperature
dh1_subset %>%
  ggplot(data = ., aes(x = date_time_PST,
                       y = temp_C,
                       color = HOBO_ID
                       )) +
  geom_point(color = "red") +
  geom_line(color = "red") +
  geom_hline(yintercept = 23) +
  geom_hline(yintercept = 29) +
```

```
  theme_classic()
```



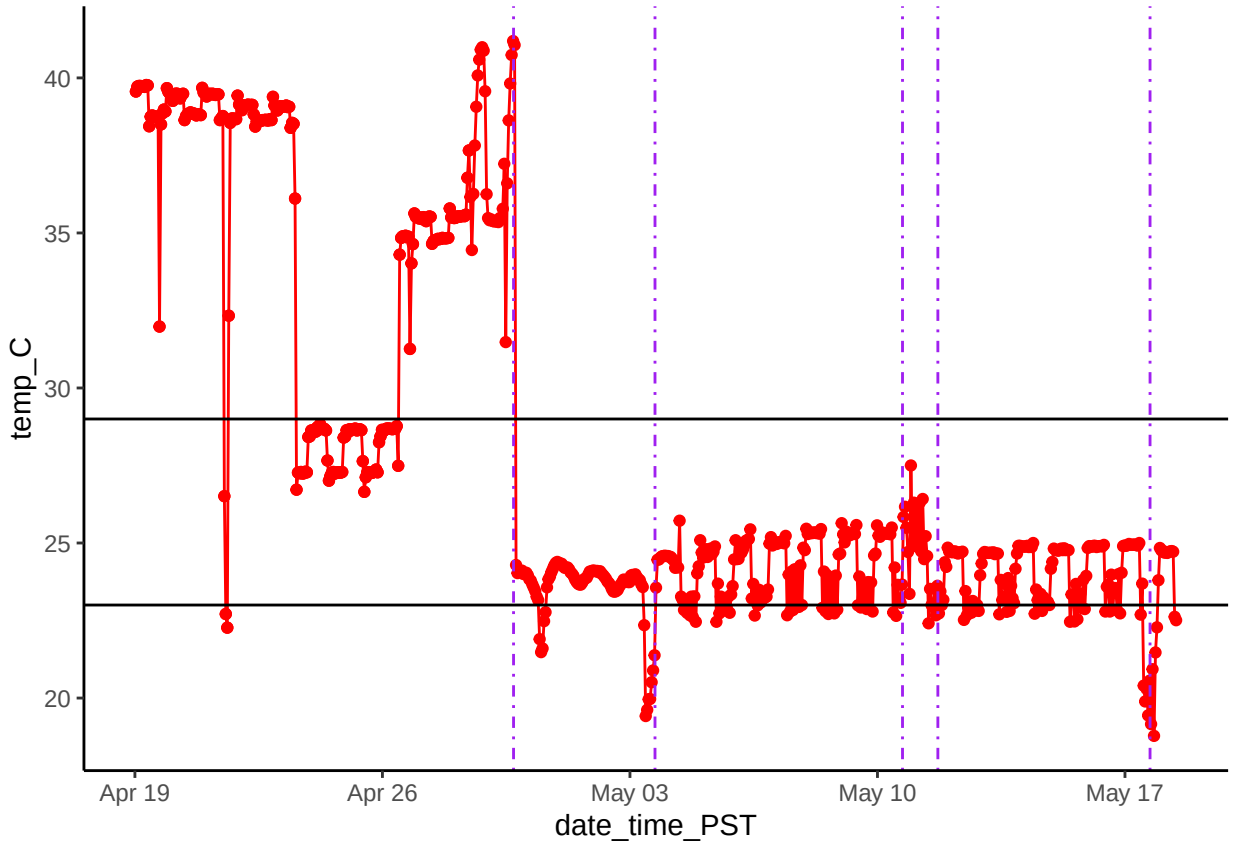

```
# humdity
dh1_subset %>%
  ggplot(data = ., aes(x = date_time_PST,
                       y = relative_humidity,
                       color = HOBO_ID
                       )) +
  geom_point(color = "blue") +
  geom_line(color = "blue") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-04")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-11")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-12")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-19")),
             linetype = 4, color = "purple") +
  geom_hline(yintercept = 23) +
  geom_hline(yintercept = 29) +
  theme_classic()
```



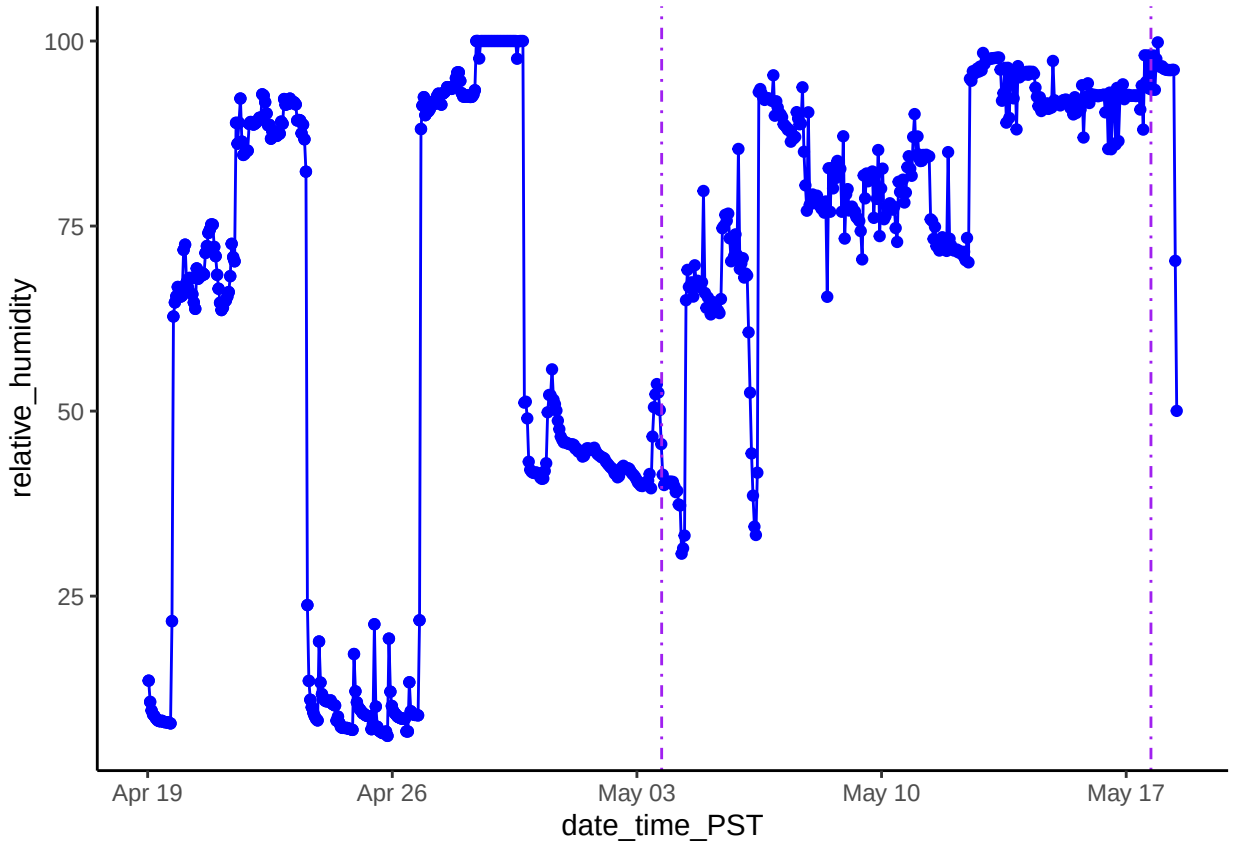
Much more realistic. This is for the cool dry treatment.

HOBO "dry hot 2":

