

# Arranging logger data from HoboLink

*Jens Åström*

*07 May, 2021*

## Contents

<b>Intro</b>	<b>1</b>
<b>Load the data</b>	<b>1</b>
<b>Package this into a function</b>	<b>6</b>
<b>A quick look at the data</b>	<b>8</b>
<b>Combine with single files from loggers that weren't synced</b>	<b>9</b>
<b>Handle the MX2201 loggers</b>	<b>11</b>
<b>A quick look at the MX2202 data</b>	<b>13</b>

## Intro

We have several HOBO loggers from ONSET that get “emptied” every now and again by field workers. The data is uploaded to the cloud through the app “HoboConnect”. After that, the data from several loggers can be downloaded from <http://www.hobolink.com>. But this data export needs a bit of wrangling before it can be used. The different data streams from each logger (temperature, humidity, light) all get a separate columns. Here we document a script to turn this into a more usable long format.

## Load the data

We have a single export file from Hobolink.com with many loggers as a csv file. We also have some individual csv files that failed to upload to the Hobo site, which we'll handle later on.

```
inputFile <- "../rawData/Insektoverv_k_2020_2020_11_10_11_56_24_UTC_1.csv"

rawDat <- read_csv(inputFile,col_types = cols(.default = "c"))
```

```

dat <- rawDat %>%
  select(-"Line#") %>%
  mutate(date = as.POSIXct(Date, format = "%m/%d/%y %H:%M:%S")) %>%
  mutate_if(is_character, as.double) %>%
  select(-Date)

```

```

## Warning: Problem with `mutate()` input `Date`.
## i NAs introduced by coercion
## i Input `Date` is `Primitive("as.double")(Date)`.

## Warning in mask$eval_all_mutate(dots[[i]]): NAs introduced by coercion
dat

```

```

## # A tibble: 234,096 x 61
##   `Temperature (M~`RH (MX-RH-2 20~`Dew Point (MX--`Temperature (M~
##           <dbl>           <dbl>           <dbl>           <dbl>
## 1             NA             NA             NA             NA
## 2          24.0          26.1          3.37          NA
## 3             NA             NA             NA             NA
## 4             NA             NA             NA          1.88
## 5             NA             NA             NA             NA
## 6             NA             NA             NA             NA
## 7             NA             NA             NA             NA
## 8             NA             NA             NA             NA
## 9             NA             NA             NA             NA
## 10            NA             NA             NA             NA
## # ... with 234,086 more rows, and 57 more variables: `RH (MX-RH-2
## #   20835814:20835814-2), %, 20835814` <dbl>, `Dew Point (MX-TEMP-2
## #   20835814:20835814-4), *C, 20835814` <dbl>, `Temperature (MX-TEMP-2
## #   20843239:20843239-1), *C, 20843239` <dbl>, `RH (MX-RH-2
## #   20843239:20843239-2), %, 20843239` <dbl>, `Dew Point (MX-TEMP-2
## #   20843239:20843239-4), *C, 20843239` <dbl>, `Temperature (MX-TEMP-2
## #   20843238:20843238-1), *C, 20843238` <dbl>, `RH (MX-RH-2
## #   20843238:20843238-2), %, 20843238` <dbl>, `Dew Point (MX-TEMP-2
## #   20843238:20843238-4), *C, 20843238` <dbl>, `Temperature (MX-TEMP-2
## #   20843235:20843235-1), *C, 20843235` <dbl>, `RH (MX-RH-2
## #   20843235:20843235-2), %, 20843235` <dbl>, `Dew Point (MX-TEMP-2
## #   20843235:20843235-4), *C, 20843235` <dbl>, `Temperature (MX-TEMP-2
## #   20843233:20843233-1), *C, 20843233` <dbl>, `RH (MX-RH-2
## #   20843233:20843233-2), %, 20843233` <dbl>, `Dew Point (MX-TEMP-2
## #   20843233:20843233-4), *C, 20843233` <dbl>, `Temperature (MX-TEMP-2
## #   20843231:20843231-1), *C, 20843231` <dbl>, `RH (MX-RH-2
## #   20843231:20843231-2), %, 20843231` <dbl>, `Dew Point (MX-TEMP-2
## #   20843231:20843231-4), *C, 20843231` <dbl>, `Temperature (MX-TEMP-2
## #   20843230:20843230-1), *C, 20843230` <dbl>, `RH (MX-RH-2
## #   20843230:20843230-2), %, 20843230` <dbl>, `Dew Point (MX-TEMP-2

