Arranging logger data from HoboLink and importing them to the database

Jens Åström

10 November, 2020

Contents

Intro **1**

Read in data **1**

Package this into a function **6**

Check the data out **8**

Combine with single files from loggers that weren't synced **9**
Double check dew points 11

Handle the MX2201 loggers **11**

Write the data to the database **12**
Make the database table 13

Write the data as CSV **15**

Intro

The data exports for the temperature and humidity MX loggers from Hobo needs a bit of data wrangling before it can be used. The different data streams from each logger all get a separate column. Here we develop a script to turn this into a more usable long format. We also make tables in a database and upload the data there.

Read in data

We have an 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, that 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 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 <dttm>
```

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.

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

The data now looks like this

```
temp
```

```
## # A tibble: 234,096 x 3
##    date                logger_id                                      temperature
##    <dttm>              <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 and merge the tables

```
temp <- temp %>%
  mutate(logger_id = str_extract(logger_id,
```

```
                                "[^, ]+$"))

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

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

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

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

```
combDat <- temp %>%
  full_join(rh,
            by = c("date" = "date",
                   "logger_id" = "logger_id")) %>%
  full_join(dew,
            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)
```

```
combDat
```

```
## # A tibble: 234,096 x 6
##    date                logger_type logger_id temperature    rh   dew
##    <dttm>              <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  <- dat %>%
    pivot_longer(cols = starts_with("Dew"),
                 names_to = "logger_id",
                 values_to = "dew") %>%
    select(date,
           logger_id,
           dew) %>%
    filter(!is.na(dew))
```

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

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

  combDat <- temp %>%
  full_join(rh,
            by = c("date" = "date",
                   "logger_id" = "logger_id")) %>%
  full_join(dew,
            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)

  return(combDat)
}
```

We can check that it produces the same results as the script.

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

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

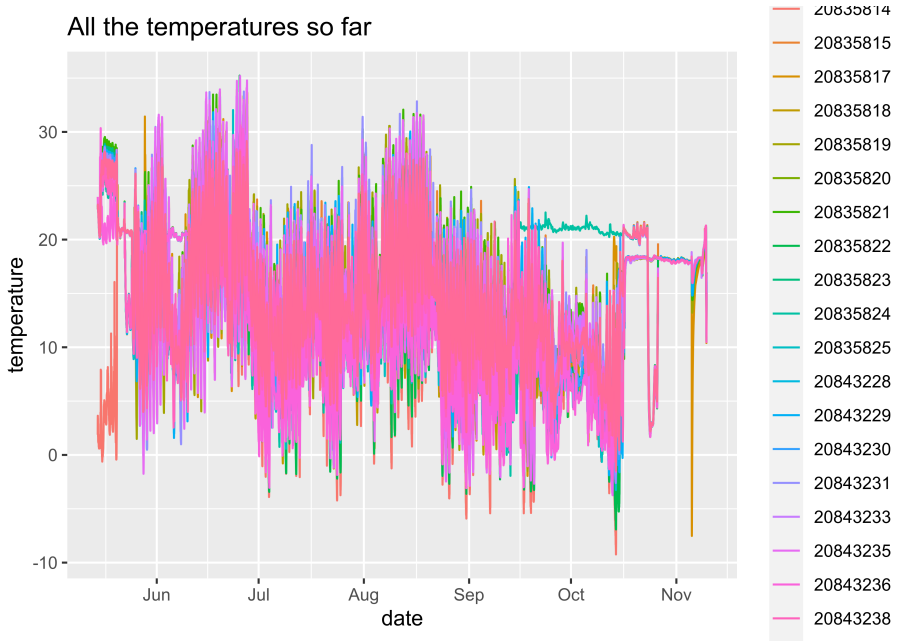
```
all(combDat == combDat2)
```

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

Check the data out

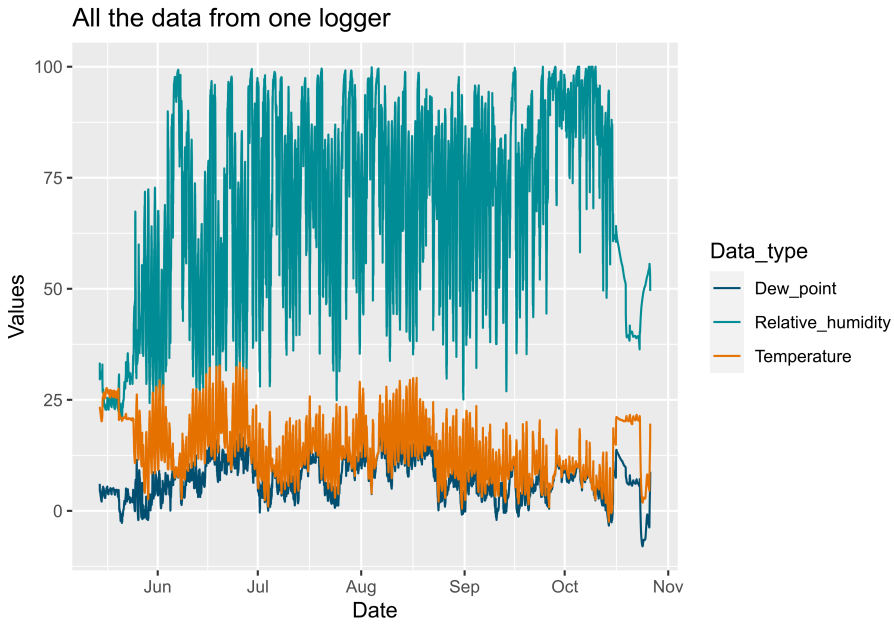
Some simple figures.

```
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) %>%
  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. So the logger files from Oslo is provided individually by email. Time to combine these as well. These have a different data format than the export from hobolink. They also have some extra columns at the end, which we can disregard.

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

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

out <- raw %>%
  mutate(logger_id = logger_id,
         date = as.POSIXct(`Date-Time (CET)`, format = "%m.%d.%Y %H.%M.%S"),
         temperature = as.double(`Ch: 1 - Temperature  °C (°C)`),
         rh = as.double(`Ch: 2 - RH  % (%)`),
         dew = as.double(`Dew Point  °C (°C)`),
         logger_type = "MX2301A") %>%
  select(date,
```

```
        logger_type,
        logger_id,
        temperature,
        rh,
        dew)

return(out)

}
```