```
# temperature
format_HOBO_data %>%
  dplyr::filter(HOBO_ID == "Dry_Hot_2") %>%
  ggplot(data = ., aes(x = date_time_PST,
                       y = temp_C,
                       color = HOBO_ID
                       )) +
  geom_point(color = "red") +
  geom_line(color = "red") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-04-30")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-04")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-18")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-11")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-12")),
             linetype = 4, color = "purple") +
  geom_hline(yintercept = 23) +
  geom_hline(yintercept = 29) +
  theme_classic()
```




```
# humidity
format_HOBO_data %>%
  dplyr::filter(HOBO_ID == "Dry_Hot_2") %>%
  ggplot(data = ., aes(x = date_time_PST,
                       y = relative_humidity,
                       color = HOBO_ID
                       )) +
  geom_point(color = "blue") +
  geom_line(color = "blue") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-04")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-18")),
             linetype = 4, color = "purple") +
  theme_classic()
```



This logger was also in the hot treatment until April 30, and was also not put into a treatment until May 4. I'm not worried about May 11-12 so much for this one, it may have been out& back into a chamber within the logging interval. On May 18, however, the dip is a measurement from outside the chamber.

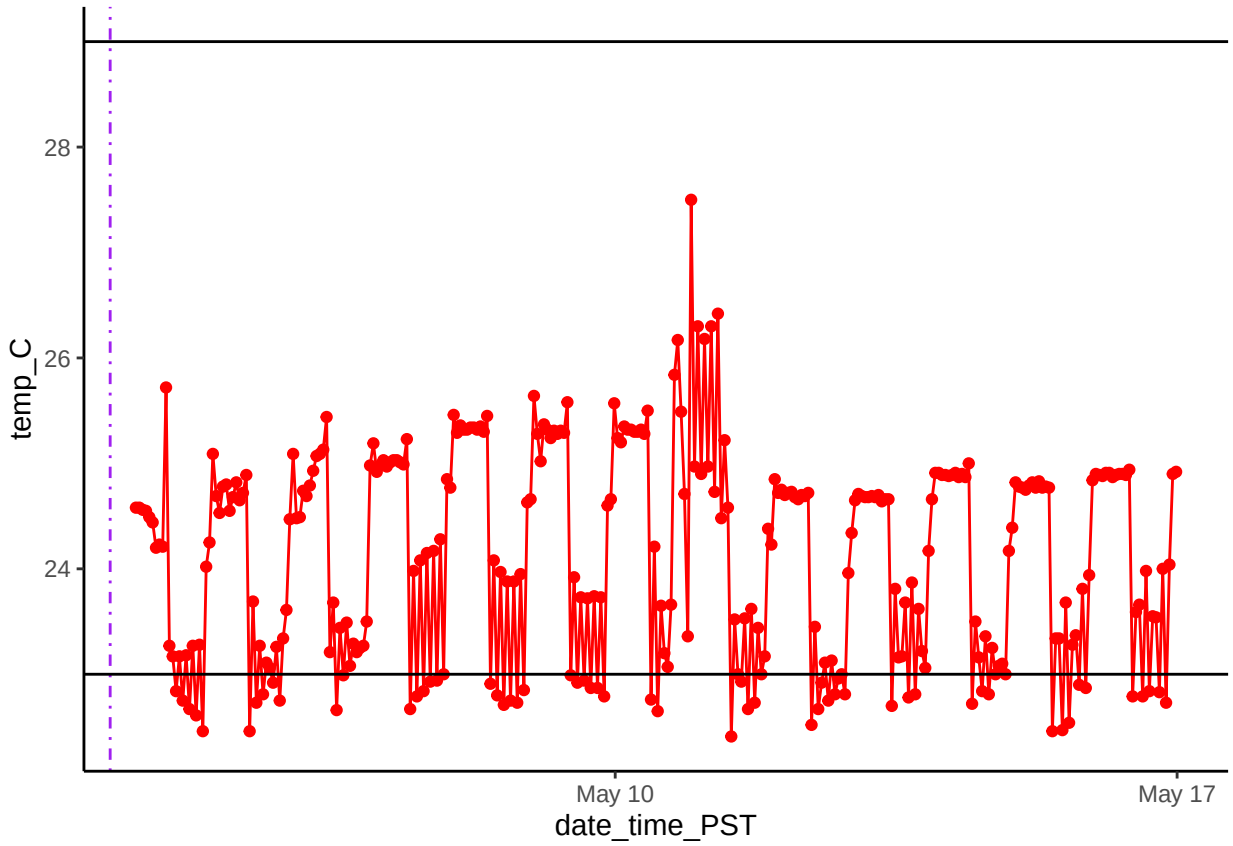
Try excluding. For some reason the dates plotted versus filtered line up differently, which is why I filter out the date before what I plotted.