```

```
## # 20843230:20843230-4), *C, 20843230` <dbl>, `Temperature (MX-TEMP-2
## # 20843229:20843229-1), *C, 20843229` <dbl>, `RH (MX-RH-2
## # 20843229:20843229-2), %, 20843229` <dbl>, `Dew Point (MX-TEMP-2
## # 20843229:20843229-4), *C, 20843229` <dbl>, `Temperature (MX-TEMP-2
## # 20843228:20843228-1), *C, 20843228` <dbl>, `RH (MX-RH-2
## # 20843228:20843228-2), %, 20843228` <dbl>, `Dew Point (MX-TEMP-2
## # 20843228:20843228-4), *C, 20843228` <dbl>, `Temperature (MX-TEMP-2
## # 20835825:20835825-1), *C, 20835825` <dbl>, `RH (MX-RH-2
## # 20835825:20835825-2), %, 20835825` <dbl>, `Dew Point (MX-TEMP-2
## # 20835825:20835825-4), *C, 20835825` <dbl>, `Temperature (MX-TEMP-2
## # 20835824:20835824-1), *C, 20835824` <dbl>, `RH (MX-RH-2
## # 20835824:20835824-2), %, 20835824` <dbl>, `Dew Point (MX-TEMP-2
## # 20835824:20835824-4), *C, 20835824` <dbl>, `Temperature (MX-TEMP-2
## # 20835823:20835823-1), *C, 20835823` <dbl>, `RH (MX-RH-2
## # 20835823:20835823-2), %, 20835823` <dbl>, `Dew Point (MX-TEMP-2
## # 20835823:20835823-4), *C, 20835823` <dbl>, `Temperature (MX-TEMP-2
## # 20835822:20835822-1), *C, 20835822` <dbl>, `RH (MX-RH-2
## # 20835822:20835822-2), %, 20835822` <dbl>, `Dew Point (MX-TEMP-2
## # 20835822:20835822-4), *C, 20835822` <dbl>, `Temperature (MX-TEMP-2
## # 20835821:20835821-1), *C, 20835821` <dbl>, `RH (MX-RH-2
## # 20835821:20835821-2), %, 20835821` <dbl>, `Dew Point (MX-TEMP-2
## # 20835821:20835821-4), *C, 20835821` <dbl>, `Temperature (MX-TEMP-2
## # 20835820:20835820-1), *C, 20835820` <dbl>, `RH (MX-RH-2
## # 20835820:20835820-2), %, 20835820` <dbl>, `Dew Point (MX-TEMP-2
## # 20835820:20835820-4), *C, 20835820` <dbl>, `Temperature (MX-TEMP-2
## # 20835819:20835819-1), *C, 20835819` <dbl>, `RH (MX-RH-2
## # 20835819:20835819-2), %, 20835819` <dbl>, `Dew Point (MX-TEMP-2
## # 20835819:20835819-4), *C, 20835819` <dbl>, `Temperature (MX-TEMP-2
## # 20835818:20835818-1), *C, 20835818` <dbl>, `RH (MX-RH-2
## # 20835818:20835818-2), %, 20835818` <dbl>, `Dew Point (MX-TEMP-2
## # 20835818:20835818-4), *C, 20835818` <dbl>, `Temperature (MX-TEMP-2
## # 20835817:20835817-1), *C, 20835817` <dbl>, `RH (MX-RH-2
## # 20835817:20835817-2), %, 20835817` <dbl>, `Dew Point (MX-TEMP-2
## # 20835817:20835817-4), *C, 20835817` <dbl>, `Temperature (MX-TEMP-2
## # 20835815:20835815-1), *C, 20835815` <dbl>, `RH (MX-RH-2
## # 20835815:20835815-2), %, 20835815` <dbl>, `Dew Point (MX-TEMP-2
## # 20835815:20835815-4), *C, 20835815` <dbl>, date <dtm>
```

That's quite the number of columns...