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

```
## Warning: 10948 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)
#logger_20835823 <- formatMX2301File("../rawData/20835823 2020-10-15 12_11_02 CET (Data CET)
#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)
```

Double check dew points

This is a simplified formula for dew point, that seems to correspond fairly OK with the logger data. Not likely to be errors here.

```
dewPoint <- function(input){

  input %>%
    mutate(calc_dew_point = temperature - ((100 - rh)/5)) %>%
  select(calc_dew_point)
}
```

```
combDat2
dewPoint(combDat2)

logger_20835817
dewPoint(logger_20835817)
```

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
```

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

Write the data to the database

In the database, we combine the logger types into one table, and make it even longer. I.e. we combine all values in one column. (Many ways to do this).

```
allMX2301Long <- allMX2301 %>%
  pivot_longer(cols = c("temperature", "rh", "dew"),
               names_to = "data_type")

allMX2202Long <- allMX2202 %>%
  pivot_longer(cols = c("temperature", "light"),
               names_to = "data_type")

allLoggersLong <- allMX2301Long %>%
  rbind(allMX2202Long)
```

Make the database table

```
createLoggerSchemaQ <- "
CREATE SCHEMA IF NOT EXISTS loggers;
"

createLoggerTableQ <- "
CREATE TABLE IF NOT EXISTS loggers.logger_data (
--id uuid NOT NULL DEFAULT gen_random_uuid(),
id serial NOT NULL,
date timestamptz NOT NULL,
logger_type text NOT NULL,
logger_id integer NOT NULL,
data_type text NOT NULL,
value double precision
);

"

dbSendQuery(con, createLoggerSchemaQ)
dbSendQuery(con, createLoggerTableQ)

dbSendQuery(con,
            "CREATE INDEX ON loggers.logger_data USING BTREE(date);")

dbSendQuery(con,
            "CREATE INDEX ON loggers.logger_data USING BTREE(logger_type);")

dbSendQuery(con,
            "CREATE INDEX ON loggers.logger_data USING BTREE(logger_id);")
```

```
dbSendQuery(con,
            "CREATE INDEX ON loggers.logger_data USING BTREE(data_type);")
```

```
dbSendStatement(con, "DELETE FROM loggers.logger_data;")
```

```
## <PqResult>
##   SQL  DELETE FROM loggers.logger_data;
##   ROWS Fetched: 0 [complete]
##        Changed: 0
```

```
dbWriteTable(con,
             name = Id(schema = "loggers", table = "logger_data"),
             value = allLoggersLong,
             append = T
             )
```

```
## Warning in result_create(conn@ptr, statement): Closing open result set,
## cancelling previous query
```

Read in logger deployments

```
logger_deployments <- read_delim("../../../Data/klimalogger/logger_deployments_2020.csv",
                                 delim = ";")
```

```
## Parsed with column specification:
## cols(
##   logger_id = col_double(),
##   logger_type = col_character(),
##   year = col_double(),
##   location = col_character()
## )
```

```
createLoggerDeploymentsQ <- "
CREATE TABLE IF NOT EXISTS loggers.logger_deployments (
--id uuid NOT NULL DEFAULT gen_random_uuid(),
id serial PRIMARY KEY,
logger_id integer NOT NULL,
logger_type text NOT NULL,
year integer NOT NULL,
location text NOT NULL
);
"
dbSendQuery(con, createLoggerDeploymentsQ)

dbSendQuery(con,
            "CREATE INDEX ON loggers.logger_deployments USING BTREE(logger_id);")

dbSendQuery(con,
```

```
            "CREATE INDEX ON loggers.logger_deployments USING BTREE(logger_type);")

dbSendQuery(con,
            "CREATE INDEX ON loggers.logger_deployments USING BTREE(year);")

dbSendQuery(con,
            "CREATE INDEX ON loggers.logger_deployments USING BTREE(location);")
```

```
dbWriteTable(con,
             name = Id(schema = "loggers", table = "logger_deployments"),
             value = logger_deployments,
             append = T
             )
```

Write the data as CSV

We also write the data as csv files, if we don't want to use the database.

```
write_csv(allLoggersLong, path = "../out/insectLogger_data_2020.csv")
write_csv(logger_deployments, path = "../out/insect_logger_deployments.csv")
```