```
unwanted_dates_dh2 <- as.Date(c(seq(as.Date("2021-04-19"),
                                    as.Date("2021-05-03"), "days"),
                                "2021-05-17", "2021-05-18"),
                              format = "%Y-%m-%d")
dh2_subset <- format_HOBO_data %>%
  dplyr::filter(HOBO_ID == "Dry_Hot_2") %>%
  dplyr::filter(date_only %nin% unwanted_dates_dh2)
```

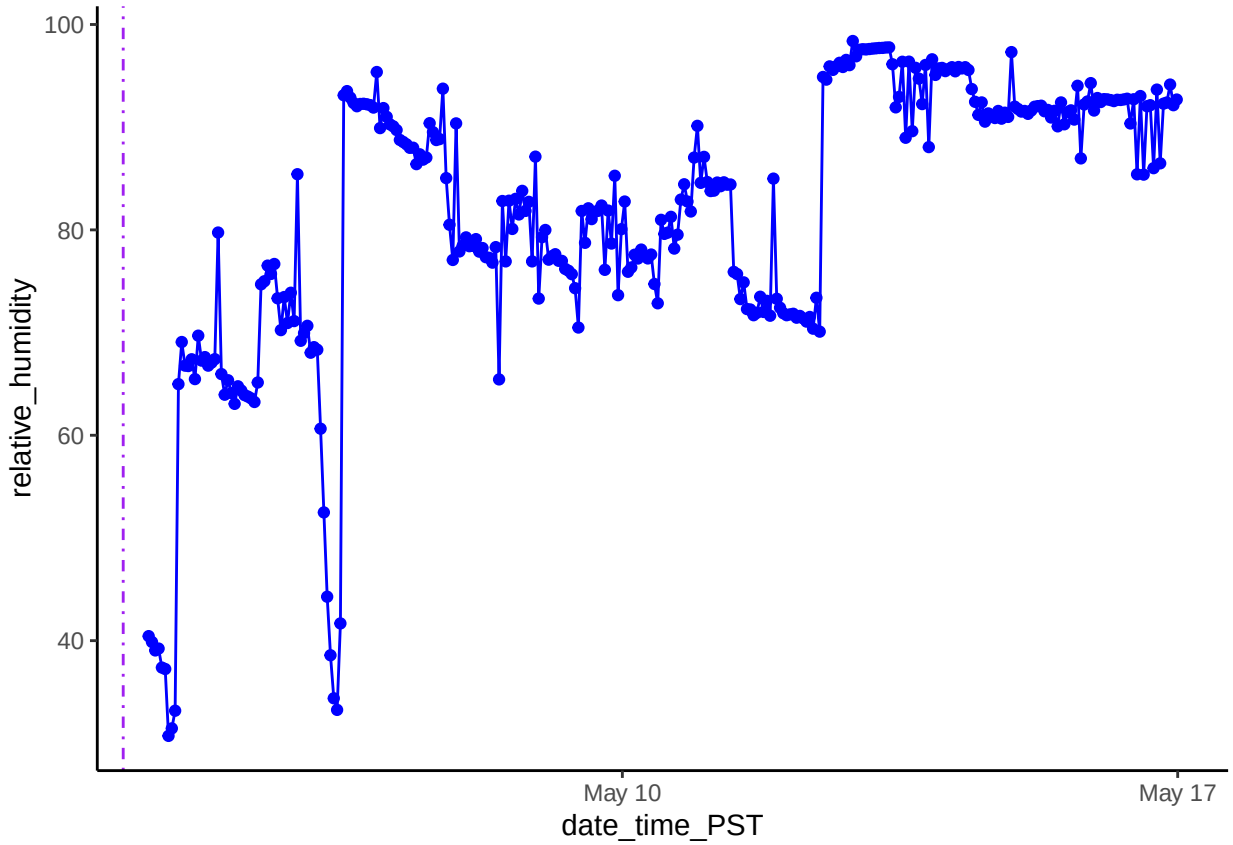
plot again to show that we removed erroneous data:

```
# temperature
dh2_subset %>%
  ggplot(data = ., aes(x = date_time_PST,
                       y = temp_C,
                       color = HOBO_ID
                       )) +
  geom_point(color = "red") +
  geom_line(color = "red") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-04")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-18")),
             linetype = 4, color = "purple") +
```

```
  geom_hline(yintercept = 23) +
  geom_hline(yintercept = 29) +
  theme_classic()
```



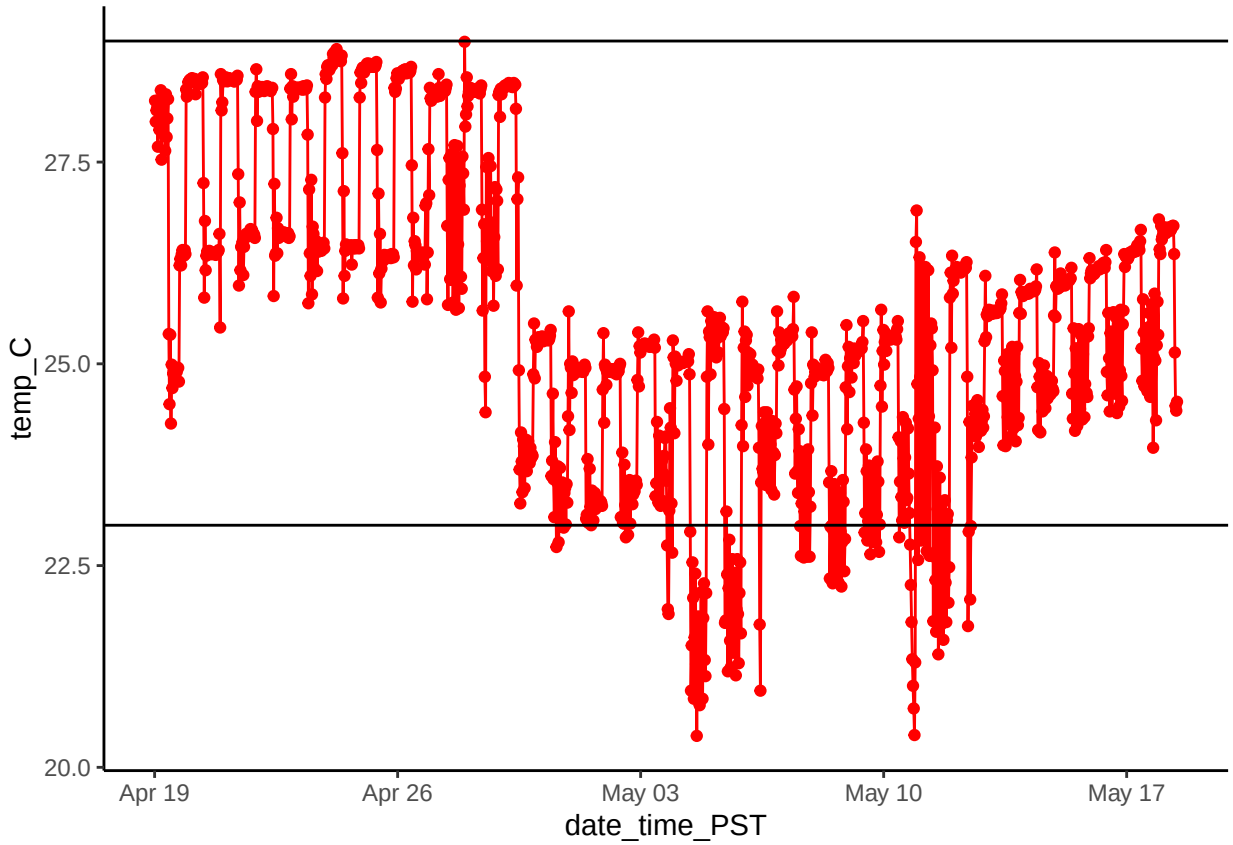