We have to pivot this data set to a longer format. We also get rid of the rows with no data. First we separate the different data types.

```
temp <- dat %>%
  pivot_longer(cols = starts_with("Temperature"),
               names_to = "logger_id",
               values_to = "temperature") %>%
```

```

select(date,
       logger_id,
       temperature) %>%
filter(!is.na(temperature))

rh <- dat %>%
  pivot_longer(cols = starts_with("RH"),
               names_to = "logger_id",
               values_to = "rh") %>%
  select(date,
         logger_id,
         rh) %>%
  filter(!is.na(rh))

dew_point <- dat %>%
  pivot_longer(cols = starts_with("Dew"),
               names_to = "logger_id",
               values_to = "dew_point") %>%
  select(date,
         logger_id,
         dew_point) %>%
  filter(!is.na(dew_point))

```

The data now looks like this.

```
temp
```

```

## # A tibble: 234,096 x 3
##   date                logger_id                temperature
##   <dtm>                <chr>                <dbl>
## 1 2020-05-14 14:36:01 Temperature (MX-TEMP-2 20835823:20835823-1),~ 23.0
## 2 2020-05-14 14:36:41 Temperature (MX-TEMP-2 20843236:20843236-1),~ 24.0
## 3 2020-05-14 14:37:59 Temperature (MX-TEMP-2 20835821:20835821-1),~ 22.8
## 4 2020-05-14 14:39:05 Temperature (MX-TEMP-2 20835814:20835814-1),~ 1.88
## 5 2020-05-14 14:42:40 Temperature (MX-TEMP-2 20835817:20835817-1),~ 23.1
## 6 2020-05-14 14:43:41 Temperature (MX-TEMP-2 20835818:20835818-1),~ 23.0
## 7 2020-05-14 14:44:35 Temperature (MX-TEMP-2 20835820:20835820-1),~ 23.1
## 8 2020-05-14 14:47:30 Temperature (MX-TEMP-2 20843231:20843231-1),~ 23.1
## 9 2020-05-14 14:49:11 Temperature (MX-TEMP-2 20843229:20843229-1),~ 23
## 10 2020-05-14 14:56:01 Temperature (MX-TEMP-2 20835823:20835823-1),~ 23.0
## # ... with 234,086 more rows

```

Time to strip the logger names.

```

temp <- temp %>%
  mutate(logger_id = str_extract(logger_id,
                                "[^, ]+$"))

```

```
rh <- rh %>%
  mutate(logger_id = str_extract(logger_id,
                                "[^, ]+$"))
dew_point <- dew_point %>%
  mutate(logger_id = str_extract(logger_id,
                                "[^, ]+$"))
```

Check to see that the dates are the same for the datasets.

```
all(all(temp$date == rh$date),
all(rh$date == dew_point$date))
```

```
## [1] TRUE
```

Merge the different data types.

```
combDat <- temp %>%
  full_join(rh,
            by = c("date" = "date",
                  "logger_id" = "logger_id")) %>%
  full_join(dew_point,
            by = c("date" = "date",
                  "logger_id" = "logger_id")) %>%
  arrange(logger_id,
           date) %>%
  mutate(logger_type = "MX2301A") %>%
  select(date,
         logger_type,
         logger_id,
         temperature,
         rh,
         dew_point)
```

The data now looks like this

```
combDat
```

```
## # A tibble: 234,096 x 6
##   date           logger_type logger_id temperature    rh dew_point
##   <dtm>          <chr>      <chr>         <dbl> <dbl>    <dbl>
## 1 2020-05-14 14:39:05 MX2301A    20835814      1.88  79.3    -1.32
## 2 2020-05-14 14:59:05 MX2301A    20835814      3.27  70.0    -1.68
## 3 2020-05-14 15:19:05 MX2301A    20835814      3.64  62.8    -2.79
## 4 2020-05-14 15:39:05 MX2301A    20835814      3.47  60.1    -3.53
## 5 2020-05-14 15:59:05 MX2301A    20835814      3.38  62.4    -3.12
## 6 2020-05-14 16:19:05 MX2301A    20835814      3.36  65.8    -2.41
## 7 2020-05-14 16:39:05 MX2301A    20835814      2.03  77.2    -1.54
## 8 2020-05-14 16:59:05 MX2301A    20835814      1.91  78.8    -1.38
```

```
## 9 2020-05-14 17:19:05 MX2301A      20835814      1.69  81.4    -1.15
## 10 2020-05-14 17:39:05 MX2301A     20835814      1.72  81.2    -1.15
## # ... with 234,086 more rows
```

## Package this into a function

```
longerHobo2301 <- function(inputFile){

  rawDat <- read_csv(inputFile,col_types = cols(.default = "c"))

  dat <- rawDat %>%
    select(-"Line#") %>%
    mutate(date = as.POSIXct(Date, format = "%m/%d/%y %H:%M:%S")) %>%
    mutate_if(is_character, as.double) %>%
    select(-Date)

  temp <- dat %>%
    pivot_longer(cols = starts_with("Temperature"),
                 names_to = "logger_id",
                 values_to = "temperature") %>%
    select(date,
           logger_id,
           temperature) %>%
    filter(!is.na(temperature))

  rh <- dat %>%
    pivot_longer(cols = starts_with("RH"),
                 names_to = "logger_id",
                 values_to = "rh") %>%
    select(date,
           logger_id,
           rh) %>%
    filter(!is.na(rh))

  dew_point <- dat %>%
    pivot_longer(cols = starts_with("Dew"),
                 names_to = "logger_id",
                 values_to = "dew_point") %>%
    select(date,
           logger_id,
           dew_point) %>%
    filter(!is.na(dew_point))
```

```

temp <- temp %>%
  mutate(logger_id = str_extract(logger_id,
                                "[^, ]+$"))
rh <- rh %>%
  mutate(logger_id = str_extract(logger_id,
                                "[^, ]+$"))
dew_point <- dew_point %>%
  mutate(logger_id = str_extract(logger_id,
                                "[^, ]+$"))

if(!all(all(temp$date == rh$date),
all(rh$date == dew_point$date))) stop("Tables datetimes doesn't match")

combDat <- temp %>%
full_join(rh,
          by = c("date" = "date",
                  "logger_id" = "logger_id")) %>%
full_join(dew_point,
          by = c("date" = "date",
                  "logger_id" = "logger_id")) %>%
arrange(logger_id,
         date) %>%
  mutate(logger_type = "MX2301A") %>%
  select(date,
         logger_type,
         logger_id,
         temperature,
         rh,
         dew_point)

return(combDat)
}

```

We can check that the function produces the same results as the step by step script.

```

combDat2 <- longerHobo2301("../rawData/Insektoverv_k_2020_2020_11_10_11_56_24_UTC_1.csv")

## Warning: Problem with `mutate()` input `Date`.
## i NAs introduced by coercion
## i Input `Date` is `.Primitive("as.double")(Date)`.
## Warning in mask$eval_all_mutate(dots[[i]]): NAs introduced by coercion