```
# humidity
dh2_subset %>%
  ggplot(data = ., aes(x = date_time_PST,
                       y = relative_humidity,
                       color = HOBO_ID
                       )) +
  geom_point(color = "blue") +
  geom_line(color = "blue") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-04")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-18")),
             linetype = 4, color = "purple") +
  theme_classic()
```



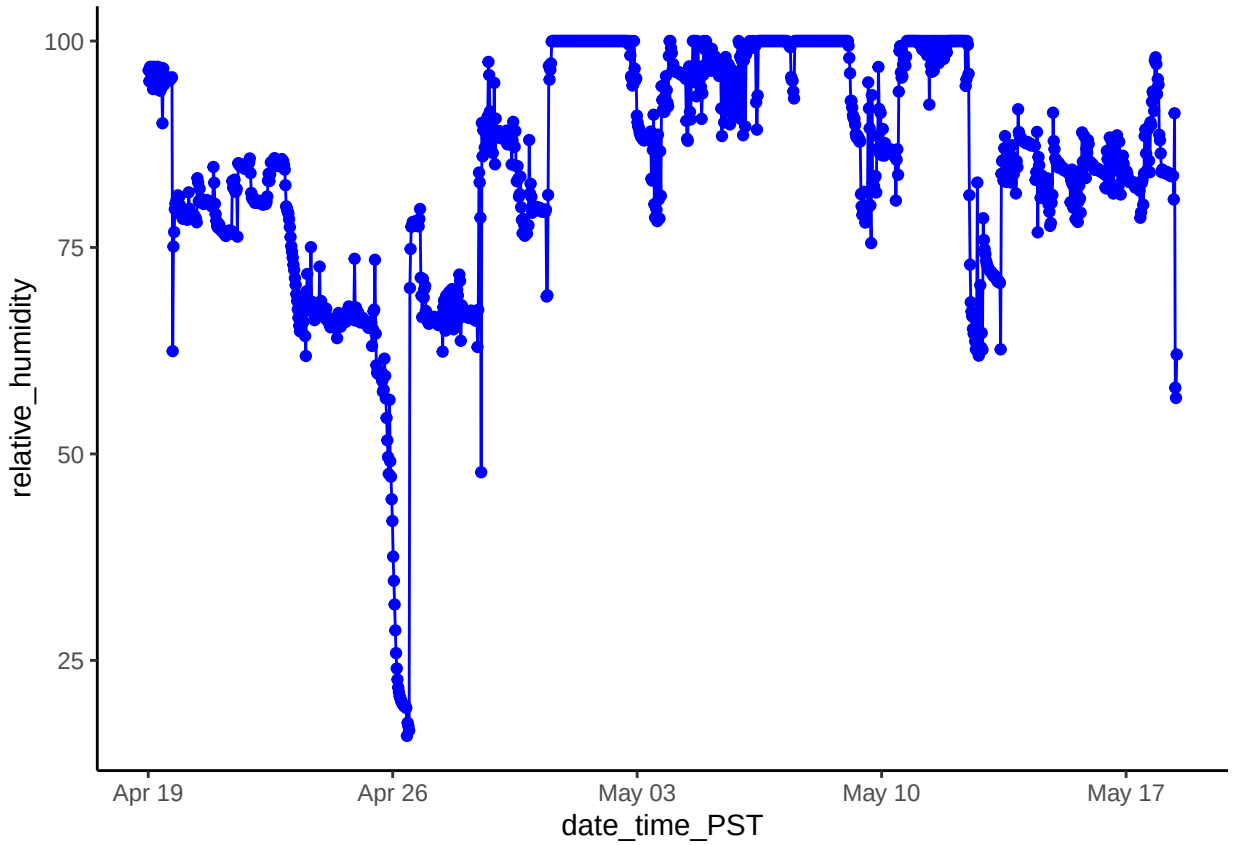
this logger was in cool & humid treatment

HOBO "humid cool":

```
# temperature
format_HOBO_data %>%
  dplyr::filter(HOBO_ID == "Humid_Cool") %>%
  ggplot(data = ., aes(x = date_time_PST,
                       y = temp_C,
                       color = HOBO_ID
                       )) +
  geom_point(color = "red") +
  geom_line(color = "red") +
  geom_hline(yintercept = 23) +
  geom_hline(yintercept = 29) +
  theme_classic()
```