```

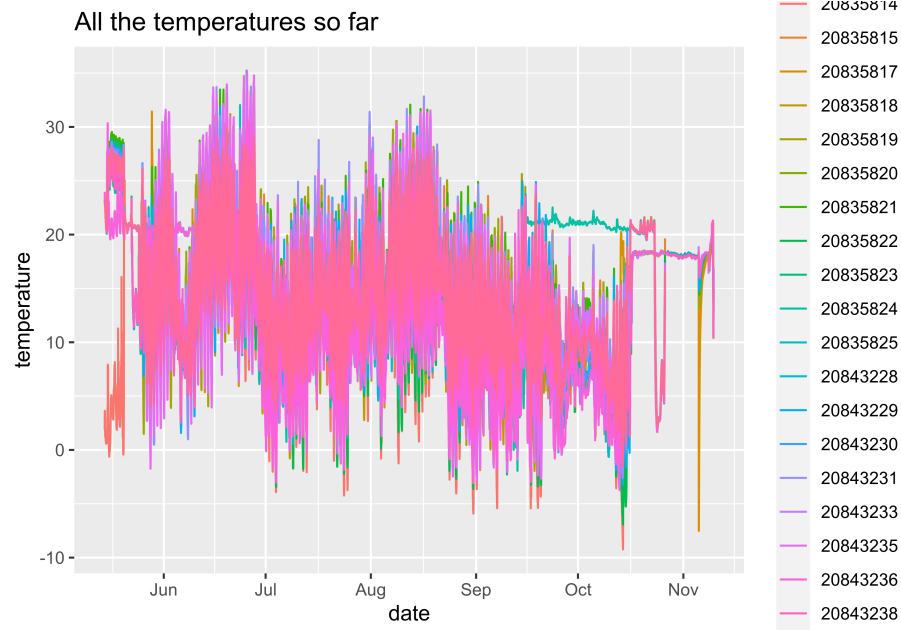
```
all(combDat == combDat2)
```

```
## [1] TRUE
```

## A quick look at the data

Some simple figures.

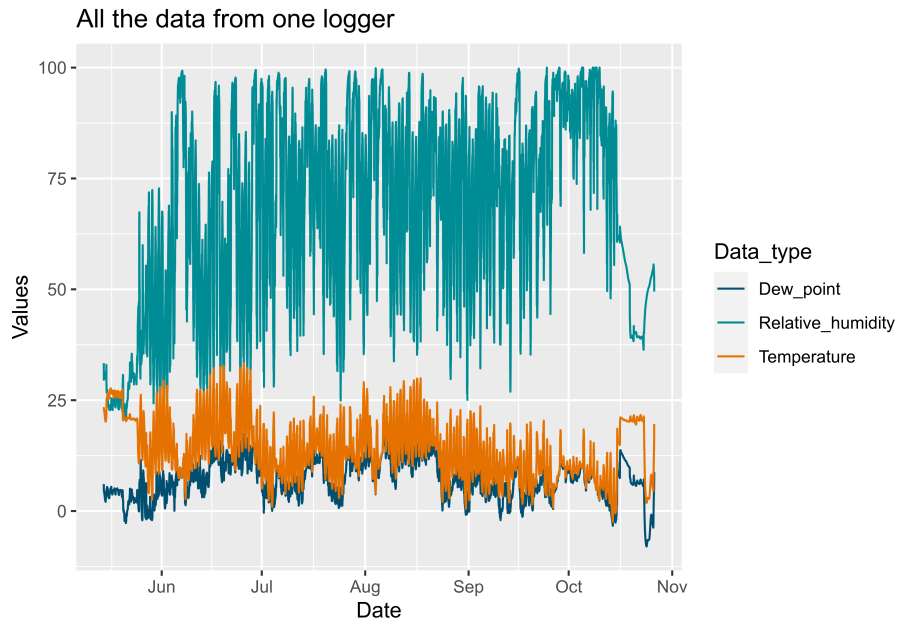
```
ggplot(combDat2) +  
  geom_line(aes(x = date, y = temperature, color = logger_id)) +  
  ggtitle("All the temperatures so far")
```



```
oneLogger <- combDat %>%  
  filter(logger_id == "20835815") %>%  
  select(Date = date,  
         logger_id,  
         Temperature = temperature,  
         Relative_humidity = rh,  
         Dew_point = dew_point) %>%  
  pivot_longer(-c(Date, logger_id),  
              names_to = "Data_type",  
              values_to = "Values")  
  
ggplot(oneLogger) +
```



```
geom_line(aes(x = Date, y = Values, color = Data_type)) +
scale_color_nina() +
ggtitle("All the data from one logger")
```



## Combine with single files from loggers that weren't synced

We had some troubles with the uploads from HoboConnect to Hobolink.com from the CAT-phones. This is probably due to the language setting on the phone (Norwegian). Therefore the some logger files is provided individually by email. Time to combine these as well. These have a different data format than the export from hobolink. They also use comma as a decimal delimiter as well as column delimiter, which complicates things. I think the HoboConnect app might pick the decimal delimiter from the locale of the phones, but doesn't adjust the column delimiter.

Here we manually combine some columns that are actually the integer and decimal values of the same measurement.

```
formatMX2301File <- function(inputFile){
raw <- read_csv(file = inputFile,
               col_types = cols(
`#` = col_integer(),
```

```

  `Date-Time (CET)` = col_datetime(format = "%m.%d.%Y %H.%M.%S"),
  `Ch: 1 - Temperature °C (°C)` = col_integer(),
  `Ch: 2 - RH % (%)` = col_integer(),
  `Dew Point °C (°C)` = col_integer(),
  `Button Down` = col_integer(),
  `Host Connected` = col_integer(),
  `End of File` = col_integer()
))

logger_id <- gsub("(.*)/([0-9]*)(.*)", "\\2", inputFile)

out <- raw %>%
  filter(!is.na(`Ch: 1 - Temperature °C (°C)`)) %>%
  unite("temperature", `Ch: 1 - Temperature °C (°C)`, `Ch: 2 - RH % (%)`, sep = ".") %>%
  unite("rh", `Dew Point °C (°C)`, `Button Down`, sep = ".") %>%
  unite("dew_point", `Host Connected`, `End of File`, sep = ".") %>%
  mutate(logger_type = "MX2301A",
         date = `Date-Time (CET)`,
         logger_id = logger_id) %>%
  select(date,
         logger_type,
         logger_id,
         temperature,
         rh,
         dew_point)

return(out)
}

#logger_20835817 <- formatMX2301File("../rawData/20835817 2020-10-13 12_18_01 CET (Data CET).csv")
#logger_20835819 <- formatMX2301File("../rawData/20835819 2020-10-14 14_33_12 CET (Data CET).csv")
logger_20835820 <- formatMX2301File("../rawData/20835820 2020-10-16 14_09_17 CET (Data CET).csv")

## Warning: 10950 parsing failures.
## row col expected actual file
## 1 -- 8 columns 11 columns '../rawData/20835820 2020-10-16 14_09_17 CET (Data CET).csv'
## 2 -- 8 columns 11 columns '../rawData/20835820 2020-10-16 14_09_17 CET (Data CET).csv'
## 3 -- 8 columns 11 columns '../rawData/20835820 2020-10-16 14_09_17 CET (Data CET).csv'
## 4 -- 8 columns 11 columns '../rawData/20835820 2020-10-16 14_09_17 CET (Data CET).csv'
## 5 -- 8 columns 11 columns '../rawData/20835820 2020-10-16 14_09_17 CET (Data CET).csv'
## ... ..
## See problems(...) for more details.

#logger_20835821 <- formatMX2301File("../rawData/20835821 2020-10-15 15_49_08 CET (Data CET).csv")
#logger_20835823 <- formatMX2301File("../rawData/20835823 2020-10-15 12_11_02 CET (Data CET).csv")