```
# humidity
format_HOBO_data %>%
  dplyr::filter(HOBO_ID == "Humid_Cool") %>%
  ggplot(data = ., aes(x = date_time_PST,
                       y = relative_humidity,
                       color = HOBO_ID
                       )) +
  geom_point(color = "blue") +
  geom_line(color = "blue") +
  theme_classic()
```



This actually looks good as-is. We let the paper towel dry out around April 26, but that should be reflected.

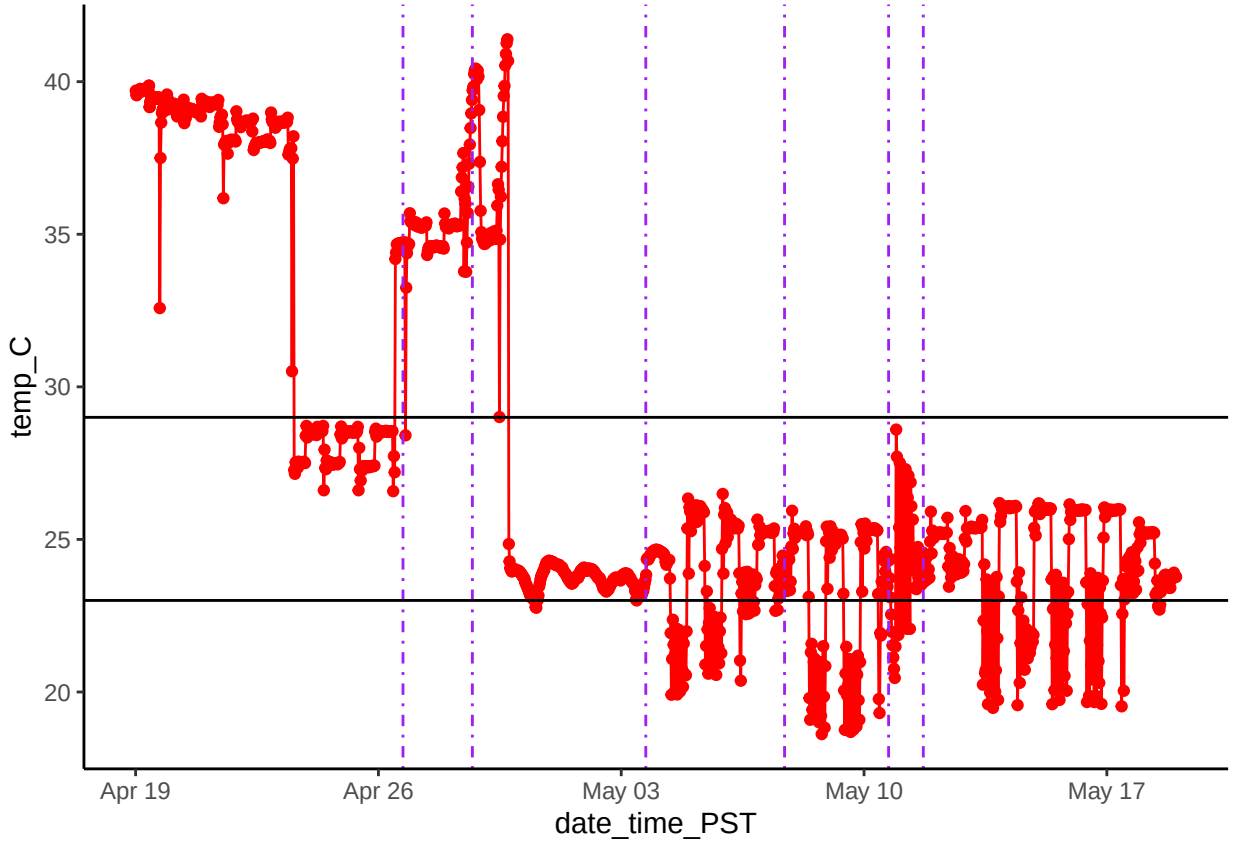
```
hc_subset <- format_HOBO_data %>%
  dplyr::filter(HOBO_ID == "Humid_Cool")
```

And, the logger was named correctly. It was in humid and cool treatment.

HOBO "humid hot":

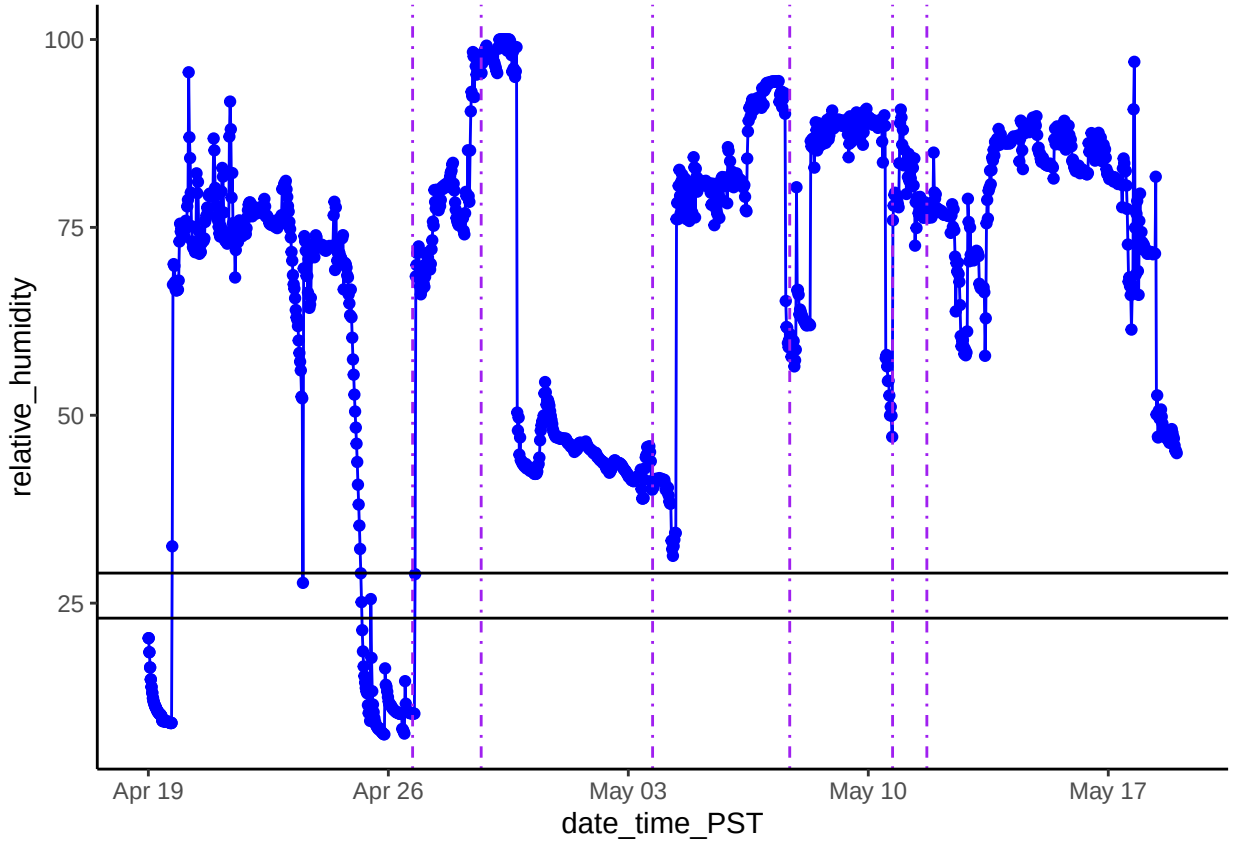
```
# temperature
format_HOBO_data %>%
  dplyr::filter(HOBO_ID == "Humid_Hot") %>%
  ggplot(data = ., aes(x = date_time_PST,
                       y = temp_C,
                       color = HOBO_ID
                       )) +
  geom_point(color = "red") +
  geom_line(color = "red") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-04-27")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-04-29")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-04")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-08")),
             linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-11")),
```