```

```
#logger_20843228 <- formatMX2301File("../rawData/20843228 2020-10-14 11_43_58 CET (Data CET)
#logger_20843229 <- formatMX2301File("../rawData/20843229 2020-10-16 09_59_17 CET (Data CET)
#logger_20843233 <- formatMX2301File("../rawData/20843233 2020-10-14 16_25_22 CET (Data CET)
#logger_20843238 <- formatMX2301File("../rawData/20843238 2020-10-16 16_52_35 CET (Data CET)
```

Combine these files to the other ones.

```
allMX2301 <- combDat2 %>%
  #rbind(logger_20835817) %>%
  #rbind(logger_20835819) %>%
  rbind(logger_20835820)
#%>%
  #rbind(logger_20835821) %>%
  #rbind(logger_20835823) %>%
  #rbind(logger_20843228) %>%
  #rbind(logger_20843229) %>%
  #rbind(logger_20843233) %>%
  #rbind(logger_20843238)
```

## Handle the MX2201 loggers

These are temperature and light loggers that where also placed at some locations (that also had sound loggers). They have slightly different format, so we adapt the function to handle these.

```
longerHobo2202 <- function(inputFile){
  rawDat <- read_csv(inputFile,
    guess_max = 10000,
    col_types = cols())

  dat <- rawDat %>%
    select(-"Line#") %>%
    mutate(date = as.POSIXct(Date, format = "%m/%d/%y %H:%M:%S")) %>%
    #mutate_if(is_character, as.double) %>%
    select(-Date)

  temp <- dat %>%
    pivot_longer(cols = starts_with("Temperature"),
      names_to = "logger_id",
      values_to = "temperature") %>%
    select(date,
      logger_id,
      temperature) %>%
```

```

    filter(!is.na(temperature))

light <- dat %>%
  pivot_longer(cols = starts_with("Light"),
               names_to = "logger_id",
               values_to = "light") %>%
  select(date,
         logger_id,
         light)%>%
  filter(!is.na(light))

temp <- temp %>%
  mutate(logger_id = str_extract(logger_id,
                                "[^, ]+$"))

light <- light %>%
  mutate(logger_id = str_extract(logger_id,
                                "[^, ]+$"))

if(!all(temp$date == light$date)) stop("Tables datetimes doesn't match")

combDat <- temp %>%
  full_join(light,
            by = c("date" = "date",
                  "logger_id" = "logger_id")) %>%
  arrange(logger_id,
          date) %>%
  mutate(logger_type = "MX2202") %>%
  select(date,
         logger_type,
         logger_id,
         temperature,
         light)

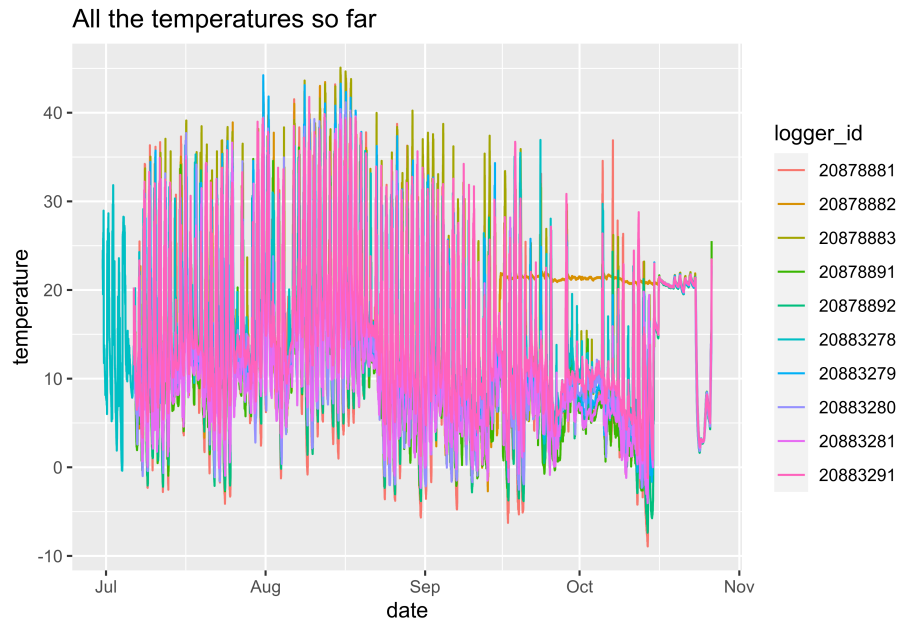
  return(combDat)
}

allMX2202 <- longerHobo2202(inputFile = "../rawData/Insect_MX2202_temp_light_2020_10_27_13_0

```

## A quick look at the MX2202 data

```
ggplot(allMX2202) +  
  geom_line(aes(x = date, y = temperature, color = logger_id)) +  
  ggtitle("All the temperatures so far")
```



These loggers weren't shielded from sunlight as the MX2301, and we can see a larger span in temperatures during the days.

We can have a look at the sunlight measurement and the temperature from one logger for one week. Here, I rescale the light values to values between 1 and 100 to fit the data into the same graph.

```
scaling <- function(x,  
  low = 0,  
  high = 50){  
  out <- (((high - low) * (x - min(x))) / (max(x) - min(x))) + low  
  
  out  
}
```

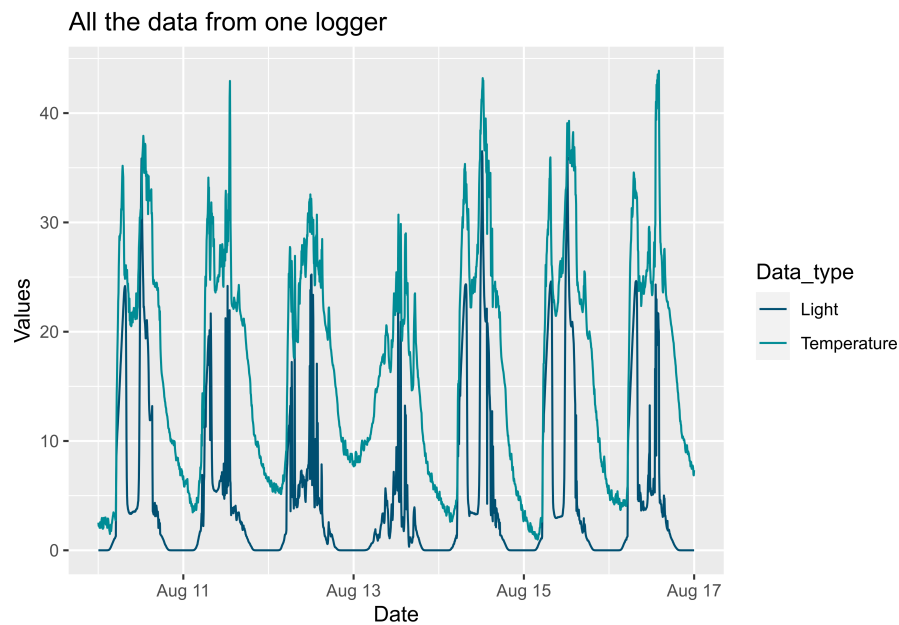
```
oneLogger <- allMX2202 %>%  
  filter(logger_id == "20878881") %>%  
  mutate(light_scaled = scaling(light)) %>%  
  select(Date = date,  
    logger_id,
```

```

    Temperature = temperature,
    Light = light_scaled) %>%
pivot_longer(-c(Date, logger_id),
             names_to = "Data_type",
             values_to = "Values") %>%
filter(Date >= '2020-08-10',
       Date < '2020-08-17'
)

ggplot(oneLogger) +
  geom_line(aes(x = Date, y = Values, color = Data_type)) +
  scale_color_nina() +
  ggtitle("All the data from one logger")

```



The temperature readings seems to be quite heavily affected by the light conditions.