```
            linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-12")),
            linetype = 4, color = "purple") +
  geom_hline(yintercept = 23) +
  geom_hline(yintercept = 29) +
  theme_classic()
```




```
# humidity
format_HOBO_data %>%
  dplyr::filter(HOBO_ID == "Humid_Hot") %>%
  ggplot(data = ., aes(x = date_time_PST,
                       y = relative_humidity,
                       color = HOBO_ID
                       )) +
  geom_point(color = "blue") +
  geom_line(color = "blue") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-04-27")),
            linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-04-29")),
            linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-04")),
            linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-08")),
            linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-11")),
            linetype = 4, color = "purple") +
  geom_vline(xintercept = as.POSIXct(as.Date("2021-05-12")),
```

```
            linetype = 4, color = "purple") +
  geom_hline(yintercept = 23) +
  geom_hline(yintercept = 29) +
  theme_classic()
```

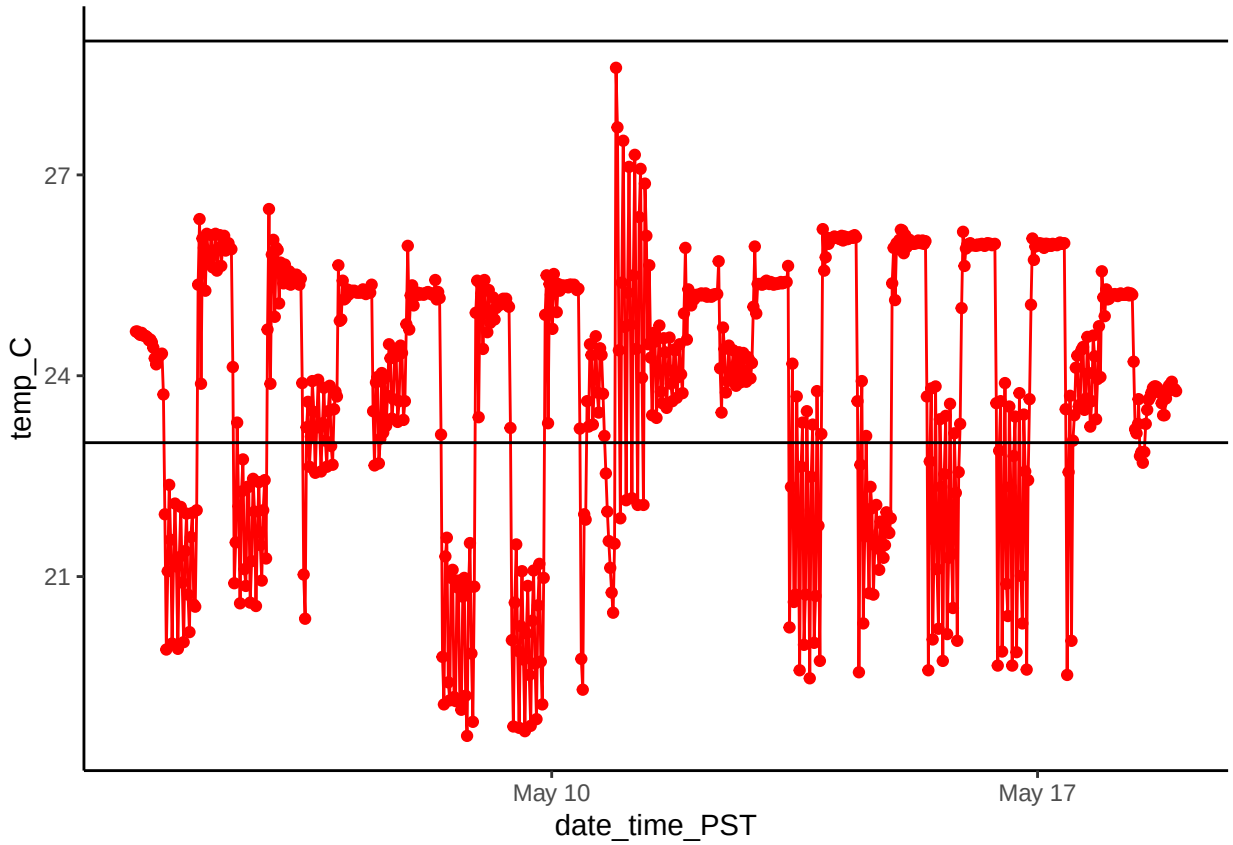


This logger was in the hot treatment, then not in any treatment, same as the ones before, so I will exclude the data up to May 4. It looks fine to be used in the cool humid treatment average otherwise.

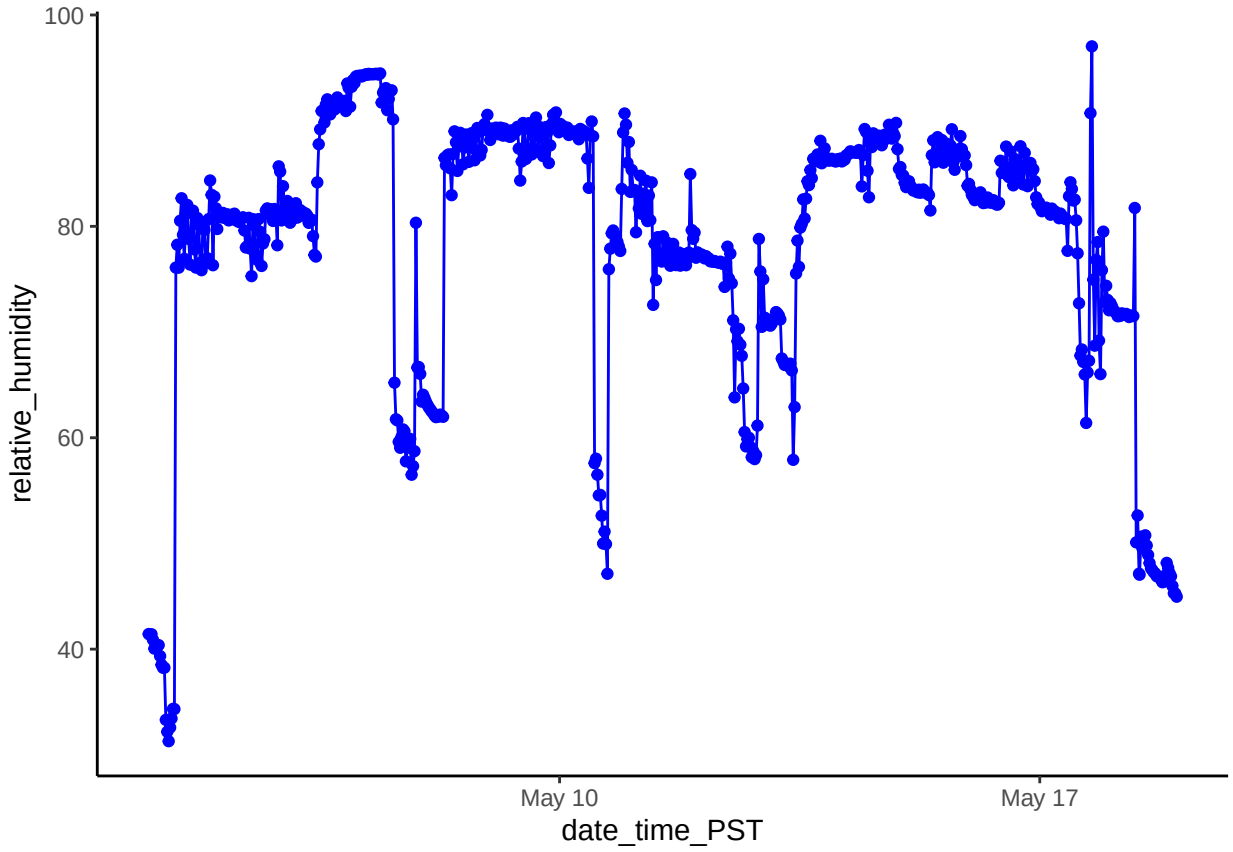
```
unwanted_dates_hh <- c(seq(as.Date("2021-04-19"),
                           as.Date("2021-05-03"), "days"))
hh_subset <- format_HOBO_data %>%
  dplyr::filter(HOBO_ID == "Humid_Hot") %>%
  dplyr::filter(date_only %nin% unwanted_dates_hh)
```

plot again to show that we removed erroneous data:

```
# temperature
hh_subset %>%
  ggplot(data = ., aes(x = date_time_PST,
                       y = temp_C,
                       color = HOBO_ID
                       )) +
  geom_point(color = "red") +
  geom_line(color = "red") +
  geom_hline(yintercept = 23) +
  geom_hline(yintercept = 29) +
  theme_classic()
```




```
# humidity
hh_subset %>%
  ggplot(data = ., aes(x = date_time_PST,
                       y = relative_humidity,
                       color = HOBO_ID
                       )) +
  geom_point(color = "blue") +
  geom_line(color = "blue") +
  theme_classic()
```



Correct Tmt Groups & Trial Numbers

```
tmts <- data.frame(HOBO_ID = unique(format_HOBO_data$HOBO_ID),
                   temp_tmt = as.factor(c("cool", "cool", "cool",
                                          "cool", "cool", "cool")),
                   humidity_tmt = as.factor(c("dry", "humid", "dry",
                                              "humid", "humid", "humid")))
trials <- data.frame(trial_num = as.factor(c(rep("1", 8),
                                     rep("2", 7),
                                     rep("3", 7),
                                     rep("4", 8))),
                                      # trial 1
                     date_only = c(c(seq(as.Date("2021-04-19"),
                                      as.Date("2021-04-26"), "days")),
                                     # trial 2
                                     c(seq(as.Date("2021-04-27"),
                                      as.Date("2021-05-03"), "days")),
                                     # trial 3
                                     c(seq(as.Date("2021-05-04"),
                                      as.Date("2021-05-10"), "days")),
                                     # trial 4
                                     c(seq(as.Date("2021-05-11"),
                                      as.Date("2021-05-18"), "days"))
                                    )
                     )
```

Rejoin All

```
clean_HOBO_data <- dc1_subset %>%
  rbind(dc2_subset) %>%
  rbind(dh1_subset) %>%
  rbind(dh2_subset) %>%
  rbind(hc_subset) %>%
  rbind(hh_subset) %>%
  left_join(tmts, by = 'HOBO_ID') %>%
  left_join(trials, by = 'date_only')
summary(clean_HOBO_data)
```

```
##  date_time_PST                    temp_C      relative_humidity
##  Min.   :2021-04-19 00:16:33   Min.   :17.84   Min.   :  2.93
##  1st Qu.:2021-04-30 12:46:33   1st Qu.:23.84   1st Qu.: 12.78
##  Median :2021-05-07 11:45:16   Median :25.06   Median : 58.86
##  Mean   :2021-05-06 08:11:52   Mean   :25.01   Mean   : 50.00
##  3rd Qu.:2021-05-13 12:15:16   3rd Qu.:25.87   3rd Qu.: 86.23
##  Max.   :2021-05-18 23:54:56   Max.   :29.61   Max.   :100.00
##     dew_pt_C             HOBO_ID       date_only          temp_tmt
##  Min.   :-22.510   Dry_Cool_1:2182   Min.   :2021-04-19   cool:6933
##  1st Qu.: -4.860   Dry_Cool_2:1152   1st Qu.:2021-04-30
##  Median : 15.590   Dry_Hot_1 :1152   Median :2021-05-07
##  Mean   :  8.746   Dry_Hot_2 : 312   Mean   :2021-05-05
##  3rd Qu.: 22.510   Humid_Cool:1414   3rd Qu.:2021-05-13
##  Max.   : 28.120   Humid_Hot : 721   Max.   :2021-05-18
##  humidity_tmt trial_num
##  dry  :3334   1:1248
##  humid:3599   2:1008
##               3:2232
##               4:2445
##
##
```

Get Means

Grand Mean

```
grand_mean <- clean_HOBO_data %>%
  group_by(temp_tmt) %>%
  summarise(temp_mean = mean(temp_C),
            temp_SD = sd(temp_C))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
grand_mean
```

```
## # A tibble: 1 x 3
##   temp_tmt temp_mean temp_SD
##   <fct>        <dbl>   <dbl>
## 1 cool          25.0    1.85
```

Treatment Overall

```
tmt_only_means <- clean_HOBO_data %>%
  group_by(humidity_tmt) %>%
  summarise(temp_mean = mean(temp_C),
            temp_SD = sd(temp_C),
            humidity_mean = mean(relative_humidity),
            humidity_SD = sd(relative_humidity))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
tmt_only_means
```

```
## # A tibble: 2 x 5
##   humidity_tmt temp_mean temp_SD humidity_mean humidity_SD
##   <fct>            <dbl>   <dbl>         <dbl>       <dbl>
## 1 dry               25.1    1.62          14.1        6.71
## 2 humid             24.9    2.04          83.3       13.7
```

Treatment by Trial

```
tmt_trial_means <- clean_HOBO_data %>%
  group_by(humidity_tmt, trial_num) %>%
  summarise(temp_mean = mean(temp_C),
            temp_SD = sd(temp_C),
            humidity_mean = mean(relative_humidity),
            humidity_SD = sd(relative_humidity)) %>%
  arrange(trial_num)
```

```
## `summarise()` regrouping output by 'humidity_tmt' (override with `.groups` argument)
```

```
tmt_trial_means
```

```
## # A tibble: 8 x 6
## # Groups:   humidity_tmt [2]
##   humidity_tmt trial_num temp_mean temp_SD humidity_mean humidity_SD
##   <fct>        <fct>         <dbl>   <dbl>         <dbl>       <dbl>
## 1 dry          1              27.4    1.21          11.1        4.16
## 2 humid        1              27.5    1.29          71.8       13.9
## 3 dry          2              25.0    1.41          10.9        4.95
## 4 humid        2              25.5    1.93          89.2       10.3
## 5 dry          3              24.5    1.05          15.0        7.42
## 6 humid        3              23.8    1.73          83.7       14.4
## 7 dry          4              24.5    1.01          15.9        6.76
## 8 humid        4              24.5    1.41          85.3       11.0
```

```
write.csv(tmt_trial_means, "./HOBO_tmt_trial_diffs.csv")
```

Check Statistical Differences

```
# based on humidity treatment
temp_tmt_mod <- lm(data = clean_HOBO_data,
                   temp_C ~ humidity_tmt)
summary(temp_tmt_mod)
```

```
##
```

```
## Call:
## lm(formula = temp_C ~ humidity_tmt, data = clean_HOBO_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.0608 -1.1861  0.0939  0.8039  4.7092
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       25.12611    0.03203 784.439  < 2e-16 ***
## humidity_tmthumid -0.22535    0.04446  -5.069  4.1e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.849 on 6931 degrees of freedom
## Multiple R-squared:  0.003693,   Adjusted R-squared:  0.00355
## F-statistic: 25.69 on 1 and 6931 DF,  p-value: 4.105e-07
```

```
humidity_tmt_mod <- lm(data = clean_HOBO_data,
                   relative_humidity ~ humidity_tmt)
summary(humidity_tmt_mod)
```

```
##
## Call:
## lm(formula = relative_humidity ~ humidity_tmt, data = clean_HOBO_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -67.437  -4.806  -0.056   5.673  31.434
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        14.0656     0.1885   74.61   <2e-16 ***
## humidity_tmthumid  69.2316     0.2616  264.61   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.88 on 6931 degrees of freedom
## Multiple R-squared:  0.9099, Adjusted R-squared:  0.9099
## F-statistic: 7.002e+04 on 1 and 6931 DF,  p-value: < 2.2e-16
```

```
# based on trial
temp_trial_mod <- lm(data = clean_HOBO_data,
                temp_C ~ trial_num)
summary(temp_trial_mod)
```

```
##
## Call:
## lm(formula = temp_C ~ trial_num, data = clean_HOBO_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.2632 -0.9525  0.1775  0.9950  4.4968
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 27.44580    0.03979  689.76   <2e-16 ***
## trial_num2  -2.10331    0.05953  -35.33   <2e-16 ***
## trial_num3  -3.34260    0.04968  -67.28   <2e-16 ***
## trial_num4  -2.99085    0.04890  -61.16   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.406 on 6929 degrees of freedom
## Multiple R-squared:  0.4246, Adjusted R-squared:  0.4244
## F-statistic:  1705 on 3 and 6929 DF,  p-value: < 2.2e-16
```

```
humidity_trial_mod <- lm(data = clean_HOBO_data,
                  relative_humidity ~ trial_num)
summary(humidity_trial_mod)
```

```
##
## Call:
## lm(formula = relative_humidity ~ trial_num, data = clean_HOBO_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -53.697 -35.761   6.883  35.509  57.788
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   39.092      1.013   38.59   <2e-16 ***
## trial_num2    20.255      1.516   13.36   <2e-16 ***
## trial_num3    12.119      1.265    9.58   <2e-16 ***
## trial_num4    11.529      1.245    9.26   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 35.79 on 6929 degrees of freedom
## Multiple R-squared:  0.02641,    Adjusted R-squared:  0.02599
## F-statistic: 62.65 on 3 and 6929 DF,  p-value: < 2.2e-16
```

temp summary: dry mean = 25.1, humid mean = 24.9, estimate = -0.225, t = -5.069, $p < 0.0001$

humidity summary: estimate = 69.232, t = 264.61, $p < 0.0001$

Both humidity treatment and trial number, AND their interaction is significant for predicting the temperature and humidity the data loggers logged.

Need pairwise to know which ones are actually different:

```
TukeyHSD(aov(temp_tmt_mod))
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = temp_tmt_mod)
##
## $humidity_tmt
##                 diff        lwr        upr p adj
## humid-dry -0.2253457 -0.3124942 -0.1381972 4e-07
```

```r
TukeyHSD(aov(humidity_tmt_mod))
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = humidity_tmt_mod)
##
## $humidity_tmt
##               diff      lwr      upr p adj
## humid-dry 69.23158 68.71869 69.74448     0
```

```r
TukeyHSD(aov(temp_trial_mod))
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = temp_trial_mod)
##
## $trial_num
##           diff        lwr        upr p adj
## 2-1 -2.1033112 -2.2562768 -1.9503456     0
## 3-1 -3.3425979 -3.4702703 -3.2149255     0
## 4-1 -2.9908483 -3.1165105 -2.8651861     0
## 3-2 -1.2392867 -1.3763613 -1.1022121     0
## 4-2 -0.8875371 -1.0227414 -0.7523329     0
## 4-3  0.3517496  0.2460048  0.4574943     0
```

```r
TukeyHSD(aov(humidity_trial_mod))
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = humidity_trial_mod)
##
## $trial_num
##           diff        lwr       upr     p adj
## 2-1 20.2548691  16.360168 24.149571 0.0000000
## 3-1 12.1190806   8.868377 15.369785 0.0000000
## 4-1 11.5293930   8.329871 14.728915 0.0000000
## 3-2 -8.1357885 -11.625884 -4.645693 0.0000000
## 4-2 -8.7254761 -12.167951 -5.283001 0.0000000
## 4-3 -0.5896876  -3.282086  2.102711 0.9430523
```